UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE INGENIERÍA

ESCUELA DE CIENCIAS Y SISTEMAS

ARQUITECTURA DE COMPUTADORES Y ENSAMBLADORES I

Proyecto

MANUAL TECNICO

Integrantes Grupo 3

Nombre:	Sara Paulina Medrano Cojulún	Carné:201908053
Nombre:	Marvin Eduardo Catalán Véliz	Carné:201905554
Nombre:	Julio José Orellana Ruíz	Carné: 201908120
Nombre:	Diego Fernando Cortez López	Carné: 201900955
Docente:	Ing. Otto Escobar	
Auxiliar:	Oscar Bernard	

Fecha: 10/03/2022

CONTENIDO

DESCRIPCIÓN

Este proyecto consiste en una empresa que brinda servicios de parqueo, por medio de una aplicación móvil , en esta aplicación se podrán crear los usuarios y así mismo hacer login a la página desde la cual se podrá ingresar al parqueo inmediatamente o así mismo reservar un parqueo.

Se realizo un modelo que simula este sistema, con un app móvil que se conecta por medio de bluetooth y un microcontrolador de arduino , y el sistema de parqueo en proteus para poder observar los resultados del mismo de manera simulada y así decidir si son la empresa de desarrollo correcta para lograr llevar a cabo la realización de su idea.

Los procesos que realiza este sistema son:

- Creación de usuarios en la aplicación móvil.
- Manejo de tokens para el ingreso al parqueo.
- Ingreso de usuarios a la aplicación movil.
- Control de sus espacios de parqueo por medio de la memoria.
- Control de una alarma.
- Control gráfico de una matriz de leds.

REQUERIMIENTOS TECNICOS PARA EJECUCIÓN

- Arduino IDE versión 1.8.19.
- Arduino library stepper.
- Procesador superior a core i5 7^{th}.gen

MATERIALES UTILIZADOS

- Fotorresistencias o LDR.
- Dos motores stepper (uno con driver y otro sin).
- Teclado alfanumérico matricial de 4x4.
- Módulo Bluetooth
- Pantalla LCD.
- Buzzer
- Matriz LED 8x8
- Driver Max 7219/7221

CÓDIGO IMPLEMENTADO

DESCRIPCIÓN ARDUINO

Arduino es una plataforma de creación de electrónica de código abierto, la cual está basada en hardware y software libre, flexible y fácil de utilizar para los creadores y desarrolladores. Esta plataforma permite crear diferentes tipos de microordenadores de una sola placa a los que la comunidad de creadores puede darles diferentes tipos de uso.

Para poder entender este concepto, primero vas a tener que entender los conceptos de hardware libre y el software libre. El hardware libre son los dispositivos cuyas especificaciones y diagramas son de acceso público, de manera que cualquiera puede replicarlos. Esto quiere decir que Arduino ofrece las bases para que cualquier otra persona o empresa pueda crear sus propias placas, pudiendo ser diferentes entre ellas pero igualmente funcionales al partir de la misma base.

El software libre son los programas informáticos cuyo código es accesible por cualquiera para que quien quiera pueda utilizarlo y modificarlo. Arduino ofrece la plataforma Arduino IDE (Entorno de Desarrollo Integrado), que es un entorno de programación con el que cualquiera puede crear aplicaciones para las placas Arduino, de manera que se les puede dar todo tipo de utilidades.

METODOS ARDUINO

VOID SHOWSCREEN ()

En este método es la parte de inicio de la ejecución del programa. Esta parte se realizó con pura utilización de las clases LCD. Se utilizo el setCursor para darle un apartado en la pantalla. Un lcd.print para pintar en pantalla y un lcd.write para escribir el icono del condado.

VOID GENTOKEN ()

En este método es una parte fundamental para las conexiones de entrada y otras funcionalidades. En este método principalmente tenemos un método semilla, al método semilla es el que usamos para generar números aleatorios. Este va conectado en una salida del Arduino análogo.

Primero generamos los dos números con simples random en cada número. Luego le aplicamos un for para extraer las dos palabras. Y las palabras hay que tenerlas en un rango de A hasta D que son las letras que tendrá el teclado y así obtenemos el token, luego este token será enviado a la APP.

VOID VALIDARTOKEN ()

Este método se trata de validar el token por el ingrese del teclado en proteus. Este método hace 3 posibles divisiones. En la cual cada salida tiene una ejecución directa a la APP o método de revalidación.

