



UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS
CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES

TEMA

“APLICATIVO MÓVIL PARA LA DETECCIÓN DE PLAZAS DE APARCAMIENTO EN LA UNIVERSIDAD TÉCNICA DEL NORTE, DESARROLLADO PARA DISPOSITIVOS ANDROID.”

Autor: Cristhian Patricio Vásquez Demera

Director: Ing. Diego Javier Trejo España

Ibarra – Ecuador

2017

MANUAL TÉCNICO

Primero desarrollamos un pequeño código programado en el software de arduino para poder realizar la conexión con los sensores, que son los que permiten detectar las plazas de parqueo.



```
#include <Arduino.h>
#include <ESP8266WiFi.h>
#include <ESP8266WiFiMulti.h>
#include <ESP8266HTTPClient.h>

ESP8266WiFiMulti WiFiMulti;
int s1, s2, s3;
int sensor1, sensor2, sensor3;
boolean envio = 0;
void setup() {

  Serial.begin(115200);
  pinMode(D0, INPUT);
  pinMode(D1, INPUT);
  pinMode(D2, INPUT);

  for (uint8_t t = 4; t > 0; t--)
  {
    Serial.printf("[SETUP] WAIT %d...\n", t);
    Serial.flush();
    delay(1000);
  }
  WiFiMulti.addAP("WUTN.Eventos", "#EventosUtn@2015");
  s1 = digitalRead(D0);
  s2 = digitalRead(D1);
```

```

s3 = digitalRead(D2);
s1 = sensor1;
s2 = sensor2;
s3 = sensor3;
}

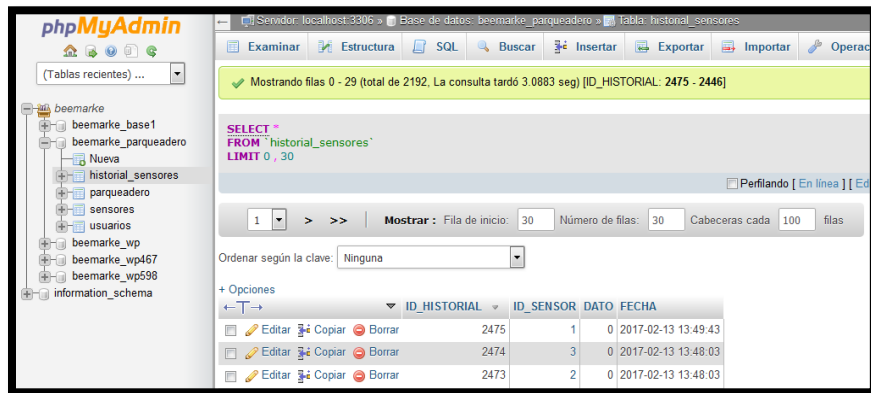
void loop() {
  sensor1 = digitalRead(D0);
  sensor2 = digitalRead(D1);
  sensor3 = digitalRead(D2);
  if (s1 != sensor1 || s2 != sensor2 || s3 != sensor3)
  {
    s1 = sensor1;
    s2 = sensor2;
    s3 = sensor3;
    // wait for WiFi connection
    if ((WiFiMulti.run() == WL_CONNECTED))
    {
      HTTPClient http;
      String url =
"http://www.beemarketing.com.ec/parqueadero/scriptBDD/add.php?sensor1=" +
String(sensor1) + "&sensor2=" + String(sensor2) + "&sensor3=" + String(sensor3);
      Serial.println(url);
      http.begin(url);

      int httpCode = http.GET();
      if (httpCode > 0)
      {
        Serial.printf("[HTTP] GET... code: %d\n", httpCode);
        if (httpCode == HTTP_CODE_OK)
        {
          String payload = http.getString();
          Serial.println(payload);
        }
      }
    }
    else
    {
      Serial.printf("[HTTP] GET... failed, error: %s\n", http.errorToString(httpCode).c_str());
    }
    http.end();
  }
}

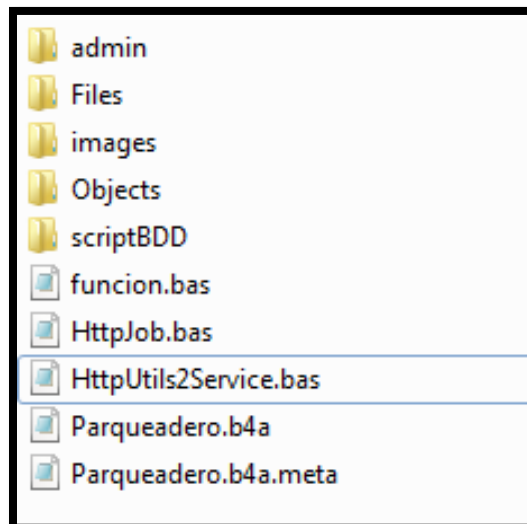
```


```
}  
delay(10);  
}
```

El código desarrollado en arduino permite enviar la señal de cada sensor a la base de datos Mysql. En la base de datos Mysql se encuentra la base llamada beemarke_parqueadero y en la tabla historial sensores es donde se guarda el cambio de estado, dentro del campo dato; que puede ser 1 o 0.



Ahora, para realizar la conexión con el programa Basic4Android utilizamos dos Servicios llamado **HttpUtils2Service** y **HttpJob** que son los que permiten hacer la conexión a la base de datos Mysql que se encuentra en el servidor.



 HttpUtils2Service.bas**Código:**

```
Sub Process_Globals
    Private hc As HttpClient
    Private TaskIdToJob As Map
    Public TempFolder
    Private taskCounter As Int
End Sub

Sub Service_Create
    TempFolder = File.DirInternalCache
    hc.Initialize("hc")
    TaskIdToJob.Initialize
End Sub

Public Sub SubmitJob(job As HttpJob) As Int
    taskCounter = taskCounter + 1
    TaskIdToJob.Put(taskCounter, job)
    If job.Username <> "" AND job.Password <> "" Then
        hc.ExecuteCredentials(job.GetRequest, taskCounter, job.Username,
job.Password)
    Else
        hc.Execute(job.GetRequest, taskCounter)
    End If
    Return taskCounter
End Sub

Sub hc_ResponseSuccess (Response As HttpResponse, TaskId As Int)
    Response.GetAsyncynchronously("response", File.OpenOutput(TempFolder, TaskId,
False), _
        True, TaskId)
End Sub

Sub Response_StreamFinish (Success As Boolean, TaskId As Int)
    If Success Then
        CompleteJob(TaskId, Success, "")
    Else
        CompleteJob(TaskId, Success, LastException.Message)
    End If
End Sub
```

```

        End If
    End Sub

    Sub hc_ResponseError (Response As HttpResponse, Reason As String, StatusCode As Int,
        TaskId As Int)
        If Response <> Null Then
            Try
                Log(Response.GetString("UTF8"))
            Catch
                Log("Failed to read error message.")
            End Try
            Response.Release
        End If
        CompleteJob(TaskId, False, Reason)
    End Sub

    Sub CompleteJob(TaskId As Int, success As Boolean, errorMessage As String)
        Dim job As HttpJob
        job = TaskIdToJob.Get(TaskId)
        TaskIdToJob.Remove(TaskId)
        job.success = success
        job.errorMessage = errorMessage
        job.Complete(TaskId)
    End Sub

```



Código:

```

Sub Class_Globals
    Public JobName As String
    Public Success As Boolean
    Public Username, Password As String
    Public ErrorMessage As String
    Private target As Object
    Private mLink As String
    Private taskId As String
    Private req As HttpRequest
End Sub

```

'Initializes the Job.

