



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

***Desarrollo de un sistema HMI en las plataformas
Android e iOS para el monitoreo de la red inalámbrica
de sensores agrícolas ubicada en el barrio
Rumipamba de Navas cantón Salcedo, provincia de
Cotopaxi***

Ingeniería Electrónica e Instrumentación



**RESPONSABLES
DEL PROYECTO**

- Endara Crespata Evelyn
Fernanda
- Vargas Ortiz Víctor Alfonso

**COLABORADORES
CIENTÍFICOS**

- **Director:** Ing. David Rivas
- **Codirector:** Ing. Mayra Erazo



Agenda

- Descripción de proyecto
- Objetivos
- Características Android e iOS
- Etapas del proyecto
- Diagrama de flujo y desarrollo de las aplicaciones
- Análisis de resultados
- Conclusiones y recomendaciones



Descripción de proyecto

El presente proyecto tiene como finalidad desarrollar un HMI móvil en las plataformas Android e IOS, que permita el monitoreo de variables climáticas de la red inalámbrica de sensores de un invernadero de producción agrícola de tomate de riñón implementado en la ciudad de Salcedo, con su respectiva visualización de históricos y su monitoreo en tiempo real.



Objetivos

Investigar las características de diversos software disponibles para la programación en plataformas Android e iOS.

Desarrollar el servidor web, que permita el acceso a la información desde cualquier lugar que tenga cobertura a internet.

Realizar las pruebas de comunicación inalámbrica en las plataformas Android e iOS con el servidor web.

Realizar un informe que contenga los resultados del proyecto y un análisis comparativo de las plataformas Android e iOS.



Comunicación móvil

Desarrolladas durante la segunda guerra mundial

Cuyo propósito es acortar distancia en el menor tiempo posible.

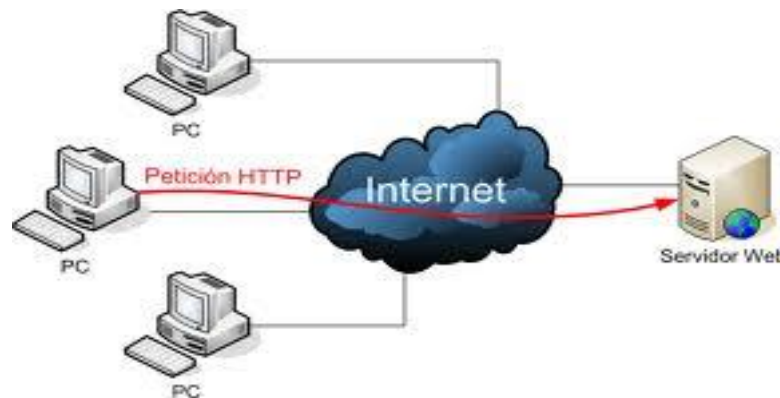
Diferentes tecnologías adaptadas como antenas, satélites, repetidoras de señal

Debe constar con un receptor y un emisor.



Servidor web

- Un servidor web recibe peticiones de un cliente o usuario de internet, emitiendo respuestas mediante el envío de ficheros solicitados, implementados con protocolos o binarios y estándares para intercambiar información y aplicaciones.



Tecnología Wireless

Utilizan un medio de propagación por modulación de ondas electromagnéticas a través del espacio.

Brinda mayor comodidad permitiendo que cualquiera que tenga acceso a la red se conecte desde diferentes ubicaciones dentro de una cobertura determinada.

La instalación y configuración es más sencilla y económica que la de una infraestructura inalámbrica o cableada.



Tecnología 3G

Acceso a internet a través del teléfono Smartphone, ordenador de bolsillo y Tablet

Permite navegar en la Web, redes sociales y recibir y enviar correos desde cualquier punto geográfico que contenga cobertura.

La transmisión de voz tiene una calidad equiparable a la de las redes fijas.



Características Android

Código abierto.

Núcleo basado en Linux.

Utiliza SQLite para el almacenamiento de datos.

Soporte de Java .

Soporte de HTML, HTML5, Adobe Flash Player, etc.

Catálogo de aplicaciones (Google Play).