1era posible salida: presiono el botón CANCEL. En este botón es directamente si se presiona el botón cancelar en el teclado al momento de estar ingresando el token.

2da posible salida: fallo de intento en el ingreso en el token de 3 veces consecutivas. Este caso pasaría si el usuario que esta ingresando el token lo ingresa de manera errónea los tres intentos y esto provocara la salida del sistema a llevar a una consulta si el quiere pedir token o quiere salir de pedir token.

3era posible salida: token ingresado correctamente. Si el token se ingreso de forma correcta esto nos llevara a lo que es la entrada del parqueo y en ese proceso lleva lo de abrir portones.

Para pedir los valores del teclado de proteus se realiza un bucle infinito para que no se pierda ni un ingreso en el teclado. Y ahí es donde se ira realizando la validación.

Para el botón cancel se valida con un do while y esto realizara un escape de todo el ciclo.

VOID COMPROBARTOKEN(STRING PASS)

Este método fue utilizado anteriormente en la validación de tokens. Explícitamente realiza la comprobación de token.

En primero comprueba los tamaños del PASS con el Token.

En segundo y consecutivo validación de cada letra que sea la misma.

METODOS CONTROL DE PUERTAS

VOID ABRIRPORTON()

En este método se realizó el proceso para girar un motor stepper que simula que una puerta se esta abriendo por medio de la librería stepper le mandamos al motor un setSpeed con parámetro de 10 u un step donde le mandamos la revoluciones para que pueda girar, en este método también se enciende un led color roja cuando gira el motor.

VOID CERRARPORTON()

En este método se realizo el proceso para girar un motor stepper que simula que una puerta se esta cerrando por medio de la librería stepper le mandamos al motor un setSpeed con parámetro de 10 y un step donde le mandamos las revoluciones para que pueda girar, en este método también se enciende un led color amarrilla cuando gira el motor.

VOID CONTROLPUERTAS()

En este método se tiene un contador y se entra a una condición en la cual solo se entra si el contador es 0. Dentro de esta condición se tiene un ciclo for para que el proceso solo se repite 1 vez dentro de este for se manda a llamar el metro abrirpuerta, se deja un tiempo de 5 segundo y se llama el método de cerrarpuerta, estos métodos son los descritos arriba.

VOID SALIDAPUERTAS()

Esto método realiza exactamente lo mismo que abrir puerta, solo que simula que el carro sale del parqueo.

VOID ENTRADAPARQUEO()

En este método se realiza un delay de 1 segundo para mandar a llamar al método descrito anteriormente llamado control de puertas donde se realiza todo el proceso de girar los motores stepper.

VOID SELECCIONPARQUEO()

En este método se tiene una condición donde si el usuario se tiene apacha el botón salir se manda a llamar al método salidaPuertas y si no lo apacha se activara la opción de seleccionar parqueo.

VOID PINESMATRIZ()

En este método se declaran los pines que de la matriz de 4x4 de las fotorresistencias que simula el parqueo.

VOID LLENARMATRIZ()

En este método se utiliza para llenar en una matriz los lugares que se encuentran ocupados en el parqueo que reciben una señal por medio de las fotorresistencias.

VOID CONVERTIR4TO8()

Para poder mostrar los lugares ocupados en el parqueo se visualizan en un led de 8x8. Para mostrar los lugares se utiliza este método que convierte la matriz de 4x4 con los lugares ocupados en una matriz de 8x8 y utilizar esa matriz para mandarle que led se deben encender.

VOID VISUALIZARLED()

En este método se recorre la nueva matriz de 8x8 con los lugares correspondientes y si la posición se encuentra ocupada, el led manda la señal que encienda la led en la ubicación seleccionada.

VOID GUARDARPOSICIONES()

En este método se guardan las matrices del parqueo en la EEPROM para que sea almacenada en la memoria del Arduino. Se obtiene el tamaño definir la nueva dirección en donde se almacenarán los usuarios en la EEPROM.

VOID GETUSUARIOS()

Con el método get() de la librería EEPROM se recorre las posiciones para obtener el objeto almacenado en esa posición. Se almacenan en una lista los usuarios que se encuentran guardados.

