

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE INGENIERÍA

ARQUITECTURA DE COMPUTADORES Y ENSAMBLADORES 1

SECCIÓN: N

CATEDRÁTICO: ROBINSON PÉREZ



Ayeser Cristián Oxlaj Juarez - 202010025

Luis Angel Barrera Velásquez - 202010223

Esteban Humberto Valdez Ennati - 202011535

Oward Francisco Alberí Sian Solis - 201901807

Josué Rodolfo Morales Castillo - 202010033

Gerber Emerson Ordoñez Tucubal - 202004060

Elías Abraham Vásquez Soto - 201900131

GUATEMALA, 16 DE DICIEMBRE DEL 2,022

# ÍNDICE

<b>CÓDIGO ARDUINO</b>	<b>2</b>
<b>CÓDIGO APLICACIÓN ANDROID</b>	<b>8</b>
<b>CODIGO DE SERVICIOS</b>	<b>17</b>
<b>DIAGRAMA DEL CIRCUITO</b>	<b>29</b>
<b>DESCRIPCIÓN HERRAMIENTAS UTILIZADAS</b>	<b>33</b>

## CÓDIGO ARDUINO

```
#include <LiquidCrystal.h>

#define COLS 16 // Columnas del LCD
#define ROWS 2 // Filas del LCD
#define VELOCIDAD 100 // Velocidad a la que se mueve el texto

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

// Textos

char *s;

String des="";
String ocp="";
String re="";

String datos="";

//Aquí creamos los nuevos caractere
    byte parte1[8] = { B01111, B10001, B10001, B10001, B10001, B10001, B11111, B11111 };
    byte parte2[8] = { B11100, B00011, B00001, B00001, B00001, B00001, B11111, B11111 };
    byte parte3[8] = { B00000, B00000, B00000, B00000, B00000, B11000, B11100, B11100 };
    byte parte4[8] = { B11111, B11111, B11111, B01001, B01001, B01001, B00110, B00000 };
    byte parte5[8] = { B11111, B11111, B11111, B11110, B00010, B00010, B00001, B00000 };
    byte parte6[8] = { B11111, B11111, B11111, B01000, B01000, B01000, B10000, B00000 };

    byte parte11[8] = { B01111, B10001, B10001, B10001, B10001, B10001, B11111, B11111 };
    byte parte22[8] = { B11100, B00011, B00001, B00001, B00001, B00001, B11111, B11111 };
    byte parte33[8] = { B00000, B00000, B00000, B00000, B00000, B11000, B11100, B11100 };
```

```

    byte parte44[8] = { B11111, B11111, B11111, B01001, B01001, B01001,
B00110, B00000 };
    byte parte55[8] = { B11111, B11111, B11111, B11110, B00010, B00010,
B00001, B00000 };
    byte parte66[8] = { B11111, B11111, B11111, B01000, B01000, B01000,
B10000, B00000 };
    byte x1[8] ={ B10001, B01010, B00100, B01010, B10001, B00000, B00000,
B00000 };

void setup() {
    Serial.begin(9600);
    lcd.begin(COLS, ROWS);

    lcd.createChar(0, x1);
    lcd.createChar(1, parte1);
    lcd.createChar(2, parte2);
    lcd.createChar(3, parte3);
    lcd.createChar(4, parte4);
    lcd.createChar(5, parte5);
    lcd.createChar(6, parte6);
    lcd.createChar(7, x1);
}

void loop() {

    if (Serial.available()>0){
        String option = Serial.readString();
        s = strtok(option.c_str(), ",");
        String movimiento = strtok(NULL, ",");

        re="";
        if (option != ""){
            s = strtok(NULL, ",");
            des=s;

```

```

    s = strtok(NULL, ",");
    ocp=s;

    s = strtok(NULL, ",");
    re=s;
}

datos="";
datos="Disp: "+des +" Ocup: "+ocp+" Reser: "+re;
int tamano_texto=datos.length();
for(int i=tamano_texto;i>0;i--){
    String texto=datos.substring(i-1);
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print(texto);
    delay(VELOCIDAD);
}

for(int i=1;i<=16;i++){
    lcd.clear();
    lcd.setCursor(i,0);
    lcd.print(datos);
    delay(VELOCIDAD);
}

for(int i=16;i>=1;i--){
    lcd.clear();
    lcd.setCursor(i,1);
    lcd.print(datos);
    delay(VELOCIDAD);
}

for(int i=1;i<=tamano_texto;i++){
    String texto=datos.substring(i-1);
    lcd.clear();
    lcd.setCursor(0,1);
    lcd.print(texto);
    delay(VELOCIDAD);
}
delay(100);

```

```

if(movimiento!="FULL") {

for(int i=1;i<=16;i++){
    lcd.clear();
    lcd.setCursor(i, 0);
    lcd.write(byte(1));

    lcd.setCursor(i+1, 0);
    lcd.write(byte(2));

    lcd.setCursor(i+2, 0);
    lcd.write(byte(3));

    lcd.setCursor(i, 1);
    lcd.write(byte(4));

    lcd.setCursor(i+1, 1);
    lcd.write(byte(5));

    lcd.setCursor(i+2, 1);
    lcd.write(byte(6));

    delay(100);

}
}else{

    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.write(byte(0));
    lcd.setCursor(1, 0);
    lcd.write(byte(1));

    lcd.setCursor(2, 0);
    lcd.write(byte(2));

    lcd.setCursor(3, 0);
    lcd.write(byte(3));
//x

    lcd.setCursor(4, 0);
    lcd.write(byte(0));

```

```

// pausa
    lcd.setCursor(0, 1);
    lcd.write(byte(0));

    lcd.setCursor(1, 1);
    lcd.write(byte(4));

    lcd.setCursor(2, 1);
    lcd.write(byte(5));

    lcd.setCursor(3, 1);
    lcd.write(byte(6));

    lcd.setCursor(4, 1);
    lcd.write(byte(7));

    delay(1500);

}

}

delay(95);

}

```

```

#define ZERO 11
#define NINETY 10
#define OPEN 9
#define CLOSE 8

int state = 0;

void setup(){
    pinMode(ZERO, OUTPUT);
    pinMode(NINETY, OUTPUT);
    pinMode(OPEN, INPUT);
    pinMode(CLOSE, INPUT);
}

```

```

digitalWrite(ZERO, HIGH);
digitalWrite(NINETY, LOW);
}

void loop(){
  switch(state){
    case 0:
      if(digitalRead(OPEN) == HIGH){
        // ABRIENDO TALANQUERA POR MEDIO DEL BOTÓN
        digitalWrite(ZERO, LOW);
        digitalWrite(NINETY, HIGH);
        state = 1;
      }
      break;
    case 1:
      if(digitalRead(CLOSE) == LOW){
        // CERRANDO TALANQUERA CUANDO EL SENSOR SE ALEJA
        digitalWrite(ZERO, HIGH);
        digitalWrite(NINETY, LOW);
        state = 0;
      }
      break;
  }
}