Bluetooth



Características iOS

Código cerrado

Basado en Darwin BSD

Interfaz gráfica está diseñada para el touch screen

Capacidad para gestos multitouch.

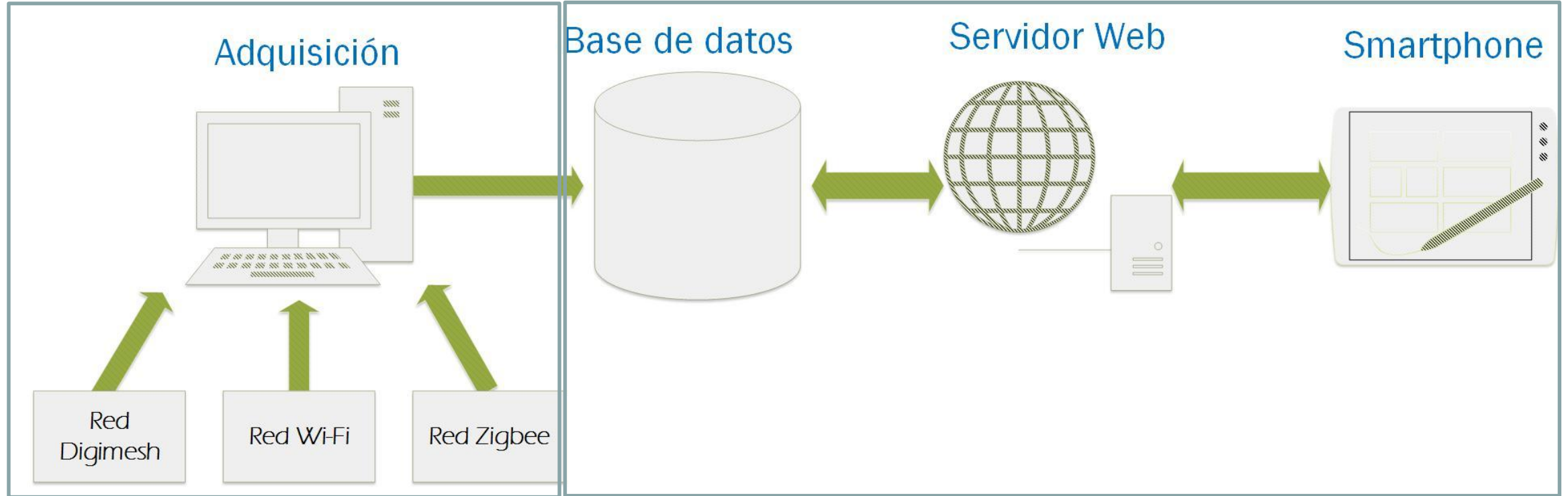
No tiene soporte para Adobe Flash ni Java

Emplea unos 500 MB de almacenamiento sistema operativo

Catalogo de aplicaciones App Store



Etapas del proyecto



VARIABLES de la red Digimesh

RED DIGIMESH						
VARIABLE	NODO 1	NODO 2	NODO 3	NODO 4	TIPO	LONGITUD
TEMPERATURA	X	X	X		FLOAT	11
HUMEDAD	X	X	X		FLOAT	11
LUMINOSIDAD	X	X	X		FLOAT	11
ULTRAVIOLETA			X		FLOAT	11
RADIACION S.	X				FLOAT	11
CO2				X	FLOAT	11
CO				X	FLOAT	11
BATERIA (v)	X	X	X	X	FLOAT	11
%BATERIA	X	X	X	X	FLOAT	11
FECHA	X	X	X	X	VARCHAR	25

VARIABLES DE LA RED ZIGBEE

RED ZIGBEE					
VARIABLE	NODO 1	NODO 2	NODO 3	TIPO	LONGITUD
TEMPERATURA	X	X	X	FLOAT	11
HUMEDAD	X	X	X	FLOAT	11
LUMINOSIDAD	X	X	X	FLOAT	11
ULTRAVIOLETA			X	FLOAT	11
RADIACION S.		X		FLOAT	11
VEL. VIENTO	X			FLOAT	11
DIR. VIENTO	X			FLOAT	11
BATERIA (v)	X	X	X	FLOAT	11
%BATERIA	X	X	X	FLOAT	11
FECHA	X	X	X	VARCHAR	25