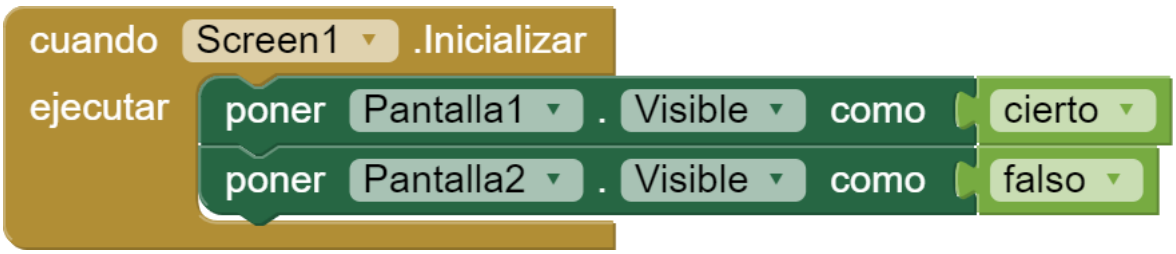
DESCRIPCIÓN APP MOVIL

La aplicación móvil es una aplicación destinada a tener varios controles sobre el parqueo a realizar, algunos son los de registrar usuario, iniciar sesión, solicitar token para poder entrar al parqueo, etc. A continuación se detallan cada uno de sus métodos divididos por bloques. Cabe resaltar que esta app se creó por medio de una herramienta llamada MiAppInventor2 que genera aplicaciones móviles para teléfonos que soporten instalaciones por medio de .APK.

METODOS APP MOVIL

CUANDO SCREEN1.INICIALIZAR

En este bloque de código lo que realizamos es poner en true la pantalla 1 y el false la pantalla dos, ya que la aplicación se divide en 2 disposiciones verticales para poder mantener la conexión bluetooth.

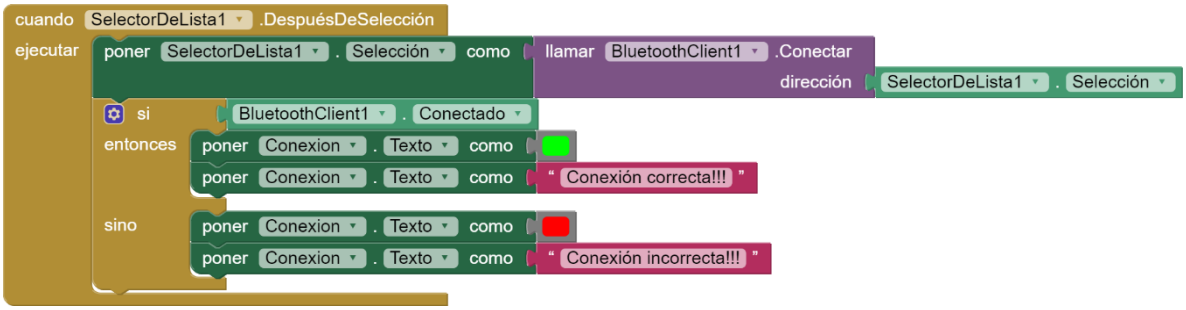


CUANDO SELECTORDELISTA1.ANTESDESELECCION

En este bloque empezamos con la comunicación por bluetooth con la que nos enlazaremos al circuito HC-05 que tenemos en nuestro Arduino Mega simulado en Proteus, en este caso antes de seleccionar la lista de elementos solicitamos al cliente bluetooth las direcciones y nombres existentes en el teléfono.

CUANDO SELECTORDELISTA1.DESPUESDESELECCION

En este bloque después de seleccionar en la lista, llamamos al cliente bluetooth de nuevo para poder realzar la conexión, después validamos si el bluetooth esta conectado para así mandar mensaje de éxito o si no conecta lanzar alerta de error.