```



# CÓDIGO APLICACIÓN ANDROID

## INICIO

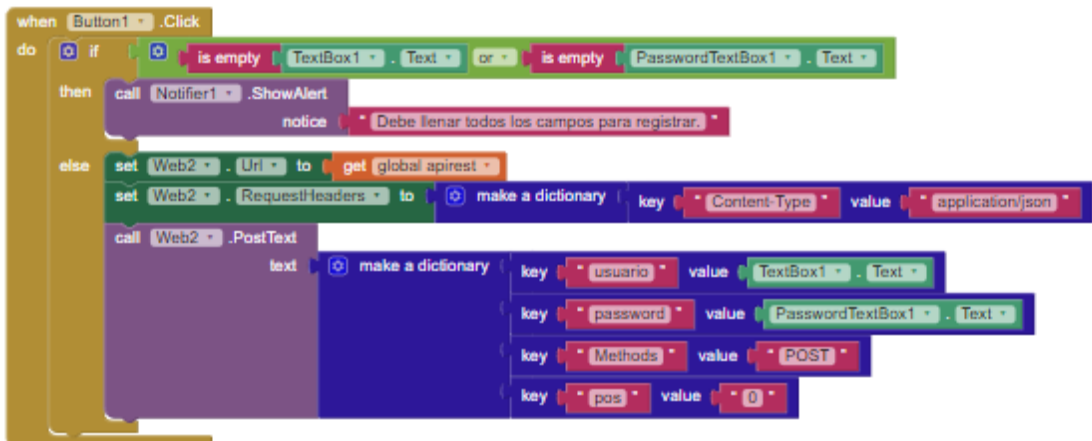
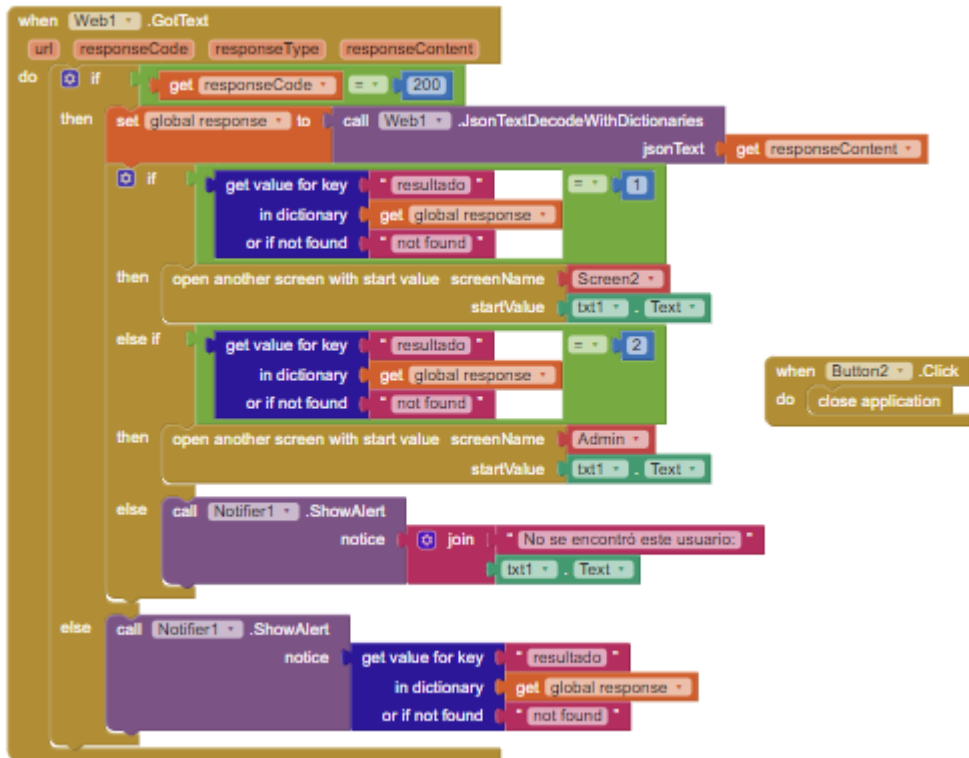


```
initialize global apirest to "https://arqui1grupo9.website/base.php"
```

```
initialize global response to create empty list
```

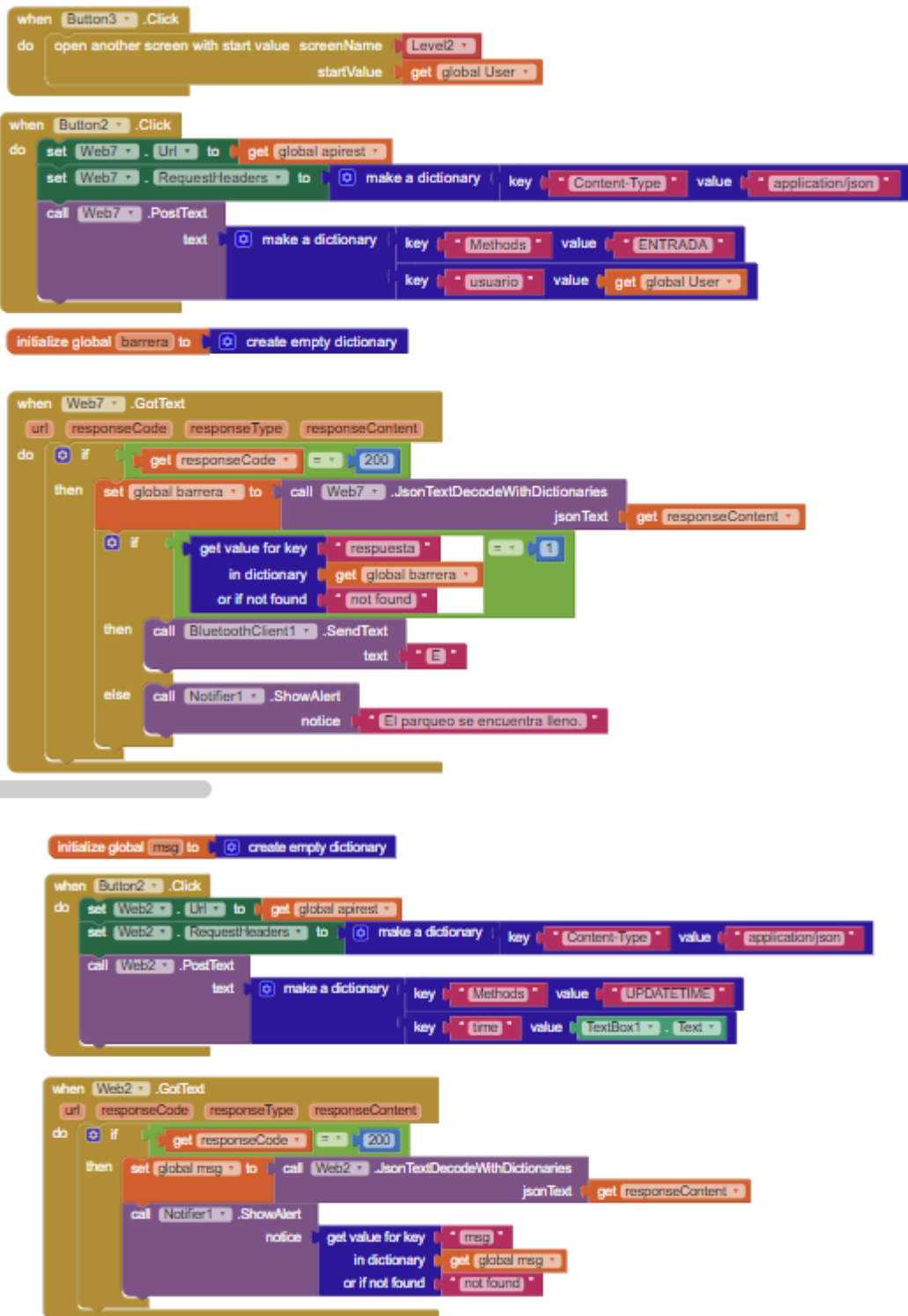
```
initialize global response2 to create empty dictionary
```