'Name - The job's name. Note that the name doesn't need to be unique.

'TargetModule - The activity or service that will handle the JobDone event.

```
Public Sub Initialize (Name As String, TargetModule As Object)
```

```
    JobName = Name
```

```
    target = TargetModule
```

```
End Sub
```

'Sends a POST request with the given data as the post data.

```
Public Sub PostString(Link As String, Text As String)
```

```
    PostBytes(Link, Text.GetBytes("UTF8"))
```

```
End Sub
```

'Sends a POST request with the given string as the post data

```
Public Sub PostBytes(Link As String, Data() As Byte)
```

```
    mLink = Link
```

```
    req.InitializePost2(Link, Data)
```

```
    CallSubDelayed2(HttpUtils2Service, "SubmitJob", Me)
```

```
End Sub
```

'Sends a POST request with the given file as the post data.

'This method doesn't work with assets files.

```
Public Sub PostFile(Link As String, Dir As String, FileName As String)
```

```
    Dim length As Int
```

```
    If Dir = File.DirAssets Then
```

```
        Log("Cannot send files from the assets folder.")
```

```
        Return
```

```
    End If
```

```
    length = File.Size(Dir, FileName)
```

```
    Dim In As InputStream
```

```
    In = File.OpenInput(Dir, FileName)
```

```
    If length < 1000000 Then '1mb
```

```
        'There are advantages for sending the file as bytes array. It allows the Http  
library to resend the data
```

```
        'if it failed in the first time.
```

```
        Dim out As OutputStream
```

```
        out.InitializeToByteArray(length)
```

```
        File.Copy2(In, out)
```

```
        PostBytes(Link, out.ToByteArray)
```

```
    Else
```

```
        req.InitializePost(Link, In, length)
```

```
        CallSubDelayed2(HttpUtils2Service, "SubmitJob", Me)
```

```

        End If
    End Sub
    'Submits a HTTP GET request.
    'Consider using Download2 if the parameters should be escaped.
    Public Sub Download(Link As String)
        mLink = Link
        req.InitializeGet(Link)
        CallSubDelayed2(HttpUtils2Service, "SubmitJob", Me)
    End Sub
    'Submits a HTTP GET request.
    'Encodes illegal parameter characters.
    '<code>Example:
    'job.Download2("http://www.example.com", _
    '    Array As String("key1", "value1", "key2", "value2"))</code>
    Public Sub Download2(Link As String, Parameters() As String)
        mLink = Link
        Dim sb As StringBuilder
        sb.Initialize
        sb.Append(Link)
        If Parameters.Length > 0 Then sb.Append("?")
        Dim su As StringUtils
        For i = 0 To Parameters.Length - 1 Step 2
            If i > 0 Then sb.Append("&")
            sb.Append(su.EncodeUrl(Parameters(i), "UTF8")).Append("=")
            sb.Append(su.EncodeUrl(Parameters(i + 1), "UTF8"))
        Next
        req.InitializeGet(sb.ToString)
        CallSubDelayed2(HttpUtils2Service, "SubmitJob", Me)
    End Sub

    'Called by the service to get the request
    Public Sub GetRequest As HttpRequest
        Return req
    End Sub

    'Called by the service when job completes
    Public Sub Complete (id As Int)
        taskId = id
        CallSubDelayed2(target, "JobDone", Me)
    End Sub

```


'Should be called to free resources held by this job.

```
Public Sub Release
    File.Delete(HttpUtils2Service.TempFolder, taskId)
End Sub
```

'Returns the response as a string encoded with UTF8.

```
Public Sub GetString As String
    Return GetString2("UTF8")
End Sub
```

'Returns the response as a string.

```
Public Sub GetString2(Encoding As String) As String
    Dim tr As TextReader
    tr.Initialize2(File.OpenInput(HttpUtils2Service.TempFolder, taskId), Encoding)
    Dim res As String
    res = tr.ReadAll
    tr.Close
    Return res
End Sub
```

'Returns the response as a bitmap

```
Public Sub GetBitmap As Bitmap
    Dim b As Bitmap
    b = LoadBitmap(HttpUtils2Service.TempFolder, taskId)
    Return b
End Sub
```

Sub GetInputStream As InputStream

```
    Dim In As InputStream
    In = File.OpenInput(HttpUtils2Service.TempFolder, taskId)
    Return In
End Sub
```

Para el control y registro de usuarios utilizamos otro servicio, llamado **funcion**.



Sub Process_Globals

'These global variables will be declared once when the application starts.

'These variables can be accessed from all modules.

'registro

Dim reg1 As HttpClient

Dim reg2 As HttpClient

Dim registro_list_usuarios As Int

registro_list_usuarios = 1

Dim registro As Boolean

Dim registro_ok As Boolean

registro=False

'autenticacion

Dim hc_login As HttpClient

Dim hc_login2 As HttpClient

Dim login_list_usuarios As Int

login_list_usuarios = 1

Dim login As Boolean

Dim login_id As String

Dim login_ok As Boolean

login=False

'edicion

Dim hc_modificar As HttpClient

Dim modificar_list_usuarios As Int

modificar_list_usuarios = 1

Dim modificar As Boolean

modificar=False

'control de conexion

Dim myip As String

```
Dim Server As ServerSocket
myip=Server.GetMyIP
```

'-----**Modificar el nombre de la página web y el nombre de la tabla**-----'

```
'consulta
Dim str_query As String
Dim linkdb As String
str_query="usuarios" nombre de la tabla
linkdb="http://www.beemarketing.com.ec/parqueadero/scriptBDD/B4A.php"link
de la pagina php
'linkdb="http://parqueadero.000webhostapp.com/B4A.php"link de la pagina php
```

```
'clave simple
Dim Claveapp As String
Claveapp="bXZhb9tb250aUBsaWJlcm8uaXb="
```

```
'Variable para el indicar el Activity_Create !!
'Eliminar los registros de lectura y verificación de texto y número
```

```
Dim modo_programador As Boolean
modo_programador=True
```

'-----
End Sub

```
Public Sub WriteIni(mykey As String,myvalue As String,fPath As String,filename As
String)As String
Dim Map1 As Map
Map1.Initialize
If File.Exists(fPath,filename) Then
Else
File.WriteMap(fPath, filename, Map1) 'para crearlo si no existe
End If
Map1 = File.ReadMap(fPath,filename)
Map1.Put(mykey, myvalue)
File.WriteMap(fPath, filename, Map1)
End Sub
```

Los 3 servicios descritos anteriormente son los que nos permiten trabajar directamente con mysql en cuanto al control de sensores y registro de usuarios.