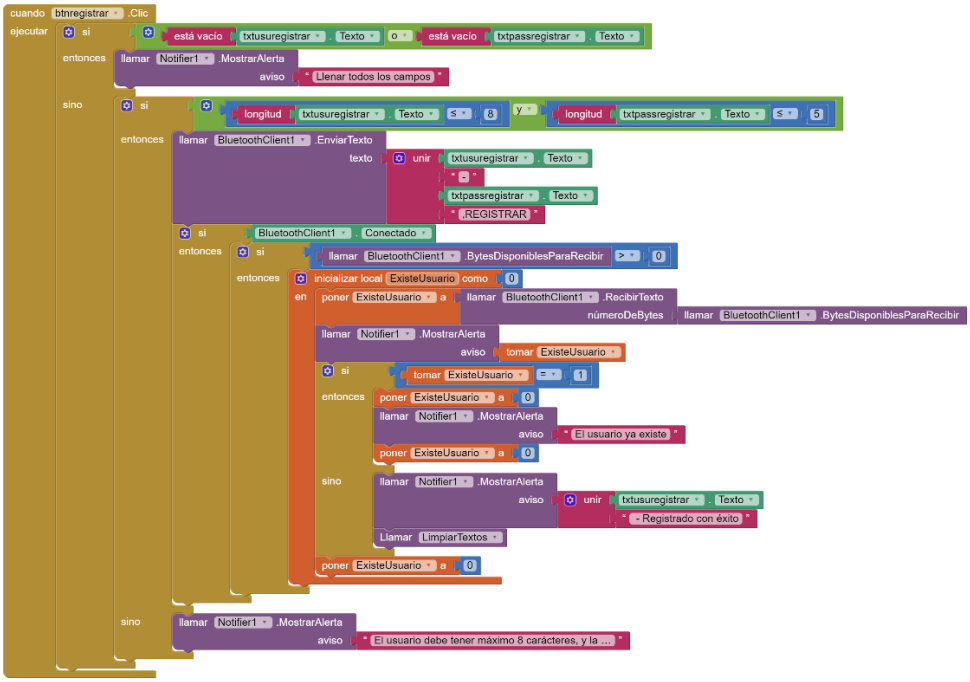


CUANDO BTNREGISTRAR.CLIC

Este bloque es la acción que realizaremos cuando al botón de registrar usuario se le de clic, tiene varias validaciones al iniciar, la primera es una validación de que los campos no estén vacíos sino se finaliza el evento con un mensaje que dice “llenar todos los campos”, al pasar esta validación tenemos otra condicional para validar la longitud tanto de el usuario a crear como la contraseña del mismo, validando de que el usuario sea menor o igual a ocho caracteres y que la contraseña sea menor o igual a cinco caracteres. Posteriormente a esto ya empezamos con la comunicación hacia nuestro Arduino, lo primero que realizamos es hacer una conexión enviando texto, vamos a enviar un único texto donde ira el usuario, contraseña y separado por una coma la función que se realizará, posteriormente a esto del lado de Arduino se splitea por la contraseña entonces nos quedan 2 variables una donde vamos a encontrar el usuario seguido de su respectiva contraseña y otra donde tendremos la

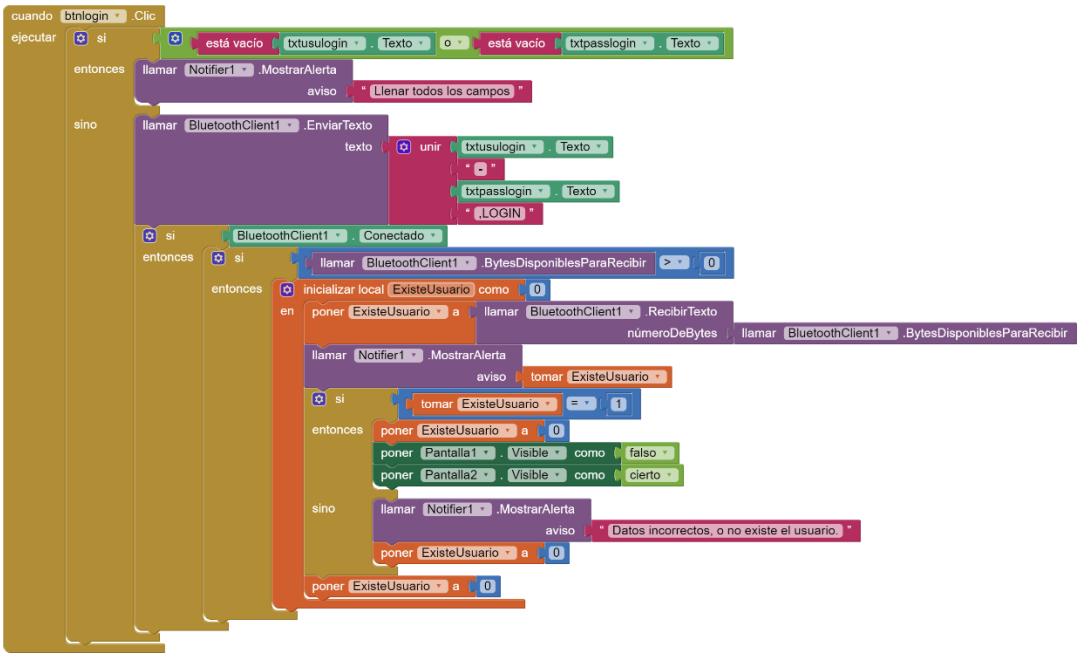
función a ejecutar que en este caso es *REGISTRAR* y validando que la función sea registrar ejecutamos la función necesaria para poder ejecutar las acciones de registrar un usuario en la memoria EPROM, primero validando si el usuario existe claramente.