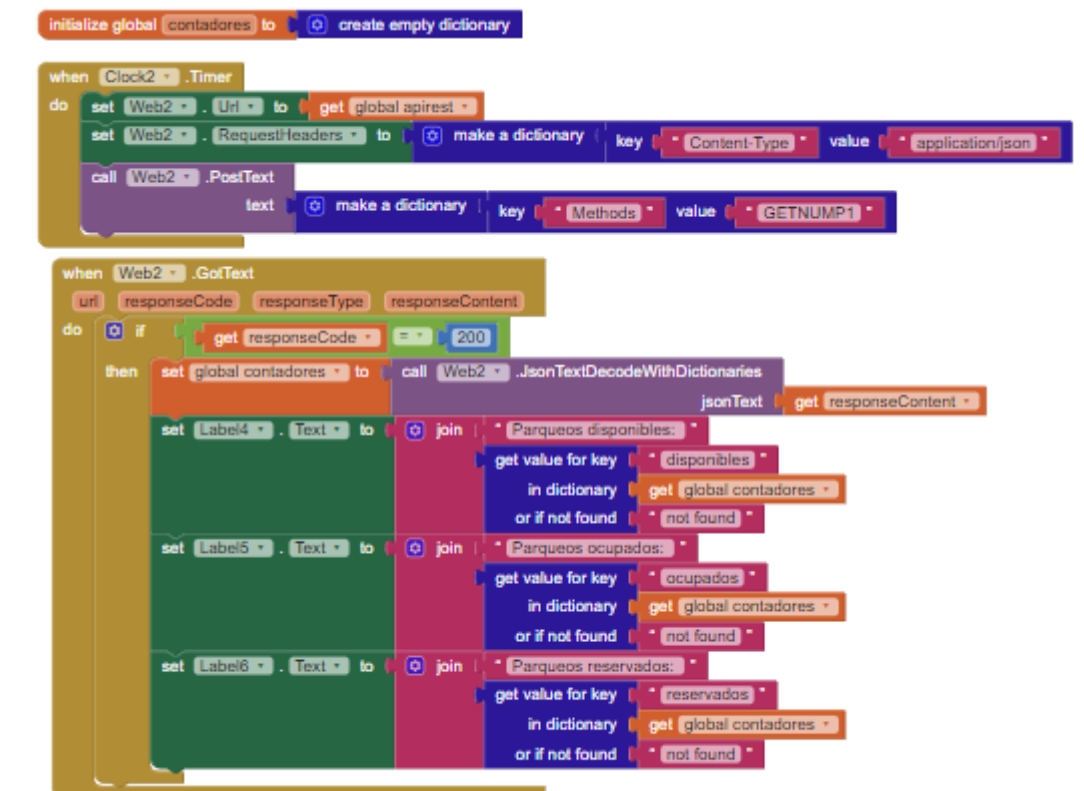
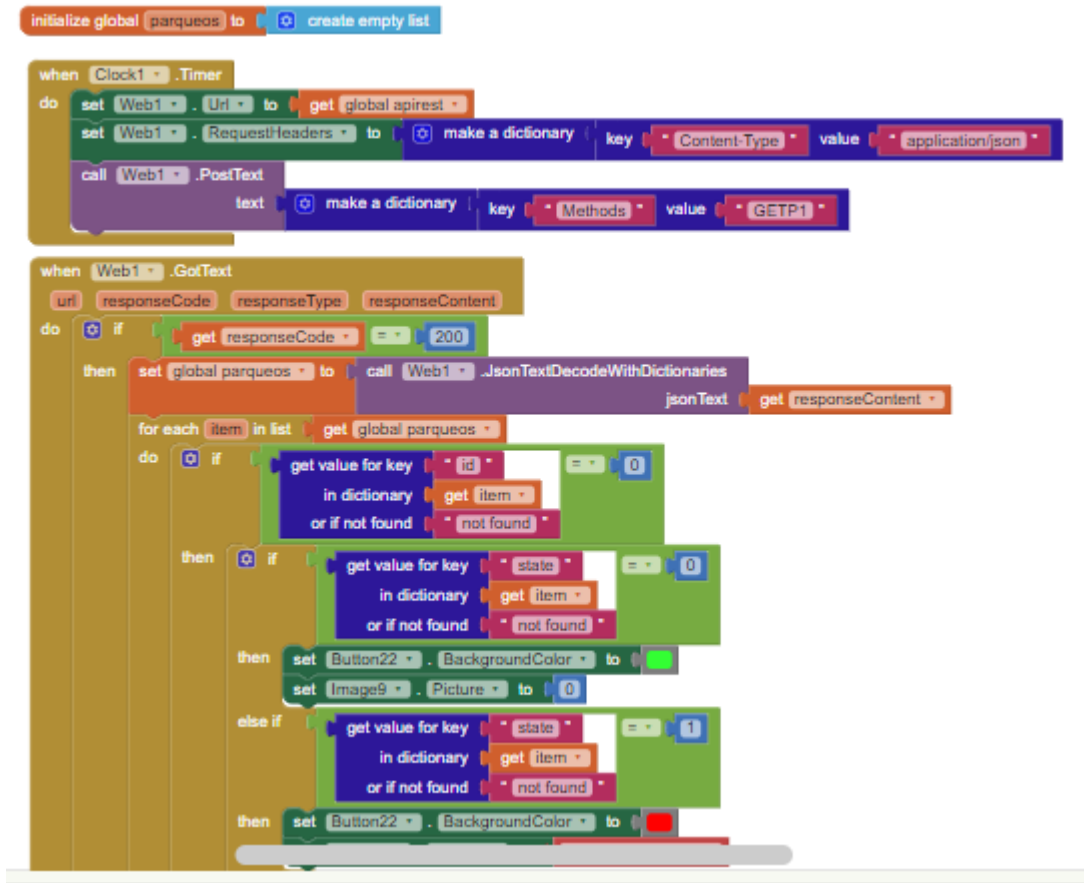
```
when btn1 . Click
do
  if is empty txt1 . Text or is empty txt2 . Text
  then
    call Notifier1 . showAlert
    notice "Debe llenar todos los campos."
  else
    set Web1 . Url to get global apirest
    set Web1 . RequestHeaders to make a dictionary key "Content-Type" value "application/json"
    call Web1 . PostText
    text make a dictionary
    key "usuario" value txt1 . Text
    key "password" value txt2 . Text
    key "Methods" value "GET"
    key "pos" value "0"
```

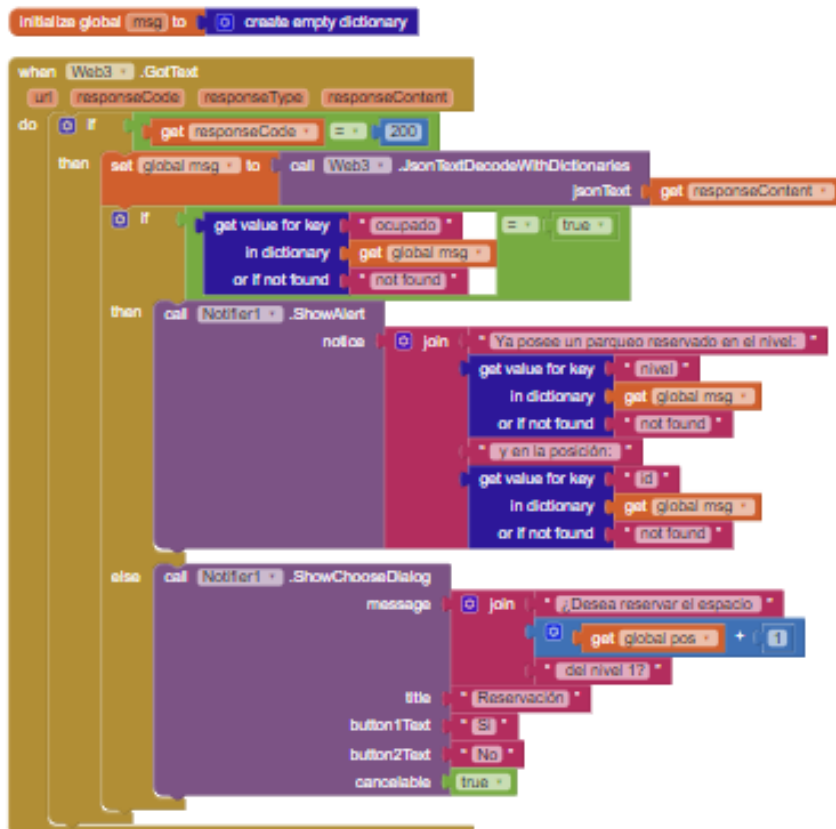
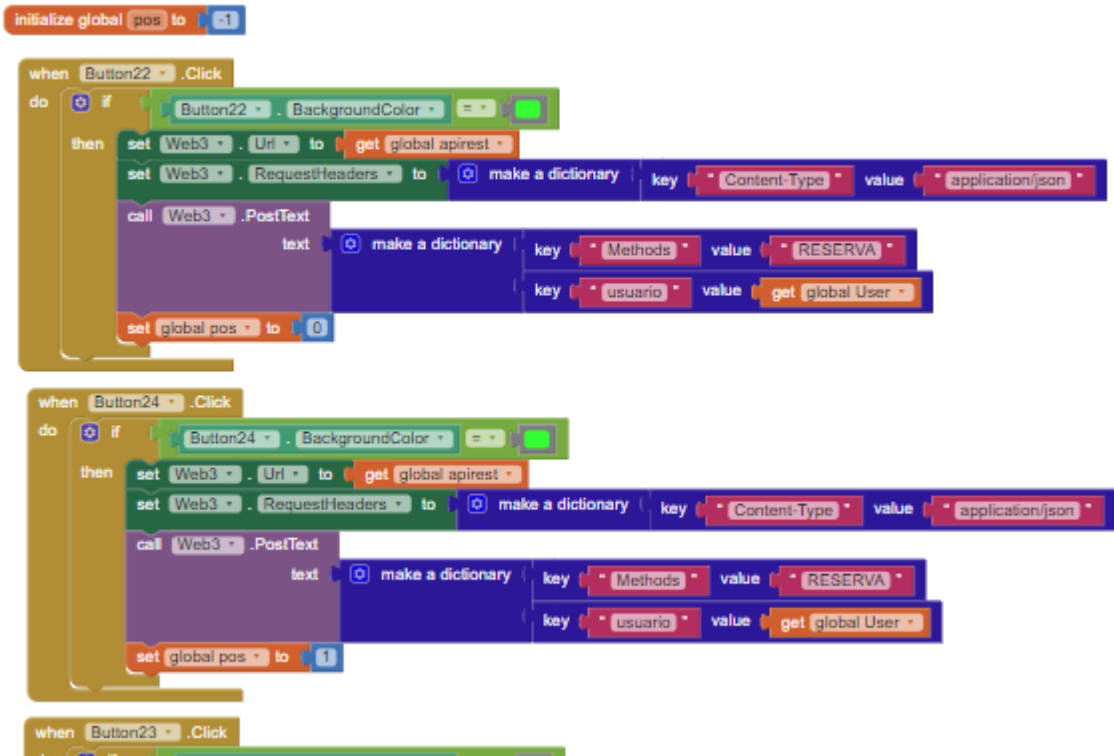