A continuación, se detalla el código del Main, donde se realiza el diseño del programa en Basic4Android.

Sub Process_Globals

'These global variables will be declared once when the application starts.

'These variables can be accessed from all modules.

Dim sensores As HttpClient

'Esta variable permite control el tiempo en milisegundos de la transmisión de datos de cada sensor.

Dim Timer1 As Timer

Dim comp As String

End Sub

Sub Globals

'These global variables will be redeclared each time the activity is created.

'These variables can only be accessed from this module.

'Panel1

Private Panel1 As Panel

Private Panel2 As Panel

Private Button_entrar As Button

Private EditText_nombre As EditText

Private EditText_mail As EditText

Private Button_iniciar_sesion As Button

Private EditText_password As EditText

Private EditText_lugar As EditText

Private Button_editar As Button

Private EditText_verificacion As EditText

Private Label_verificacion As Label

Private Label_nombre As Label

Private Label_email As Label

Private Label_password As Label

Private Label_lugar As Label

Private Label_verifica_spam As Label

Private Label_name As Label

'teclado

Dim ocultar_teclado As IME

'Log

Private Button_cerrar_sesion As Button

Private Button_log As Button

'menu

Private Button_menu As Button

Private CheckBox_password As CheckBox

'Boton ir

Private Button_ir As Button

'Panel2

Private Label_0 As Label

Private Label_1 As Label

Private Label_2 As Label

Private Label_3 As Label

Private Label_4 As Label

Private Label_5 As Label

Private Button_layout2 As Button

Private Label_6 As Label

Private Label_7 As Label

Dim EditText1, EditText2 As EditText

'Private WebView1 As WebView

'Sensores

Private lblsensor1 As Label

Private lblsensor2 As Label

Private lblsensor3 As Label

Private ImageView_Carro As ImageView

Private ImageView_Carro1 As ImageView

Private ImageView_Carro2 As ImageView

'Variables para control de los sensores

Dim id, s1, s2, s3, e1, e2, e3 As String

Private ImageView_Activado As ImageView

```

Private ImageView_Activado1 As ImageView
Private ImageView_Activado2 As ImageView
End Sub

Sub Activity_Create(FirstTime As Boolean)
'Do not forget to load the layout file created with the visual designer. For example:
'Activity.LoadLayout("Layout1")
If FirstTime Then
    Activity.LoadLayout("Layout1")
    funcion.reg1.Initialize("reg1")
    funcion.reg2.Initialize("reg2")
    funcion.hc_login.Initialize("hc_login")
    funcion.hc_login2.Initialize("hc_login2")
    funcion.hc_modificar.Initialize("hc_modificar")
    Panel1.SetBackgroundImage(LoadBitmap(File.DirAssets,"1.jpg"))
    Panel1.SetLayout(0,0,100%x,100%y)
    Panel2.SetLayout(0,0,100%x,100%y)
    Activity.AddMenuItem("<< Atrás","mnuPage2")
    Activity.AddMenuItem("Login","mnuPage3")
    Activity.AddMenuItem("Modificar mis datos","mnuPage1")
    Activity.AddMenuItem("Registrarse","mnuPage4")
    EditText_nombre.Hint = "Nombre"
    EditText_password.Hint = "Password"
End If
load_config
End Sub

Sub Button_ir_Click
Panel1.visible=False
Panel2.visible=True
Button_menu.Visible=False
'-----Tiempo de respuesta de cada sensor-----

Timer1.Initialize("Timer1", 5000) ' 1000 = 1 segundo
Timer1.Enabled = True
'-----

FetchCountriesList

```

```
'      WebView1.Width=100%x
'      WebView1.Height=100%y
'      WebView1.Top=0
'      WebView1.Left=0
'      WebView1.LoadUrl("http://192.168.1.14")
```

End Sub

```
Sub Button_layout2_Click
    Panel2.visible=False
    Panel1.visible=True
    Timer1.Enabled=False
    Button_menu.Visible=True
```

End Sub

```
Sub Button_menu_Click
    ocultar_teclado.HideKeyboard
    Activity.OpenMenu()
```

End Sub

```
Sub Button_log_Click
    mnuPage3_Click
```

End Sub

```
Sub Button_cerrar_sesion_Click
    funcion.login_ok=False
    funcion.minombre=""
    Button_ir.Visible=False
    Button_cerrar_sesion.Visible=False
    Button_log.Visible=True
    If CheckBox_password.Checked=False Then
        EditText_password.Text=""
    End If
    mnuPage2_Click
```

End Sub

```
Sub mnuPage1_Click 'Modificar Datos
    Button_ir.Visible=False
    If funcion.login_ok=True Then
        Button_cerrar_sesion.Visible=True
        Button_log.Visible=False
        CheckBox_password.Visible=False
```

```
        verifica_spam
        If funcion.modos_programador=False Then
            EditText_password.text=""
        End If
        Panel1.Visible=True
        EditText_nombre.Visible=False
        Label_name.Visible=True
        Label_name.text=EditText_nombre.text
        EditText_password.Visible=True
        Button_entrar.Visible=False
        Button_iniciar_sesion.Visible=False
        Button_editar.Visible=True
            EditText_verificacion.Visible=True
        Label_verificacion.Visible=True
        Label_nombre.visible=True
        Label_password.visible=True
        Label_verifica_spam.visible=True
        Else
            Button_log.Visible=True
        Button_cerrar_sesion.Visible=False
            MsgBox("Primero debe iniciar sesión!", "")
        End If
    End Sub
```

Sub mnuPage2_Click' ----- Atrás

```
    If funcion.login_ok=True Then
        Button_ir.Visible=True
        Button_cerrar_sesion.Visible=True
        Button_log.Visible=False
    Else
        Button_ir.Visible=False
        Button_log.Visible=True
        Button_cerrar_sesion.Visible=False
    End If
        CheckBox_password.Visible=False
        Panel1.Visible=True

        EditText_nombre.Visible=False
        EditText_nombre.Enabled=True
        Label_name.Visible=False
```



```
    EditText_password.Visible=False
    Button_entrar.Visible=False
    Button_iniciar_sesion.Visible=False
    Button_editar.Visible=False
    EditText_verificacion.Visible=False
    Label_verificacion.Visible=False
    Label_nombre.visible=False
    Label_password.visible=False
    Label_verifica_spam.visible=False
End Sub
```