Variables de la red Wi Fi

RED WIFI						
VARIABLE	NODO 1	NODO 2	NODO 3	NODO 4	TIPO	LONGITUD
TEMPERATURA	X	X	X		FLOAT	11
HUMEDAD	X	X	X		FLOAT	11
LUMINOSIDAD	X	X	X		FLOAT	11
CANT. LLUVIA (min)	X				FLOAT	11
CANT. LLUVIA (h)	X				FLOAT	11
ULTRAVIOLETA			X		FLOAT	11
RADIACION S.		X			FLOAT	11
CO2				X	FLOAT	11
CO				X	FLOAT	11
BATERIA (v)	X	X	X	X	FLOAT	11
%BATERIA	X	X	X	X	FLOAT	11
FECHA	X	X	X	X		

Tablas

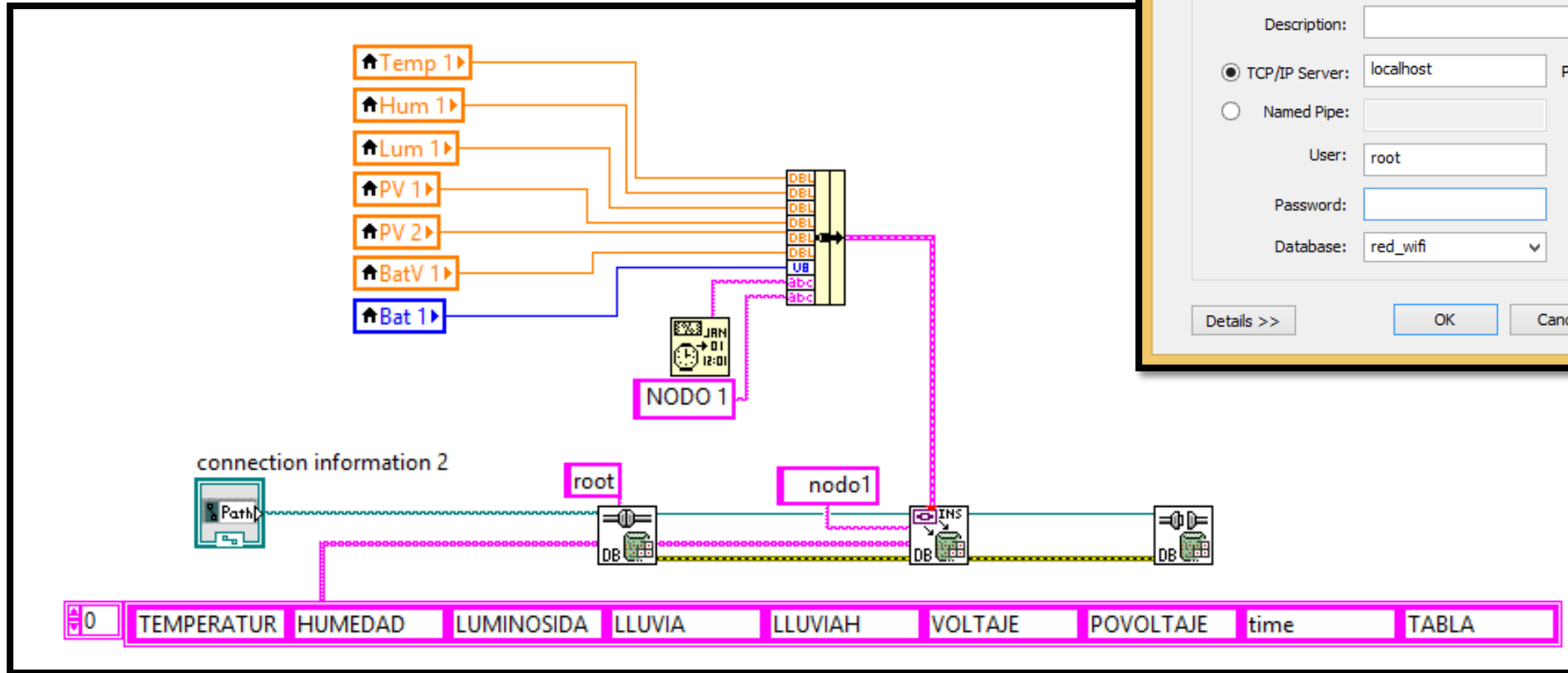
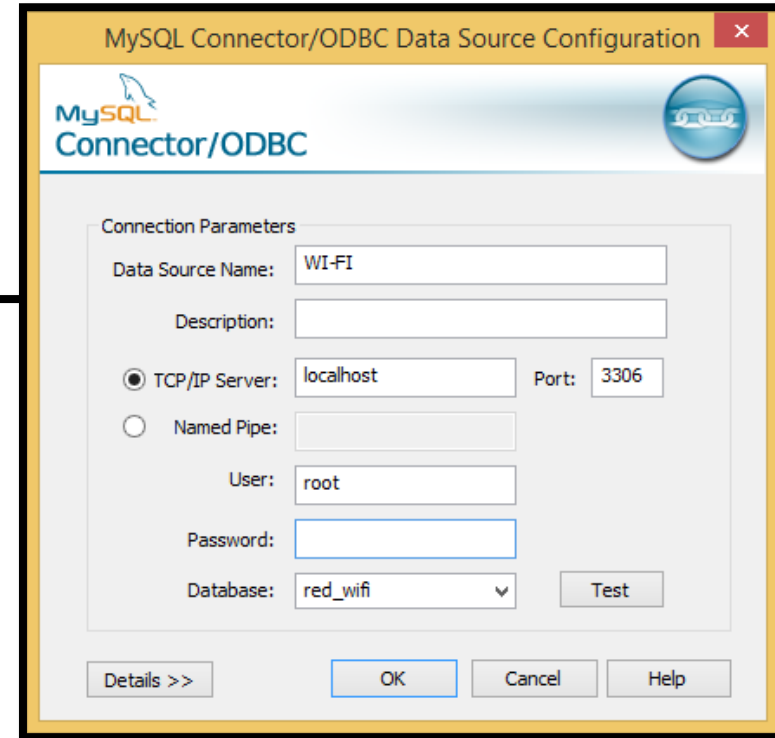
localhost ▶ red_wfi ▶ nodo1