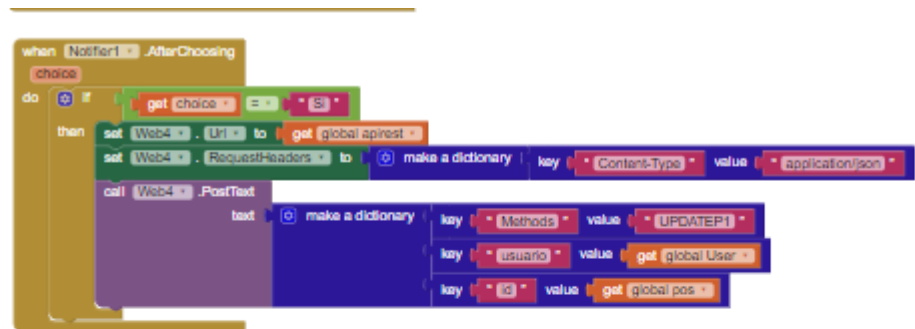
## NIVEL 1 Y 2 DEL PARQUEO.



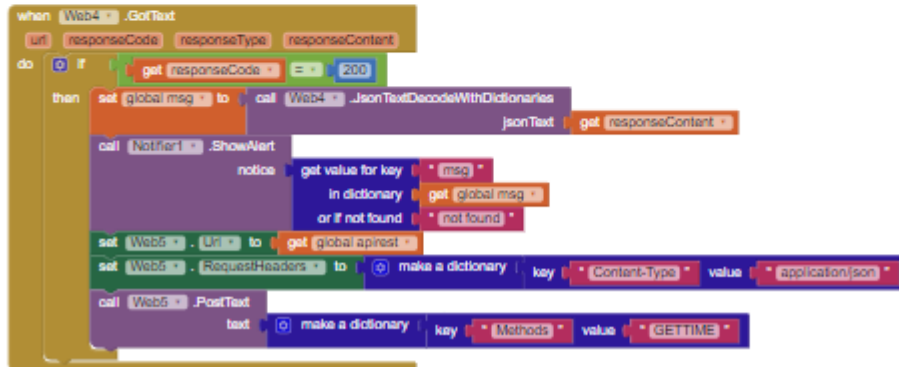




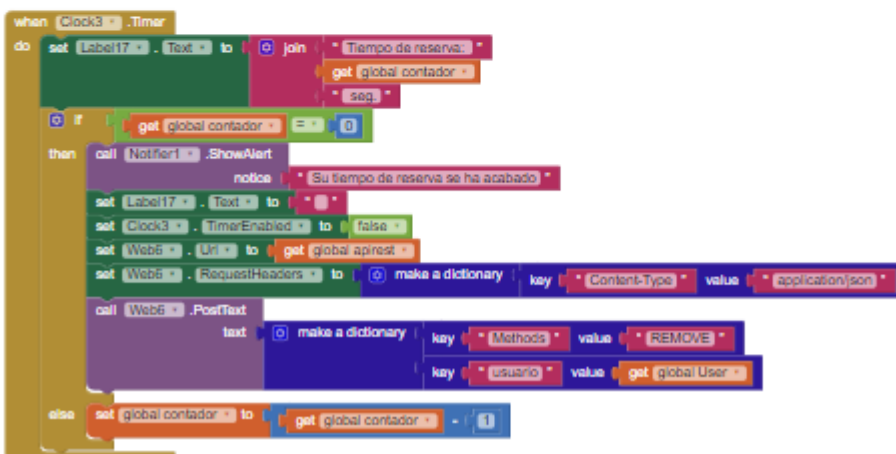
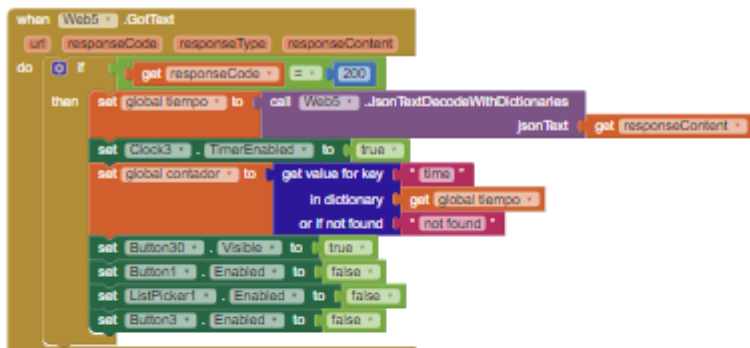


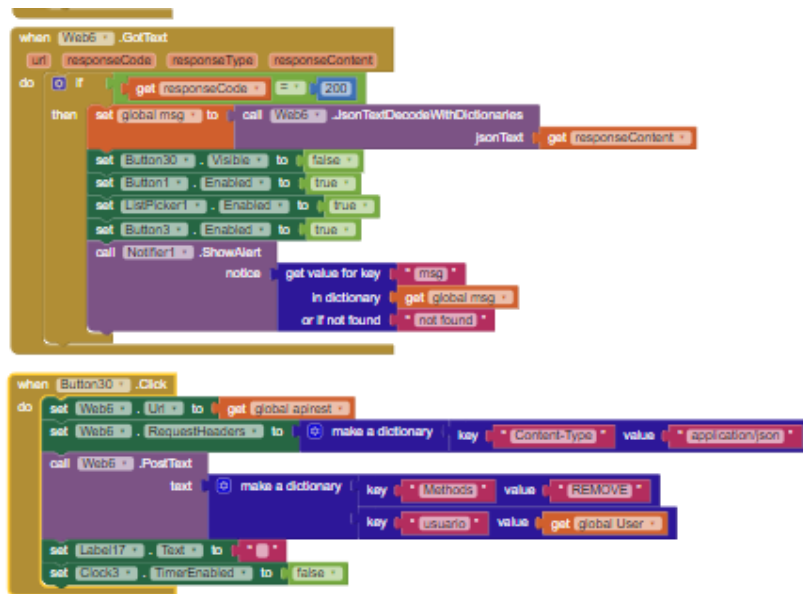


initialize global [tiempo] to [create empty dictionary]

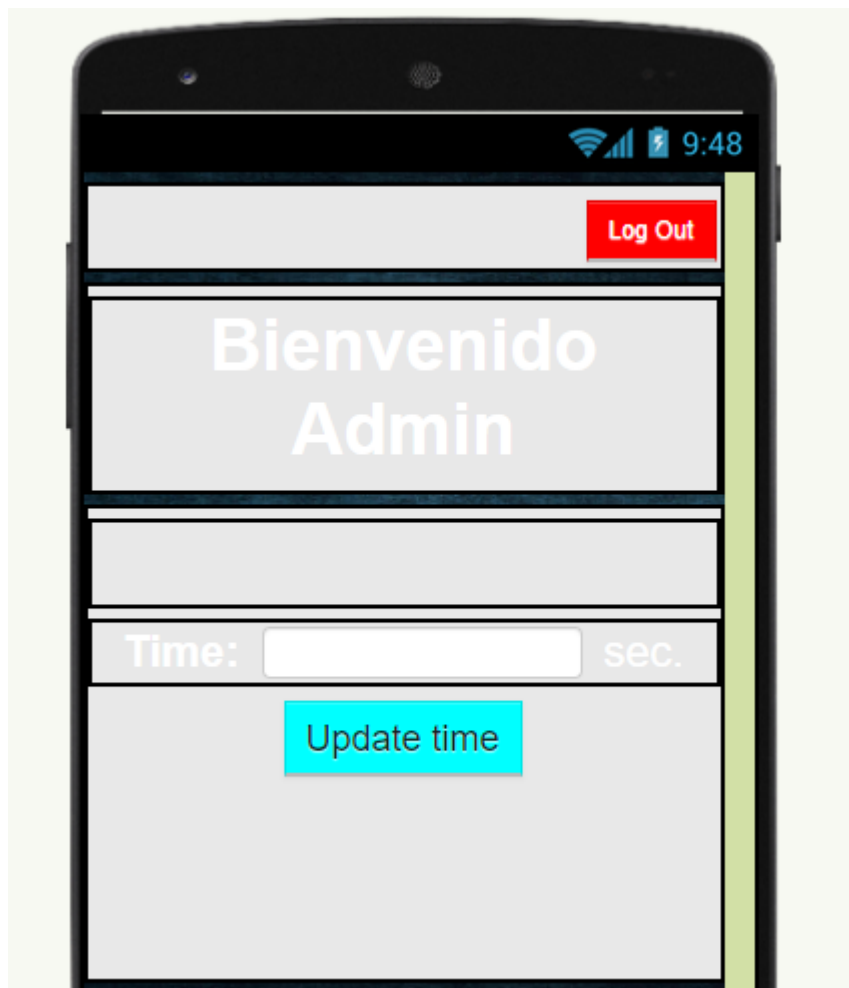


initialize global [contador] to [0]