Después llega el momento de solicitar datos para poder indicarle al usuario que si el usuario se pudo registrar o si el usuario ingresado ya existía. Entonces pedimos datos enviados desde Arduino y si este nos devuelve un verdadero o un falso lanzaremos diferentes mensajes.



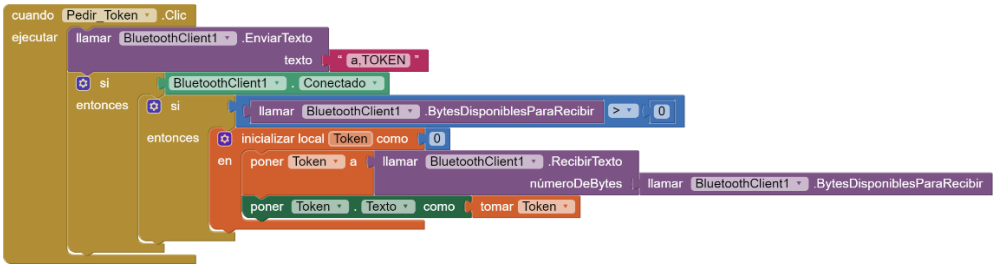
CUANDO BTNLOGIN.CLIC

Este bloque lo activamos cuando se le da clic al botón de iniciar sesión, la primera validación es la misma que el de registrar, que no haya campos vacíos. Y de igual manera para verificar si el usuario exista mandamos usuario, contraseña y el nombre de la función que queremos ejecutar en el Arduino, que en este caso sería *LOGIN*. Posteriormente a esto, desde Arduino mandamos los datos de que, si hace match con algún usuario existente, y si este nos devuelve un true, nosotros le daremos acceso al usuario a la siguiente pantalla y esto lo realizamos ocultando la pantalla 1 y pasándolo a la pantalla 2.



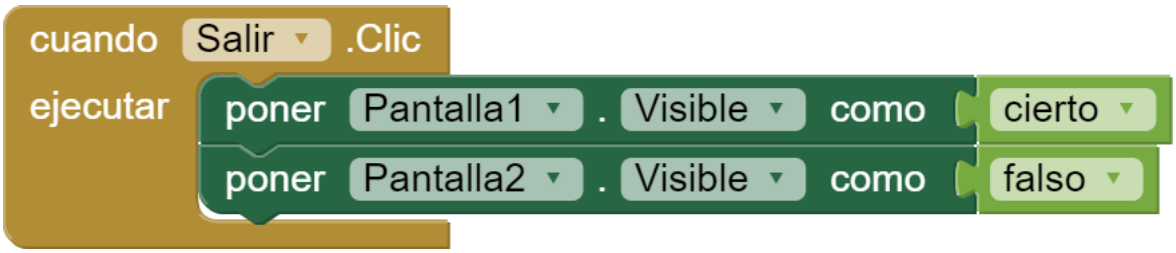
CUANDO PEDIR_TOKEN.CLIC

Este bloque se ejecuta en la pantalla 2, es un botón el cual pide al Arduino el token que le permitirá al usuario ingresar al parqueo y lo pondrá en una etiqueta que se encuentra en la pantalla.



CUANDO SALIR.CLIC

Este es un botón situado en la pantalla dos que lo presionará el usuario cuando quiera sacar un vehículo de el parqueo, y lo único que hace es habilitar la pantalla 1 y además de esto manda la señal para borrar el espacio ocupado en la matriz en la que están representados los parqueos.



COMO LIMPIARTEXTOS

Este es un bloque que es un procedimiento que podemos llamar en cualquier lado de la aplicación que lo único que hace es limpiar todos los cuadros de textos luego de registrar un usuario.