Examinar Estructura SQL Buscar Insertar Exportar Importar Operaciones Seguimiento

#	Columna	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra	Acción
<input type="checkbox"/>	1 id	int(244)			No	Ninguna	AUTO_INCREMENT	Cambiar Eliminar Más ▼
<input type="checkbox"/>	2 TEMPERATURA	float			No	Ninguna		Cambiar Eliminar Más ▼
<input type="checkbox"/>	3 HUMEDAD	float			No	Ninguna		Cambiar Eliminar Más ▼
<input type="checkbox"/>	4 LUMINOSIDAD	float			No	Ninguna		Cambiar Eliminar Más ▼
<input type="checkbox"/>	5 LLUVIA	float			No	Ninguna		Cambiar Eliminar Más ▼
<input type="checkbox"/>	6 LLUVIAH	float			No	Ninguna		Cambiar Eliminar Más ▼
<input type="checkbox"/>	7 VOLTAJE	float			No	Ninguna		Cambiar Eliminar Más ▼
<input type="checkbox"/>	8 POVOLTAJE	float			No	Ninguna		Cambiar Eliminar Más ▼
<input type="checkbox"/>	9 time	timestamp			No	CURRENT_TIMESTAMP		Cambiar Eliminar Más ▼
<input type="checkbox"/>	10 TABLA	varchar(33) latin1_swedish_ci			No	Ninguna		Cambiar Eliminar Más ▼