Sub mnuPage3_Click' Login

```
If Button_ir.Visible=True Then
    Button_ir.Visible=True
Else
    Button_ir.Visible=False
End If
If funcion.login_ok=True Then
    Button_cerrar_sesion.Visible=True
    Button_log.Visible=False
    MsgBox("Ya se ha logeado!", "")
Else
    Button_log.Visible=True
    Button_cerrar_sesion.Visible=False
    CheckBox_password.Visible=True
    Panel1.Visible=True
    EditText_nombre.Visible=True
    EditText_nombre.Enabled=True
    Label_name.Visible=False
    EditText_password.Visible=True
    Button_entrar.Visible=False
    Button_iniciar_sesion.Visible=True
    Button_editar.Visible=False
    EditText_verificacion.Visible=False
    Label_verificacion.Visible=False
    Label_nombre.visible=True
    Label_password.visible=True
    Label_verifica_spam.visible=False
End If
End Sub
```

Sub mnuPage4_Click' Registro

```
If Button_ir.Visible=True Then
Button_ir.Visible=True
Else
Button_ir.Visible=False
End If
If funcion.login_ok=True Then
    Button_cerrar_sesion.Visible=True
    Button_log.Visible=False
    MsgBox("Ya se ha registrado!", "")
    Else
    CheckBox_password.Visible=False
    verifica_spam
    If funcion.modos_programador=False Then
    EditText_nombre.Text=""
    EditText_password.Text=""
    End If
    Panel1.Visible=True
    EditText_nombre.Visible=True
    EditText_nombre.Enabled=True
    Label_name.Visible=False
    EditText_password.Visible=True
    Button_entrar.Visible=True
    Button_entrar.Enabled=True
    Button_iniciar_sesion.Visible=False
    Button_editar.Visible=False
    EditText_verificacion.Visible=True
    Label_verificacion.Visible=True
    Label_nombre.visible=True
    Label_password.visible=True
    Label_verifica_spam.visible=True
    End If
End Sub
```

'Encriptacion de datos

```
Sub Encrypt(dataToEncrypt As String ) As String

    Dim kg As KeyGenerator
    Dim c As Cipher
    Dim B64 As Base64
```

```

    Dim bconv As ByteConverter
    Dim data(0) As Byte
    Dim iv(0) As Byte
    iv = Array As Byte(212, 5, 1234, 24, 55, 166, 7, 88) ' 16 bytes for AES
    c.Initialize("DESEDE/CBC/PKCS5Padding")
    c.InitialisationVector = iv
    kg.Initialize("DESEDE")
    kg.KeyFromBytes(bconv.StringToBytes("1234567890123456", "ASCII"))
    data = bconv.StringToBytes(dataToEncrypt, "ASCII")
    data = c.Encrypt(data, kg.Key, True)
    Return B64.EncodeBtoS(data, 0, data.Length)
End Sub

Sub Activity_KeyPress (KeyCode As Int) As Boolean 'Consultamos si se ha pulsado alguna tecla
    'En la variable KeyCode obtenemos el código ASCII de la tecla pulsada
    If KeyCode=KeyCodes.KEYCODE_BACK Then

        Dim pulsarTecla As Int
        'En caso que pulsemos la tecla de "Retorno", nos preguntará mediante un
MsgBox si queremos salir del programa
        pulsarTecla = MsgBox2("Quieres salir?","Urgente","Si","", "No",Null)
        If pulsarTecla = DialogResult.POSITIVE Then

            ExitApplication

        Else If pulsarTecla = DialogResult.NEGATIVE Then
            Return True
        End If
    End If
End Sub

Sub load_config
    If File.Exists(File.DirDefaultExternal, "config.ini") Then
        Dim str1 As String
        str1=funcion.ReadIni(1,File.DirDefaultExternal, "config.ini")
        EditText_nombre.Text= str1
        funcion.minombre=str1
        Dim str3 As String
        str3=funcion.ReadIni(3,File.DirDefaultExternal, "config.ini")
        EditText_password.Text= str3
    End If
End Sub

```

```

Dim str5 As String
str5=funcion.ReadIni(5,File.DirDefaultExternal, "config.ini")
If str5 = "0" Then
CheckBox_password.Checked=False
EditText_password.Text=""
Else If str5 = "1" Then
CheckBox_password.Checked=True
End If
Else
'Si no existe el archivo
File.WriteString(File.DirDefaultExternal, "config.ini", "")
funcion.WriteIni(1,"",File.DirDefaultExternal, "config.ini")'nombre
funcion.WriteIni(3,"",File.DirDefaultExternal, "config.ini")'Password
funcion.WriteIni(5,"",File.DirDefaultExternal, "config.ini")'CheckBox_password
load_config
End If
End Sub

Sub save_config(message As String)
    Try
        Dim strconfig1 As String
        strconfig1=EditText_nombre.text
        funcion.WriteIni(1,strconfig1,File.DirDefaultExternal, "config.ini")
        If CheckBox_password.Checked=False Then
            Dim strconfig3 As String
            strconfig3="*****"
            funcion.WriteIni(3,strconfig3,File.DirDefaultExternal, "config.ini")
        Else If CheckBox_password.Checked=True Then
            Dim strconfig3 As String
            strconfig3=EditText_password.text
            funcion.WriteIni(3,strconfig3,File.DirDefaultExternal, "config.ini")
        End If
        If CheckBox_password.Checked=False Then
            funcion.WriteIni(5,"0",File.DirDefaultExternal, "config.ini")
        Else If CheckBox_password.Checked=True Then
            funcion.WriteIni(5,"1",File.DirDefaultExternal, "config.ini")
        End If
        MsgBox(message,"")
    Catch
        MsgBox("Error!", "")
    End Try

```

End Sub

Sub EditText_nombre_TextChanged (Old As String, New As String)

 Select Case EditText_nombre.text.Length

 Case 0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20

 Case Else

 EditText_nombre.text=Old

 End Select

End Sub

Sub EditText_verificacion_TextChanged (Old As String, New As String)

 Select EditText_verificacion.text.Length

 Case 0,1,2,3,4,5,6

 Case Else

 EditText_verificacion.text=Old

 End Select

End Sub

Sub verifica_spam

 EditText_verificacion.Text=""

 Label_verificacion.Text=""