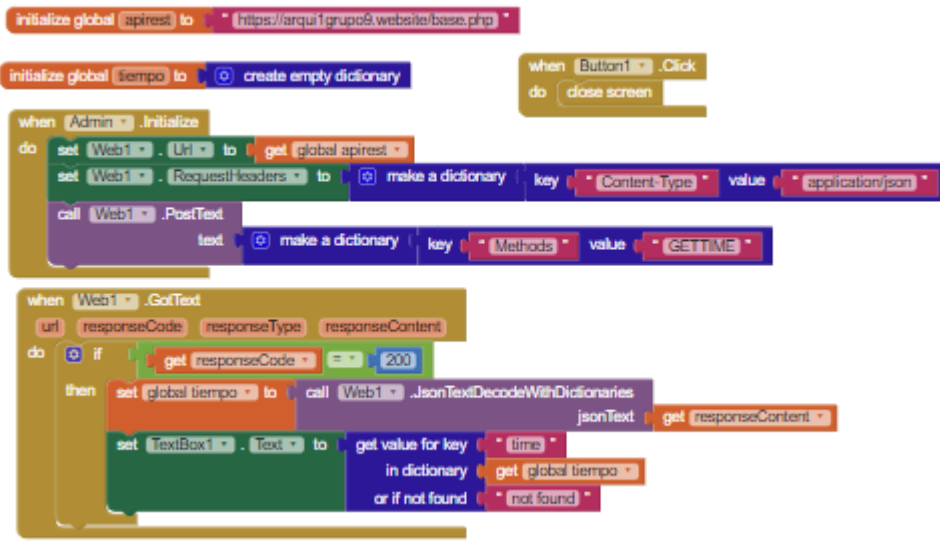




Administrador:







## CODIGO DE SERVICIOS

### CONEXIÓN SERIAL PARA BASE DE DATOS (PYTHON)

```
import tkinter as tk
import serial
import requests
import time
import threading
import json

s = serial.Serial(port='COM2', baudrate=9600, timeout=200,
write_timeout=1)
s2 = serial.Serial(port='COM3', baudrate=9600, timeout=200,
write_timeout=1)

estadoLED1_old = "0"
estadoLED2_old = "0"
estadoLED3_old = "0"
estadoLED4_old = "0"
estadoLED5_old = "0"
estadoLED6_old = "0"
estadoLED7_old = "0"
estadoLED8_old = "0"
estadoLED9_old = "0"
estadoLED10_old = "0"
estadoLED11_old = "0"
estadoLED12_old = "0"
estadoLED13_old = "0"
estadoLED14_old = "0"
estadoLED15_old = "0"
estadoLED16_old = "0"

def hilo():
    global estadoLED1_old
    global estadoLED2_old
    global estadoLED3_old
    global estadoLED4_old
    global estadoLED5_old
    global estadoLED6_old
    global estadoLED7_old
    global estadoLED8_old
    global estadoLED9_old
```

```

global estadoLED10_old
global estadoLED11_old
global estadoLED12_old
global estadoLED13_old
global estadoLED14_old
global estadoLED15_old
global estadoLED16_old

while(True):

    url1 = 'https://arquilapi.000webhostapp.com/base.php'
    args1 = {"Methods": "GETP2"}

    response1 = requests.post(url1, json=args1, timeout=4)

    url = 'https://arquilapi.000webhostapp.com/base.php'
    args = {"Methods": "GETP1"}

    response = requests.post(url, json=args, timeout=4)

    if response.status_code == 200 and response1.status_code==200 :
        payload = response.json()
        payload1=response1.json()
        payload.extend(payload1)
        resultado = payload
        cadena = ""
        estadoLED1 = resultado[0].get('state')

        if estadoLED1!=estadoLED1_old:
            estadoLED1_old = estadoLED1
            if estadoLED1 == "2":
                cadena = "A,2"

        estadoLED2 = resultado[1].get('state')
        if estadoLED2!=estadoLED2_old:
            estadoLED2_old = estadoLED2
            if estadoLED2 == "2":
                if(cadena != ""):
                    cadena += ","
                cadena += "B,2"

        estadoLED3 = resultado[2].get('state')

```

```

if estadoLED3!=estadoLED3_old:
    estadoLED3_old = estadoLED3
    if estadoLED3 == "2":
        if(cadena != ""):
            cadena += ","
        cadena += "C,2"

estadoLED4 = resultado[3].get('state')
if estadoLED4!=estadoLED4_old:
    estadoLED4_old = estadoLED4
    if estadoLED4 == "2":
        if(cadena != ""):
            cadena += ","
        cadena += "D,2"

estadoLED5 = resultado[4].get('state')
if estadoLED5!=estadoLED5_old:
    estadoLED5_old = estadoLED5
    if estadoLED5 == "2":
        if(cadena != ""):
            cadena += ","
        cadena += "E,2"

estadoLED6 = resultado[5].get('state')
if estadoLED6!=estadoLED6_old:
    estadoLED6_old = estadoLED6
    if estadoLED6 == "2" :
        if(cadena != ""):
            cadena += ","
        cadena += "F,2"

estadoLED7 = resultado[6].get('state')
if estadoLED7!=estadoLED7_old:
    estadoLED7_old = estadoLED7
    if estadoLED7 == "2":
        if(cadena != ""):
            cadena += ","
        cadena += "G,2"

```

```

estadoLED8 = resultado[7].get('state')
if estadoLED8!=estadoLED8_old:
    estadoLED8_old = estadoLED8
    if estadoLED8 == "2":
        if(cadena != ""):
            cadena += ","
            cadena += "H,2"

estadoLED9 = resultado[8].get('state')
if estadoLED9!=estadoLED9_old:
    estadoLED9_old = estadoLED9
    if estadoLED9 == "2":
        if(cadena != ""):
            cadena += ","
            cadena += "I,2"

estadoLED10 = resultado[9].get('state')
if estadoLED10!=estadoLED10_old:
    estadoLED10_old = estadoLED10
    if estadoLED10 == "2":
        if(cadena != ""):
            cadena += ","
            cadena += "J,2"

estadoLED11 = resultado[10].get('state')
if estadoLED11!=estadoLED11_old:
    estadoLED11_old = estadoLED11
    if estadoLED11 == "2":
        if(cadena != ""):
            cadena += ","
            cadena += "K,2"

estadoLED12 = resultado[11].get('state')
if estadoLED12!=estadoLED12_old:
    estadoLED12_old = estadoLED12
    if estadoLED12 == "2":
        if(cadena != ""):
            cadena += ","
            cadena += "L,2"

estadoLED13 = resultado[12].get('state')
if estadoLED13!=estadoLED13_old:
    estadoLED13_old = estadoLED13

```

```

        if estadoLED13 == "2":
            if(cadena != ""):
                cadena += ","
            cadena += "M,2"

estadoLED14 = resultado[13].get('state')
if estadoLED14!=estadoLED14_old:
    estadoLED14_old = estadoLED14
    if estadoLED14 == "2":
        if(cadena != ""):
            cadena += ","
        cadena += "N,2"

estadoLED15 = resultado[14].get('state')
if estadoLED15!=estadoLED15_old:
    estadoLED15_old = estadoLED15
    if estadoLED15 == "2":
        if(cadena != ""):
            cadena += ","
        cadena += "O,2"

estadoLED16 = resultado[15].get('state')
if estadoLED16!=estadoLED16_old:
    estadoLED16_old = estadoLED16
    if estadoLED16 == "2":
        if(cadena != ""):
            cadena += ","
        cadena += "P,2"

if (cadena != ""):
    print(cadena)
    s.write(cadena.encode())

time.sleep(2)

def lectura():
    while True:
        time.sleep(2)
        try:

```

```

        raw_string_b = s.readline()
        raw_string_s = raw_string_b.decode('utf-8')
        print(raw_string_s)
        parametros=raw_string_s.split(",")
        url = 'https://arquilapi.000webhostapp.com/base.php'
                args = {"Methods": parametros[0], "estado" :
parametros[1], "id":parametros[2]}

        response = requests.post(url, json=args)
        payload = response.json()
        #print(payload)

except:
    pass
    #print("fallo")

def consultarlcd():

    while(True):

        url = 'https://arquilgrupo9.website/base.php'
        args = {"Methods": "GETNUMP12"}

        response = requests.post(url, json=args, timeout=4)

        cadena=""

        if response.status_code == 200 :
            payload = response.json()

            ocupado = payload.get('ocupados')
            res = payload.get('reservados')
            tot = ocupado + res
            cadena=" ,"
            if tot==16:
                cadena += "FULL, "
            elif ocupado==16:

```

```

        cadena += "FULL, "
    else:
        cadena += "FREE, "

    cadena += str(payload.get('disponibles')) + ", "
    cadena += str(payload.get('ocupados')) + ", "
    cadena += str(payload.get('reservados'))

    if (cadena != ""):
        print(cadena)
        s.write(cadena.encode())

    time.sleep(5)

def encender():
    t=threading.Thread(target=hilo,args=())
    t.start()
    t2=threading.Thread(target=lectura,args=())
    t2.start()

    t3=threading.Thread(target=consultarlcd,args=())
    t3.start()

sc = tk.Tk()

sc.config(width=300, height=180)

sc.title("Control de Led")

b1 = tk.Button(text="On", command=encender)
b1.place(x=50, y=10)

sc.mainloop()

```



## BASE DE DATOS (PHP)

```
<?php
require_once './Conexion.php';
$conexion = new Conexion();
$conectar = $conexion->Conectar();

$body = json_decode(file_get_contents("php://input"), true);
$data = array();

if ($body['Methods'] != '') {
    switch ($body['Methods']) {
        case 'POST':
            // $data["usuario"] = $body['usuario'];
            // $data["password"] = $body['password'];

            $consulta2 = "SELECT * FROM `usuarios` WHERE `user` LIKE
'{$body['usuario']}'";
            $resultado2 = $conectar->prepare($consulta2);
            $resultado2->execute();

            if ($resultado2->rowCount() >= 1) {
                $data["msg"] = "The user already exists.";
            } else {
                $consulta = "INSERT INTO `usuarios` (`id`, `user`,
`password`, `pos`) VALUES (NULL, '{$body['usuario']}',
'{$body['password']}', '{$body['pos']}')";
                $resultado = $conectar->prepare($consulta);
                $resultado->execute();
                $data["msg"] = "The user was added.";
            }

            echo json_encode($data);
            break;

        case 'GET':
            $usuario = $body['usuario'];
            $clave = $body['password'];
            $consulta = "SELECT * FROM `usuarios` WHERE `user` LIKE
'{$usuario}' AND `password` LIKE '{$clave}'";
            $resultado = $conectar->prepare($consulta);
```

```

$resultado->execute();

if ($resultado->rowCount() >= 1) {
    if($usuario == "admin"){
        $data["resultado"] = 2;
    }else{
        $data["resultado"] = 1;
    }

} else {
    $data["resultado"] = 0;
}

echo json_encode($data);
break;

case 'GETP1':
    $consulta = "SELECT * FROM `Parking 1`";
    $ejecutar = $conectar->prepare($consulta);
    $ejecutar->execute();
    $resultado = $ejecutar->fetchall(PDO::FETCH_OBJ);
    echo json_encode($resultado);
    break;

case 'GETP2':
    $consulta = "SELECT * FROM `Parking 2`";
    $ejecutar = $conectar->prepare($consulta);
    $ejecutar->execute();
    $resultado = $ejecutar->fetchall(PDO::FETCH_OBJ);
    echo json_encode($resultado);
    break;

case 'GETNUMP1':
    $consulta = "SELECT COUNT(*) FROM `Parking 1` WHERE `state`
LIKE '0'";
    $ejecutar = $conectar->prepare($consulta);
    $ejecutar->execute();
    $resultado = $ejecutar->fetch();
    $data["disponibles"] = $resultado[0];
    $consulta = "SELECT COUNT(*) FROM `Parking 1` WHERE `state`
LIKE '1'";
    $ejecutar = $conectar->prepare($consulta);
    $ejecutar->execute();
    $resultado = $ejecutar->fetch();

```

```

        $data["ocupados"] = $resultado[0];
        $consulta = "SELECT COUNT(*) FROM `Parking 1` WHERE `state`
LIKE '2'";
        $ejecutar = $conectar->prepare($consulta);
        $ejecutar->execute();
        $resultado = $ejecutar->fetch();
        $data["reservados"] = $resultado[0];
        echo json_encode($data);
        break;

    case 'GETNUMP2':
        $consulta = "SELECT COUNT(*) FROM `Parking 2` WHERE `state`
LIKE '0'";
        $ejecutar = $conectar->prepare($consulta);
        $ejecutar->execute();
        $resultado = $ejecutar->fetch();
        $data["disponibles"] = $resultado[0];
        $consulta = "SELECT COUNT(*) FROM `Parking 2` WHERE `state`
LIKE '1'";
        $ejecutar = $conectar->prepare($consulta);
        $ejecutar->execute();
        $resultado = $ejecutar->fetch();
        $data["ocupados"] = $resultado[0];
        $consulta = "SELECT COUNT(*) FROM `Parking 2` WHERE `state`
LIKE '2'";
        $ejecutar = $conectar->prepare($consulta);
        $ejecutar->execute();
        $resultado = $ejecutar->fetch();
        $data["reservados"] = $resultado[0];
        echo json_encode($data);
        break;

    case 'UPDATEP1':
        $consulta = "UPDATE `Parking 1` SET `state` = '2', `user` =
'{$body['usuario']}' WHERE `Parking 1`.`id` = {$body['id']}";
        $resultado = $conectar->prepare($consulta);
        $resultado->execute();
        $data["msg"] = "Parking reserved";
        echo json_encode($data);
        break;

    case 'UPDATEP2':

```

```

        $consulta = "UPDATE `Parking 2` SET `state` = '2', `user` =
'{$body['usuario']}' WHERE `Parking 2`.`id` = {$body['id']}";

        $resultado = $conectar->prepare($consulta);
        $resultado->execute();
        $data["msg"] = "Parking reserved";
        echo json_encode($data);
        break;

    case 'RESERVA':
        $usuario = $body['usuario'];
        $consulta = "SELECT * FROM `Parking 1` WHERE `user` LIKE
'{$usuario}'";
        $resultado = $conectar->prepare($consulta);
        $resultado->execute();

        if ($resultado->rowCount() >= 1) {
            $resultado2 = $resultado->fetch(PDO::FETCH_BOTH);
            $data["id"] = $resultado2[0] + 1;
            $data["ocupado"] = true;
            $data["nivel"] = 1;

        } else {
            $consulta = "SELECT * FROM `Parking 2` WHERE `user`
LIKE '{$usuario}'";
            $resultado = $conectar->prepare($consulta);
            $resultado->execute();

            if ($resultado->rowCount() >= 1) {
                $resultado2 = $resultado->fetch(PDO::FETCH_BOTH);
                $data["id"] = $resultado2[0] + 1;
                $data["ocupado"] = true;
                $data["nivel"] = 2;

            } else {
                $data["id"] = 0;
                $data["ocupado"] = false;
                $data["nivel"] = 0;
            }
        }
        echo json_encode($data);

        break;