- LabView Configuraciones



Base de datos Wifi nodo1

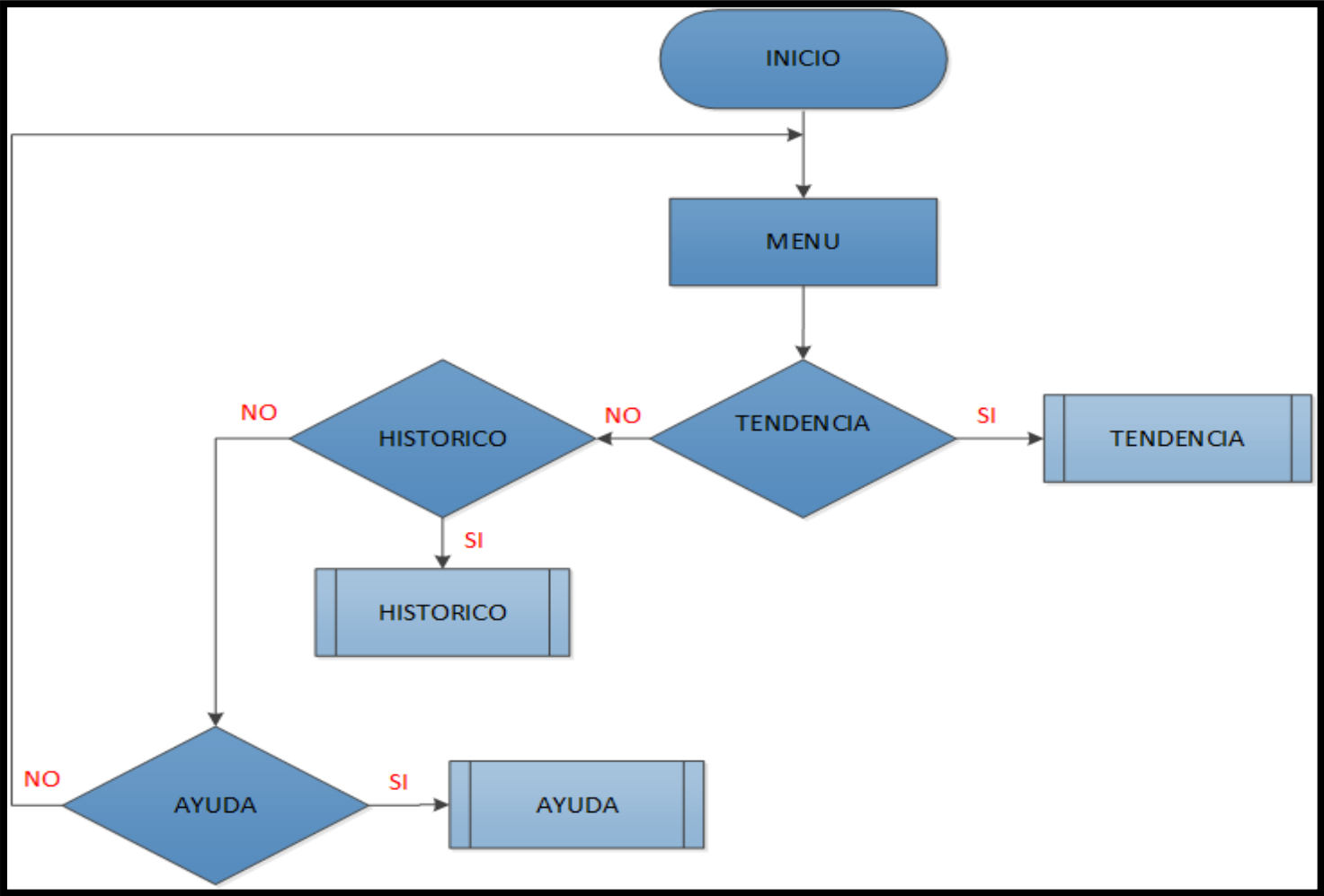
id	TEMPERATURA	HUMEDAD	LUMINOSIDAD	LLUVIA	LLUVIAH	VOLTAJE	POVOLTAJE	time	TABLA
1	27.1	33.9	73161.1	0	0	4.17	96	2015-06-05 17:16:04	NODO 1
2	13.87	40.3	12540.8	0	0	4.15	94	2015-06-05 17:19:52	NODO 1
3	13.87	41	17805.1	0	0	4.15	94	2015-06-05 17:22:23	NODO 1
4	13.87	42.7	17805.1	0	0	4.15	94	2015-06-05 17:24:56	NODO 1
5	13.87	42.4	16412.1	0	0	4.15	94	2015-06-05 17:27:30	NODO 1
6	13.87	42	16412.1	0	0	4.15	94	2015-06-05 17:29:53	NODO 1
7	13.87	40.4	16412.1	0	0	4.16	94	2015-06-05 17:32:38	NODO 1
8	13.87	39.6	15893.6	0	0	4.16	94	2015-06-05 17:35:55	NODO 1
9	13.87	39.6	15893.6	0	0	4.16	94	2015-06-05 17:41:09	NODO 1
10	13.87	38.2	14105.4	0	0	4.16	94	2015-06-05 17:44:35	NODO 1
11	13.87	38.2	32104.7	0	0	4.16	94	2015-06-05 17:48:30	NODO 1
12	14.52	38.2	63340.6	0	0	4.16	94	2015-06-05 17:52:44	NODO 1