 Dim numero1 As Int

 Dim numero2 As Int

 Dim numero3 As Int

 Dim numero4 As Int

 Dim numero5 As Int

 Dim numero6 As Int

 numero1 = Rnd(1, 9)

 numero2 = Rnd(1, 9)

 numero3 = Rnd(1, 9)

 numero4 = Rnd(1, 9)

 numero5 = Rnd(1, 9)

 numero6 = Rnd(1, 9)

 Label_verificacion.Text = numero1 & numero2 & numero3 & numero4 & numero5 & numero6

 comp = Label_verificacion.Text

End Sub

```

Sub Button_entrar_Click
    If funcion.Server.GetMyIP = "127.0.0.1" Then
        ToastMessageShow("No hay conexion a internet!", True)
    Else
        If EditText_nombre.Text.Length<6 Then
            ToastMessageShow("Nombre debe contener al menos 6 digitos!",
True)
                Return
            End If
        If EditText_password.Text.Length<6 Then
            ToastMessageShow("Password debe contener al menos 6 digitos!",
True)
                Return
            End If
        If EditText_verificacion.Text <> comp Then
            ToastMessageShow("Verificación incorrecta", True)
            Return
        End If
        If funcion.modos_programador=False Then
            If EditText_verificacion.Text=Label_verificacion.Text Then
            Else
                ToastMessageShow("Verificación Incorrecta", True)
                Return
            End If
        End If
        Button_entrar.Enabled=False
        RegistroUsuarios
        verifica_spam
        ocultar_teclado.HideKeyboard
    End If
End Sub

```

'Codigo para la conexión con la BDD-----

```

Sub RegistroUsuarios
    funcion.login_id = ""
    funcion.login_ok = False
    funcion.registro=True
    ProgressDialogShow("Registro!")

```

```
ExecuteRemoteQuery_reg1(funcion.Claveapp & "&registrousuarios&SELECT  
password FROM "&funcion.str_query&" WHERE Nombre = "&EditText_nombre.text&"  
LIMIT 1",funcion.registro_list_usuarios)
```

```
funcion.registro_ok=False
```

```
End Sub
```

```
Sub reg1_ResponseError (Response As HttpResponseMessage, Reason As String, StatusCode As Int,  
TaskId As Int)
```

```
    If funcion.modos_programador=True Then
```

```
        Log("Error: " & Reason & ", StatusCode: " & StatusCode)
```

```
    End If
```

```
    If Response <> Null Then
```

```
        Log(Response.GetString("UTF8"))
```

```
        Response.Release
```

```
    End If
```

```
    ProgressDialogHide
```

```
    Button_entrar.Enabled=True
```

```
End Sub
```

```
Sub ExecuteRemoteQuery_reg1(Query As String, TaskId As Int)
```

```
    Dim req As HttpRequest
```

```
    req.InitializePost2(funcion.linkdb, Query.GetBytes("UTF8"))
```

```
    funcion.reg1.Execute(req, TaskId)
```

```
End Sub
```

```
Sub reg1_ResponseSuccess (Response As HttpResponseMessage, TaskId As Int)
```

```
    Dim res As String
```

```
    res = Response.GetString("UTF8")
```

```
    Dim res2 As Int
```

```
    res2=res
```

```
    If funcion.modos_programador=True Then
```

```
        Log("Respuesta del servidor: " & res)
```

```
    End If
```

```
Try
```

```
If funcion.registro=True Then
```

```
If res2 = 1 Then
```

```
    ProgressDialogHide
```

```
    MsgBox("El nombre ya existe", "")
```

```
    Button_entrar.Enabled=True
```

```
    funcion.registro=False
```

```

        Response.Release
        Return
    Else If res2 = 0 Then
        'Modificar caracteres especiales
        Dim nombre1,imei1,email1,password1,lugar1 As String
        nombre1= EditText_nombre.text.Replace("","'")
        password1=Encrypt( EditText_password.text).Replace("","'")

        ExecuteRemoteQuery_reg2(funcion.Claveapp & "&insertarusuario&INSERT INTO
"&funcion.str_query&"(Nombre,password,estado) values('"&nombre1&"','"& password1
&"','1');",funcion.registro_list_usuarios)
        funcion.registro_ok=True
    End If
    End If
        funcion.registro=False
        Response.Release
        Catch
        MsgBox("Error 1!", "")
        Button_entrar.Enabled=True
        funcion.registro=False
        Response.Release
        mnuPage4_Click
        ProgressDialogHide
    End Try
        'hc 2
    End Sub

Sub reg2_ResponseError (Response As HttpResponse, Reason As String, StatusCode As Int,
TaskId As Int)

    If funcion.modos_programador=True Then
        Log("Error: " & Reason & ", StatusCode: " & StatusCode)
    End If
    If Response <> Null Then
        Log(Response.GetString("UTF8"))
        Response.Release
    End If
    ProgressDialogHide
End Sub

Sub ExecuteRemoteQuery_reg2(Query As String, TaskId As Int)
    Dim req As HttpRequest

```



```

        req.InitializePost2(funcion.linkdb, Query.GetBytes("UTF8"))
        funcion.reg2.Execute(req, TaskId)
End Sub
Sub reg2_ResponseSuccess (Response As HttpResponse, TaskId As Int)
    Dim res As String
    res = Response.GetString("UTF8")
    If funcion.modos_programador=True Then
        Log("Respuesta del servidor: " & res)
    End If
Try
    If funcion.registro_ok=True Then

        If res.Contains("HTTP/1.1 500 Internal Server Error")Then
            funcion.registro=False
            MsgBox("Es posible que tenga problemas de conexión o caracteres especiales para
reemplazar inténtelo de nuevo gracias! ", " ¡Error!")
            ProgressDialogHide
            Button_entrar.Enabled=False
            Return
        Else
            Button_entrar.Enabled=False
mnuPage3_Click
            ProgressDialogHide
            save_config("Registro exitoso")
            End If
            End If
            funcion.registro_ok=False
        Catch
            MsgBox("Error 2!", "")
            Button_entrar.Enabled=True
            funcion.registro=False
            funcion.registro_ok=False
            Response.Release
            mnuPage4_Click
            ProgressDialogHide
        End Try
End Sub

Sub Button_iniciar_sesion_Click
    If funcion.Server.GetMyIP = "127.0.0.1" Then
        ToastMessageShow("Conexion a internet ausente", True)
    End If
End Sub