```

```

        case 'GETTIME':
            $consulta = "SELECT * FROM `Tiempo` WHERE `id` = 0";
            $resultado = $conectar->prepare($consulta);
            $resultado->execute();
            $resultado2 = $resultado->fetchAll();
            $data["time"] = $resultado2[0]['time'];
            echo json_encode($data);

        break;

        case 'UPDATETIME':
            $consulta = "UPDATE `Tiempo` SET `time` = '{$body['time']}'
WHERE `Tiempo`.`id` = 0";
            $resultado = $conectar->prepare($consulta);
            $resultado->execute();
            $data["msg"] = "The time was updated";
            echo json_encode($data);

        break;

        case 'UPDATEP1_PROT':
            $consulta = "UPDATE `Parking 1` SET `state` =
 '{$body['estado']}', `user` = '' WHERE `Parking 1`.`id` =
 {$body['id']}";
            $resultado = $conectar->prepare($consulta);
            $resultado->execute();
            $data["msg"] = "Estado actualizado.";
            echo json_encode($data);

        break;

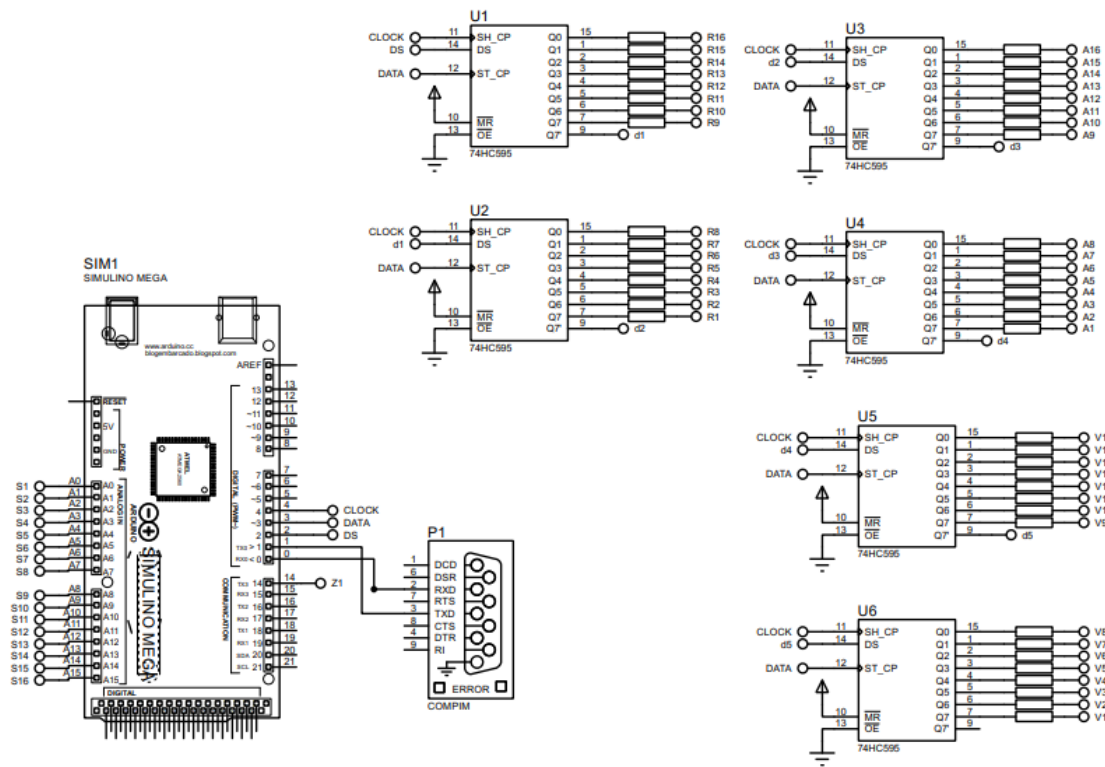
        case 'UPDATEP2_PROT':
            $consulta = "UPDATE `Parking 2` SET `state` =
 '{$body['estado']}', `user` = '' WHERE `Parking 2`.`id` =
 {$body['id']}";
            $resultado = $conectar->prepare($consulta);
            $resultado->execute();
            $data["msg"] = "Estado actualizado.";
            echo json_encode($data);

        break;
    }
} else {
    $data = ['error' => 'No hay Methods'];
}
?>

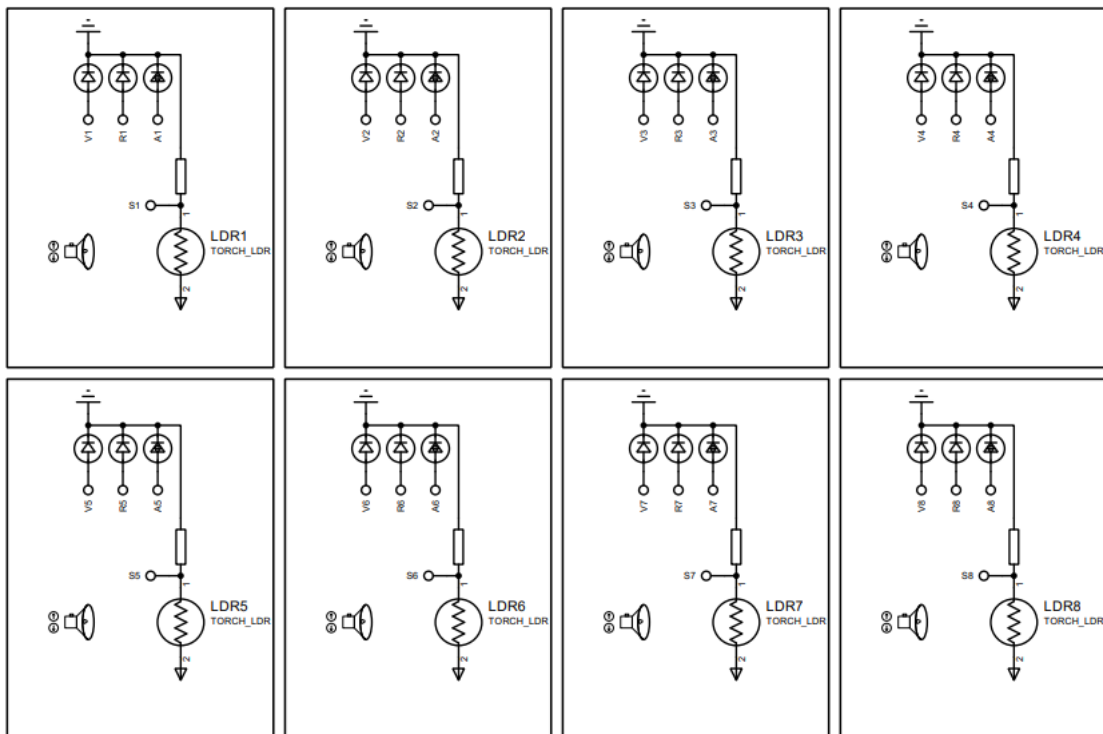
```