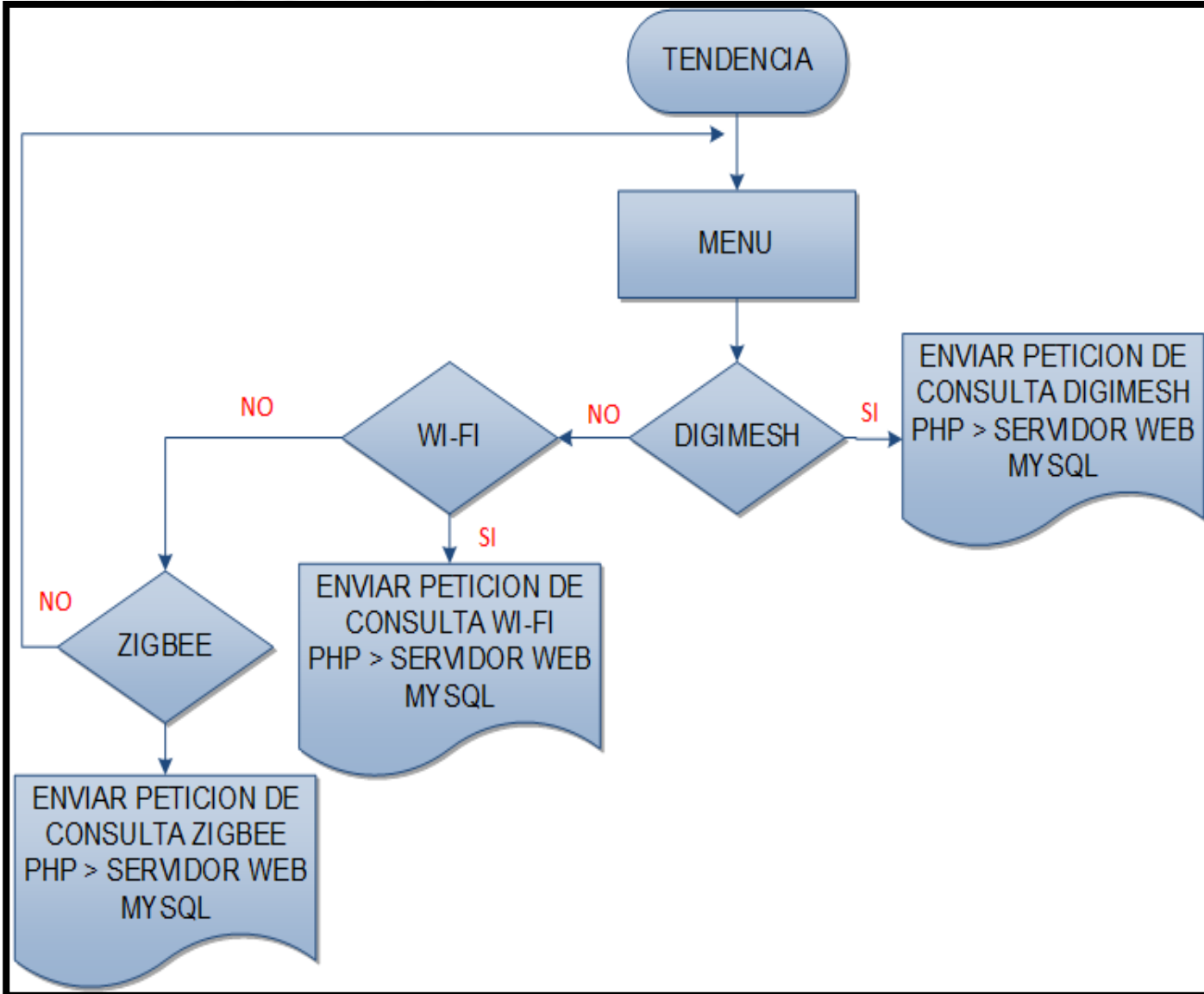
Diseño Y Desarrollo de las Aplicaciones en la plataformas Android e iOS