```

```

Else
    funcion.login_id = ""
    funcion.login_ok = False

    funcion.login = True
    ProgressDialogShow("Iniciar Sesión!")
    ExecuteRemoteQuery_login(funcion.Claveapp & "&login&SELECT ID, Nombre,
password, estado FROM "&funcion.str_query&" WHERE Nombre =
"&EditText_nombre.text&" AND password = "&Encrypt(EditText_password.text)&" AND
estado='1'",funcion.login_list_usuarios)
    ocultar_teclado.HideKeyboard
    End If
End Sub

Sub ExecuteRemoteQuery_login(Query As String, TaskId As Int)
    Dim req As HttpRequest
    req.InitializePost2(funcion.linkdb, Query.GetBytes("UTF8"))
    funcion.hc_login.Execute(req, TaskId)
End Sub

Sub login_ResponseError (Response As HttpResponse, Reason As String, StatusCode As
Int, TaskId As Int)
    If funcion.modos_programador=True Then
        Log("Error: " & Reason & ", StatusCode: " & StatusCode)
    End If
    If Response <> Null Then
        Log(Response.GetString("UTF8"))
        Response.Release
    End If
    ProgressDialogHide
End Sub

Sub hc_login_ResponseSuccess (Response As HttpResponse, TaskId As Int)
    Try
        Dim res As String
        res = Response.GetString("UTF8")
        res = res.Replace("''''", "")
    End Try
    If funcion.modos_programador=True Then
        Log("Respuesta del servidor login: " & res)
    End If
End Sub

```

```

        If res.Contains("HTTP/1.1 500 Internal Server Error")Then
            MsgBox("Es posible que tenga problemas para conectarse o caracteres especiales para
reemplazar inténtelo de nuevo gracias! ", " ¡Error!")
            ProgressDialogHide
            funcion.login=False
            funcion.login_ok=False
            Response.Release
        Return
        End If
        If funcion.login_ok=False Then
            If funcion.login=True Then
                Dim strlogin As Int
                strlogin= res
                Select True
                Case strlogin > 0
                    funcion.login_id=strlogin
                    ExecuteRemoteQuery_login2(funcion.Claveapp & "&datosdesesion&SELECT ID,
Nombre, password, estado FROM "&funcion.str_query&" WHERE ID =
"&funcion.login_id&" AND estado = '1"',funcion.login_list_usuarios)'login
                    Case strlogin = 0
                        ToastMessageShow("Contraseña o nombre incorrecto", True)
                        funcion.login_ok=False
                        funcion.login=False
                        Response.Release
                        ProgressDialogHide
                        Return
                    End Select
                End If
            End If
            funcion.login=False
            Response.Release
            Catch
                MsgBox("Error!", "")
                funcion.login=False
                funcion.login_ok=False
                Response.Release
                ProgressDialogHide
                mnuPage3_Click
            Return
        End Try
    End Sub

```

'-----EjecucionRemotaConsulta_login2-----'

```
Sub hc_login2_ResponseError (Response As HttpResponse, Reason As String, StatusCode  
As Int, TaskId As Int)
```

```
    If funcion.modos_programador=True Then  
        Log("Error: " & Reason & ", StatusCode: " & StatusCode)  
    End If
```

```
    If Response <> Null Then  
        Log(Response.GetString("UTF8"))  
        Response.Release
```

```
    End If  
    ProgressDialogHide
```

```
End Sub
```

```
Sub ExecuteRemoteQuery_login2(Query As String, TaskId As Int)
```

```
    Dim req As HttpRequest  
    req.InitializePost2(funcion.linkdb, Query.GetBytes("UTF8"))  
    funcion.hc_login2.Execute(req, TaskId)
```

```
End Sub
```

```
Sub hc_login2_ResponseSuccess (Response As HttpResponse, TaskId As Int)
```

```
    Try  
        Dim res As String  
        res = Response.GetString("UTF8")  
        res= res.Replace("''''''", "")
```

```
    If funcion.modos_programador=True Then  
        Log("Respuesta del servidor login: " & res)
```

```
    End If  
    If res.Contains("HTTP/1.1 500 Internal Server Error")Then
```

```
        MsgBox("Es posible que tenga problemas para conectarse o caracteres especiales para  
reemplazar inténtelo de nuevo gracias! ", " ¡Error!")
```

```
        ProgressDialogHide  
        funcion.login=False  
        funcion.login_ok=False  
        Response.Release
```

```
    Return  
    End If  
    funcion.login_ok=True  
    mnuPage2_Click  
    save_config("Login OK!")  
    ProgressDialogHide
```

```
        Catch
        MsgBox("Error!", "")
        funcion.login=False
        funcion.login_ok=False
        Response.Release
        ProgressDialogHide
        mnuPage3_Click
        Return
    End Try
End Sub

#End Region
#Region "modificar"
Sub Button_editar_Click
    If funcion.Server.GetMyIP = "127.0.0.1" Then
        ToastMessageShow("Conexion a internet ausente!", True)
    Else

If EditText_nombre.Text.Length<6 Then
    ToastMessageShow("Nombre al menos de 6 caracteres!", True)
    Return
End If

If EditText_password.Text.Length<6 Then
    ToastMessageShow("Password al menos de 6 caracteres!", True)
    Return
End If
If funcion.modo_programador=False Then
If EditText_verificacion.Text=Label_verificacion.Text Then
    Else
    ToastMessageShow("Numero de verificacion incorrecto!", True)
    Return
End If
End If
ActualizarNombre
Button_editar.Enabled=False
ocultar_teclado.HideKeyboard
End If
End Sub
```

```
Sub ActualizarNombre
```

```
    funcion.modificar=True
```

```
        ProgressDialogShow("Modificar dato!")
```

```
    Try
```

```
        If funcion.modificar=True Then
```

```
            If funcion.login_ok=True Then
```

```
                If funcion.login_id="" Then
```

```
                    Else
```

```
                'modificar caracteres especiales
```

```
                    Dim nombre1,imei1,email1,password1,lugar1 As String
```

```
                    nombre1= EditText_nombre.text.Replace("","")
```

```
                    password1=Encrypt( EditText_password.text).Replace("","")
```

```
                    ExecuteRemoteQuery_modificar(funcion.Claveapp & "&editarusuarios&UPDATE  
                    "&funcion.str_query&" SET Nombre="" & nombre1 & ",password=""& password1 & "  
                    WHERE "&funcion.str_query&".ID=""& funcion.login_id &  
                    "";" ,funcion.modificar_list_usuarios)
```

```
                        End If
```

```
                    End If
```

```
                        End If
```

```
                            funcion.modificar=False
```

```
                                Catch
```

```
                                    MsgBox("Error!", "")
```

```
                                        Button_editar.Enabled=True
```

```
                                            funcion.modificar=False
```

```
                                                ProgressDialogHide
```

```
                                End Try
```

```
End Sub
```

```
Sub ExecuteRemoteQuery_modificar(Query As String, TaskId As Int)
```

```
    Dim req As HttpRequest
```

```
    req.InitializePost2(funcion.linkdb, Query.GetBytes("UTF8"))
```

```
    funcion.hc_modificar.Execute(req, TaskId)
```

```
End Sub
```

```
Sub modificar_ResponseError (Response As HttpResponse, Reason As String, StatusCode  
As Int, TaskId As Int)
```

```
    If funcion.modos_programador=True Then
```

```
        Log("Error: " & Reason & ", StatusCode: " & StatusCode)
```

```
    End If
```

```
    If Response <> Null Then
```

```
        Log(Response.GetString("UTF8"))
```

```

        Response.Release
    End If
    ProgressDialogHide
    Button_editar.Enabled=True
End Sub
Sub hc_modificar_ResponseSuccess (Response As HttpResponse, TaskId As Int)
    Dim res As String
    res = Response.GetString("UTF8")
    If funcion.mododo_programador=True Then
        Log("Respuesta del servidor modificar: " & res)
    End If
    If res.Contains("HTTP/1.1 500 Internal Server Error")Then
        MsgBox("Es posible que tenga problemas para conectarse o caracteres especiales para reemplazar inténtelo de nuevo gracias! ", " ¡Error!")
        Button_editar.Enabled=True
        ProgressDialogHide
        funcion.modificar=False

    Else
        verifica_spam
        funcion.login_id=""
        funcion.login_ok=False
        funcion.modificar=False
        mnuPage3_Click
        EditText_nombre.Enabled=True
        ProgressDialogHide
        save_config("Datos de modificación exitosa! Tiene que volver a inicio de sesión")
        Button_editar.Enabled=True
    End If
    Response.Release
    EditText_nombre.Enabled=True
End Sub

'-----Codigo de conexión con los sensores-----

Sub FetchCountriesList
    ProgressDialogShow("Cargando...")
    ExecuteRemoteQuery("SELECT sensor1,sensor2,sensor3 FROM esp8266", "r1")
    '-----Estado de los sensores-----
    ExecuteRemoteQuery("SELECT estado as 'e1' FROM sensores WHERE
ID_SENSOR=1", "r2")