# DIAGRAMA DEL CIRCUITO

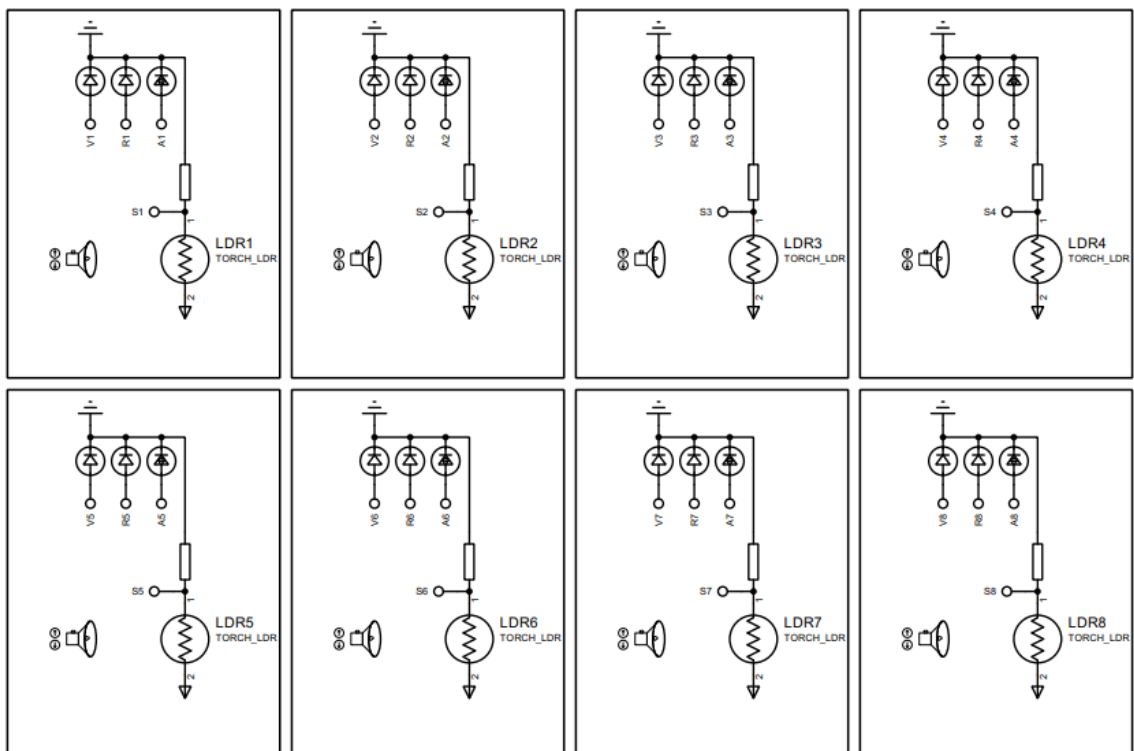
## Diagrama De Los Niveles



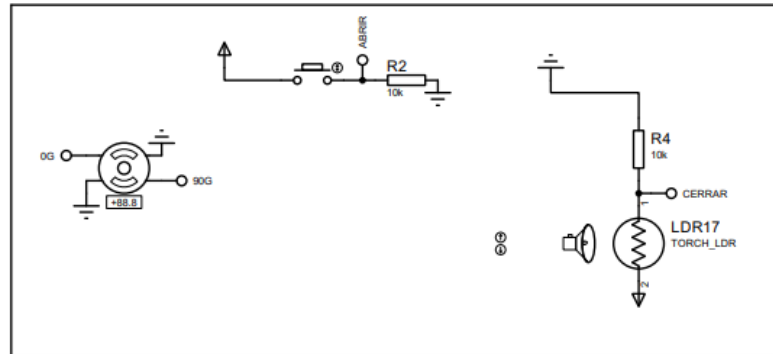
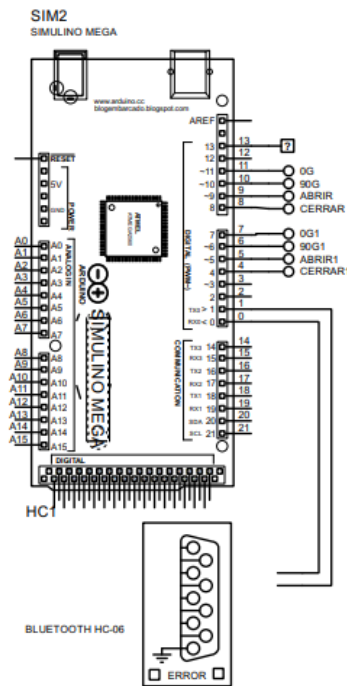
# PARQUEO 1



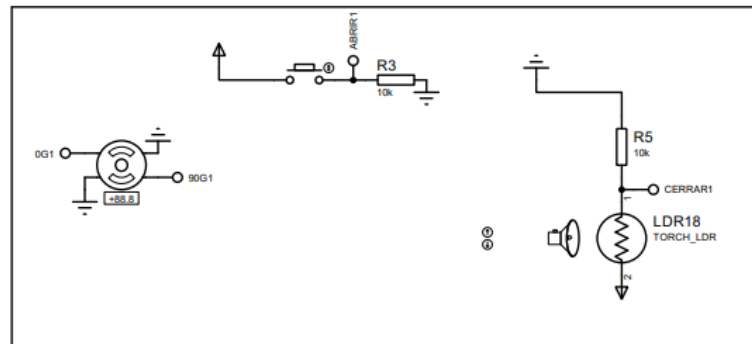
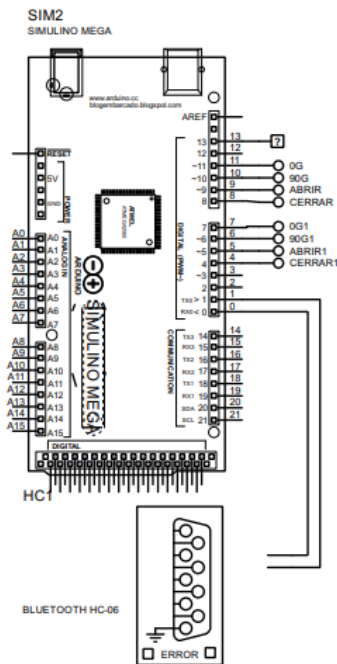
# PARQUEO 2



# Barra De Entrada

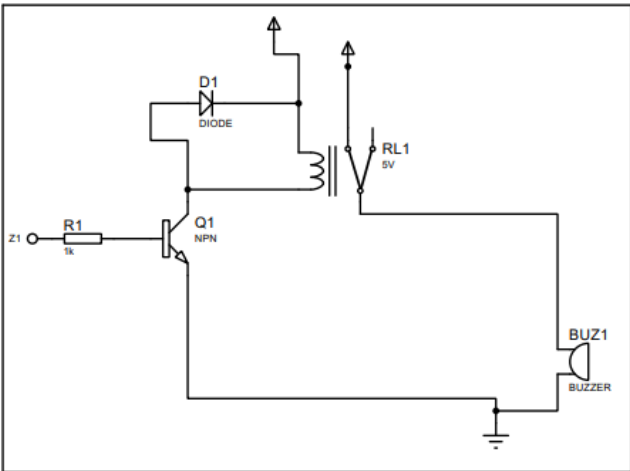
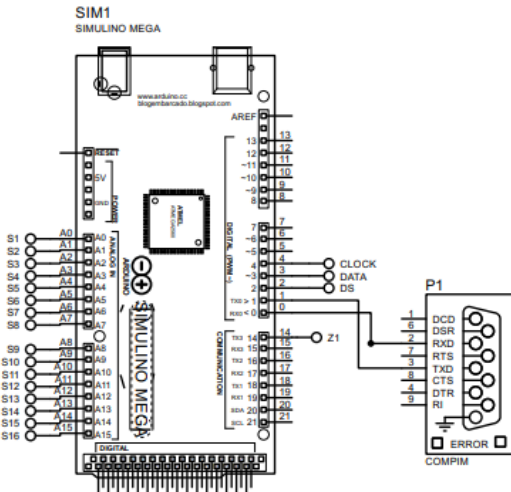


# Barra De Salida

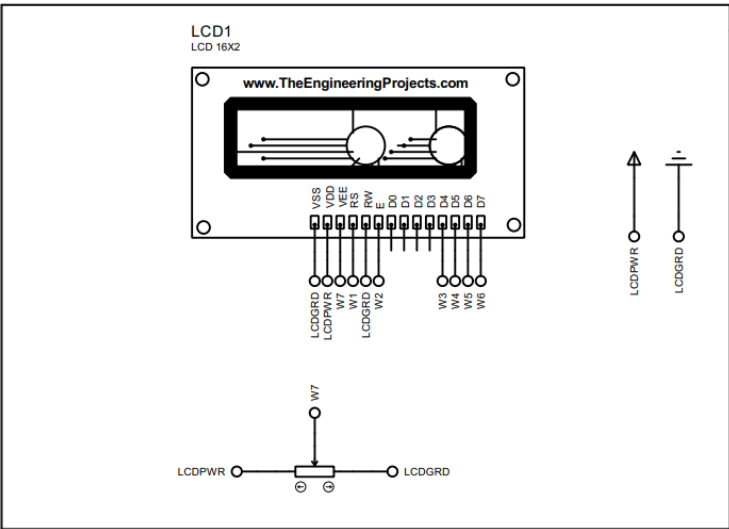
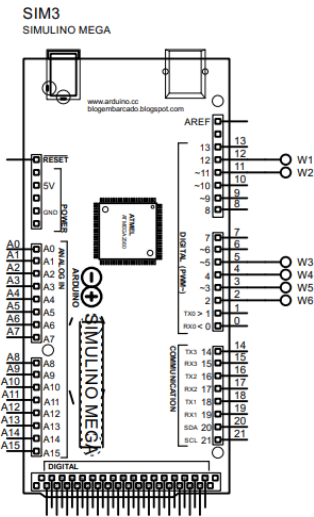




# Alarma



# PANTALLA LCD



## **DESCRIPCIÓN HERRAMIENTAS UTILIZADAS**

### **Python:**

Python es un lenguaje de programación ampliamente utilizado en las aplicaciones web, el desarrollo de software, la ciencia de datos y el machine learning (ML). Los desarrolladores utilizan Python porque es eficiente y fácil de aprender, además de que se puede ejecutar en muchas plataformas diferentes. El software Python se puede descargar gratis, se integra bien a todos los tipos de sistemas y aumenta la velocidad del desarrollo.

### **Pyserial:**

PySerial es una librería de python que permite comunicarse a través de comunicaciones por serial (RS-232).

Esto puede ser muy útil para mandar o recibir datos de periféricos que sean comunes o que tú mismo hayas hecho de una manera increíblemente sencilla y sin tener que complicarse para nada con este tipo de programación.

### **Hostinger:**

Hostinger International, Ltd. es un proveedor de alojamiento web y dominios web. Fundada en el año 2004, Hostinger ahora posee más de 30 millones de usuarios, en conjunto con filiales en 178 países. La empresa ofrece servicios de alojamiento web compartido, en la nube y VPS con MySQL, FTP y PHP así como una fácil integración con WordPress, Shopify, Drupal, etc.

Hostinger es una compañía asociada con 000Webhost, Niagahoster y Weblink.

**PHP:**

Es un lenguaje de programación destinado a desarrollar aplicaciones para la web y crear páginas web, favoreciendo la conexión entre los servidores y la interfaz de usuario. Entre los factores que hicieron que PHP se volviera tan popular, se destaca el hecho de que es de código abierto.