```

```

        ExecuteRemoteQuery("SELECT estado as 'e2' FROM sensores WHERE
ID_SENSOR=2", "r2")
        ExecuteRemoteQuery("SELECT estado as 'e3' FROM sensores WHERE
ID_SENSOR=3", "r2")
        '-----Sensores-----'
        ExecuteRemoteQuery("SELECT dato as 's1' FROM historial_sensores WHERE
ID_SENSOR=1 order by ID_HISTORIAL DESC limit 0,1", "r1")
        ExecuteRemoteQuery("SELECT dato as 's2' FROM historial_sensores WHERE
ID_SENSOR=2 order by ID_HISTORIAL DESC limit 0,1", "r1")
        ExecuteRemoteQuery("SELECT dato as 's3' FROM historial_sensores WHERE
ID_SENSOR=3 order by ID_HISTORIAL DESC limit 0,1", "r1")
    End Sub

Sub ExecuteRemoteQuery(Query As String, JobName As String)
    Dim job As HttpJob
    job.Initialize(JobName, Me)
    job.PostString("http://www.beemarketing.com.ec/parqueadero/scriptBDD/archivo
.php", Query)
End Sub

Sub JobDone(Job As HttpJob)
    ProgressDialogHide
    If Job.Success Then
        Select Job.JobName

            Case "r2"
                Dim parser As JSONParser
                parser.Initialize(Job.GetString)
                Dim root As List = parser.NextArray
                For Each colroot As Map In root
                    Dim e1 As String = colroot.Get("e1")
                    Dim e2 As String = colroot.Get("e2")
                    Dim e3 As String = colroot.Get("e3")

                    Log(e1)
                    Log(e2)
                    Log(e3)
                    If e1 = "1" Then
                        ImageView_Activado.Visible=False
                    End If
                    If e1 = "0" Then

```



```

        lblsensor1.Visible=False
        ImageView_Carro.Visible=False
        ImageView_Activado.Visible=True
    End If
    If e2 = "1" Then
        ImageView_Activado1.Visible=False
    End If
    If e2 = "0" Then
        lblsensor2.Visible=False
        ImageView_Carro1.Visible=False
        ImageView_Activado1.Visible=True
    End If
    If e3 = "1" Then
        ImageView_Activado2.Visible=False
    End If
    If e3 = "0" Then
        lblsensor3.Visible=False
        ImageView_Carro2.Visible=False
        ImageView_Activado2.Visible=True
    End If
Next
Case "r1"
    Log(Job.GetString)
    Dim parser As JSONParser
    parser.Initialize(Job.GetString)
    Dim root As List = parser.NextArray
    For Each colroot As Map In root
        Dim s1 As String = colroot.Get("s1")
        Dim s2 As String = colroot.Get("s2")
        Dim s3 As String = colroot.Get("s3")
        Log(s1)
        Log(s2)
        Log(s3)
        If s1 = "1" Then
            lblsensor1.Visible=False
            ImageView_Carro.Visible=True
        End If
        If s1 = "0" Then
            lblsensor1.Visible=True
            ImageView_Carro.Visible=False
        End If
    End If

```

```
If s2 = "1" Then
    lblsensor2.Visible=False
    ImageView_Carro1.Visible=True
End If
If s2 = "0" Then
    lblsensor2.Visible=True
    ImageView_Carro1.Visible=False
End If
If s3 = "1" Then
    lblsensor3.Visible=False
    ImageView_Carro2.Visible=True
End If
If s3 = "0" Then
    lblsensor3.Visible=True
    ImageView_Carro2.Visible=False
End If
```

```
Next
```

```
End Select
```

```
Else
```

```
    ToastMessageShow("Error: " & Job.ErrorMessage, True)
```

```
End If
```

```
Job.Release
```

```
End Sub
```

```
Sub Timer1_Tick
```

```
' Cada vez que pasan 100 milisegundos
```

```
FetchCountriesList
```

```
'-----
```

```
End Sub
```

CODIGO DESARROLADO EN PHP.

Para la conexión con la base de datos utilizamos tres archivos:



EL PRIMER Y SEGUNDO ARCHIVO PERMITE LA CONEXIÓN CON LOS SENSORES.

```
<?php
```

```
$databasehost = "localhost";
```

```
$databasename = "beemarke_parquadero";
```

```
$databaseusername ="beemarke_parq";
```

```
$databasepassword = "zCOVu8FyTPqh";
```

```
$con = mysql_connect($databasehost,$databaseusername,$databasepassword) or  
die(mysql_error());
```

```
mysql_select_db($databasename) or die(mysql_error());
```

```
$query = file_get_contents("php://input");
```

```
$sth = mysql_query($query);
```

```
if (mysql_errno()) {  
    header("HTTP/1.1 500 Internal Server Error");  
    echo $query.\n';  
    echo mysql_error();  
}
```

```
else  
{
```

```
    $rows = array();
```

```
    while($r = mysql_fetch_assoc($sth)) {
```

```
        $rows[] = $r;
```

```
    }
```

```
    print json_encode($rows);
```

```
}
```

```
?>
```

-----SEGUNDO ARCHIVO-----

```
<?php
```

```
//$sensor1=$_GET['sensor1'];
```

```
//$sensor2=$_GET['sensor2'];
```

```
//$sensor3=$_GET['sensor3'];
```

```
$servername = "localhost";
```

```
$username = "beemarke_parq";
```

```
$password = "zCOVu8FyTPqh";
```

```
$dbname = "beemarke_parquadero";
```

```
//Create connection
```

```
$conn = new mysqli($servername, $username, $password, $dbname);
```

```
//Check connection
```

```
if($conn->connect_error){
```

```
    die("Connection failed:" . $conn->connect_error);
```

```
}
```

```
$val = $_GET['sensor1'];
```

```
$sql="INSERT INTO historial_sensores (ID_SENSOR, DATO, FECHA)
```

```
SELECT '1', '$val', CURRENT_TIME()
```

```
FROM sensores
```

```
WHERE exists (SELECT ID_SENSOR
```

```
    FROM sensores
```

```
    WHERE ID_SENSOR=1 AND estado=1) AND ID_SENSOR=1";
```

```
if($conn->query($sql)== TRUE){
```

```
    echo "save ok";
```

```
}else{
```

```
    echo "Error:" . $sql . "<br>" . $conn->error;
```

```
}
```

```
$conn->close();
```

```
//Create connection
```

```
$conn = new mysqli($servername, $username, $password, $dbname);
```

```
//Check connection
```

```
if($conn->connect_error){
```

```

        die("Connection failed:" . $conn->connect_error);
    }
    $val1 = $_GET['sensor2'];
    $sql="INSERT INTO historial_sensores (ID_SENSOR, DATO, FECHA)
    SELECT '2', '$val1', CURRENT_TIME()
    FROM sensores
    WHERE exists (SELECT ID_SENSOR
        FROM sensores
        WHERE ID_SENSOR=2 AND estado=1) AND ID_SENSOR=2";

    if($conn->query($sql)== TRUE){
        echo "save ok";
    }else{
        echo "Error:" . $sql . "<br>" . $conn->error;
    }
    $conn->close();

```

```

//Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
//Check connection
if($conn->connect_error){
    die("Connection failed:" . $conn->connect_error);
}
$val2 = $_GET['sensor3'];
$sql="INSERT INTO historial_sensores (ID_SENSOR, DATO, FECHA)
SELECT '3', '$val2', CURRENT_TIME()
FROM sensores
WHERE exists (SELECT ID_SENSOR
    FROM sensores
    WHERE ID_SENSOR=3 AND estado=1) AND ID_SENSOR=3";

if($conn->query($sql)== TRUE){
    echo "save ok";
}else{
    echo "Error:" . $sql . "<br>" . $conn->error;
}

```

```
$conn->close();  
?>
```

El tercer archivo permite realizar la conexión con los usuarios.

```
<?php
```

```
$userdb="beemarke_parq"; // user db  
$passworddb="zCOVu8FyTPqh";// Password db  
$databasedb="beemarke_parquadero";// Name db
```

```
$rows = array();
```

```
$con = mysql_connect("localhost",$userdb,$passworddb) or die(mysql_error());  
mysql_select_db($databasedb) or die(mysql_error());  
$query = file_get_contents("php://input");  
//$sth = mysql_query($query);
```

```
    $action = array();  
    $action = split(" ", strtolower($query));
```

```
$key = "bXZhbm9tb250aUBsaWJlcm8uaXB=";  
$arr1 = array();  
$arr1 = split($key, ($query));
```

```
$key1 = (strtolower($action[0]));  
$key2 = $key;  
$key2 = (strtolower($key2));  
$action[0] = str_replace($key2, "", $action[0]);  
// clave para consultar  
$query2 = str_replace($key, "", $query);
```

```
$controlclave = str_replace($query2, "", $query);
```

```
    // Se comprueba que la clave no es nula  
    if ( $controlclave == null )  
    {  
        print json_encode("clave invalida!");  
        return;  
    }
```

```

else
{
// Se verifica que la clave sea igual a la enviada
if ( $controlclave == $key )
{
// -----select-----
if ( $action[0] == "&login&select" )
{
$query2 = str_replace("&login&", "", $query2);
$sth = mysql_query($query2);
}
if ( $action[0] == "&datosdesesion&select" )
{
$query2 = str_replace("&datosdesesion&", "", $query2);
$sth = mysql_query($query2);
}
if ( $action[0] == "&registrousuarios&select" )
{
$query2 = str_replace("&registrousuarios&", "", $query2);
$sth = mysql_query($query2);
}
// -----update-----
if ( $action[0] == "&editarusuarios&update" )
{
$query2 = str_replace("&editarusuarios&", "", $query2);
$sth = mysql_query($query2);
}
// -----insert-----
if ( $action[0] == "&insertarusuario&insert" )
{
$query2 = str_replace("&insertarusuario&", "", $query2);
$sth = mysql_query($query2);
}

//-----//
// Seguridad de la base de datos //
//-----//
if ( $action[0] == "delete" )
{
print json_encode("No es posible");
}

```

```

    }

if ( $action[0] == "create" )
{
    print json_encode("No es posible!");
}

}

}

//-----Db-Error-----//
if (mysql_errno()) {
    header("HTTP/1.1 500 Internal Server Error");

    echo ("HTTP/1.1 500 Internal Server Error");
    echo $query.'\n';
    echo mysql_error();
}
else
{
//-----select-----

if ( $action[0] == "&login&select" )
{
$row = mysql_fetch_assoc($sth);
$num = mysql_num_rows ($sth);

//print json_encode($row['ID']);
//print json_encode($row);
if($num == '0'){
// contraseña incorrecta
print json_encode(0);
}else{
// inicio de sesión correcto
print json_encode($row['ID']);
}
}

if ( $action[0] == "&datosdesesion&select" )

```



```
{
$row = mysql_fetch_assoc($sth);
$num = mysql_num_rows ($sth);

print json_encode($row);

}

//comprobar si el usuario ya está registrado
if ( $action[0] == "&registrousuarios&select" )
{
$row = mysql_fetch_assoc($sth);
$num = mysql_num_rows ($sth);

if($num == '0'){
//Nombre no existe!
print json_encode(0);
}else{
//Nombre existe!
print json_encode(1);
}
}
}
?>
```