Diagrama de flujo menú principal



1. Tendencia



Programación en Android

Archivo .java

```
public void getData() {
    String result = "";
    InputStream isr = null;
    try {
        HttpClient httpClient = new DefaultHttpClient();
        HttpPost httpost = new HttpPost("http://190.99.73.45:8080/wifi/respuesta.php");
        HttpResponse response = httpClient.execute(httpost);
        HttpEntity entity = response.getEntity();
        isr = entity.getContent();
    } catch (Exception e) {
        Log.e("log_tag", "ERROR EN LA CONEXION " + e.toString());

        final ProgressDialog ringProgressDialog = ProgressDialog.show(
            Redwifit.this, "ACCESO NEGADO...", "REVISE LA CONEXION ",
            true);
        ringProgressDialog.setCancelable(true);
        new Thread(new Runnable() {
            @Override
            public void run() {
                try {
                    Thread.sleep(10000);

                } catch (Exception e) {
                }

                ringProgressDialog.dismiss();
            }
        }).start();
    }
}
```



Programación en iOS

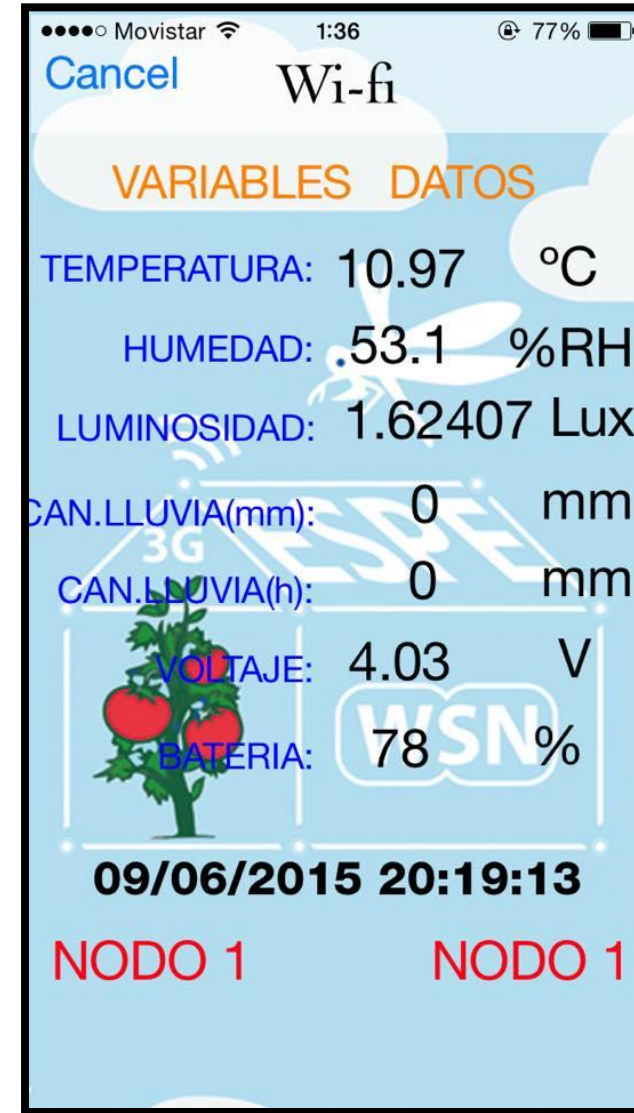
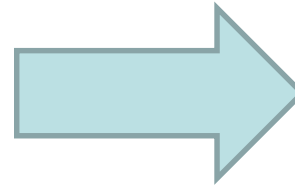
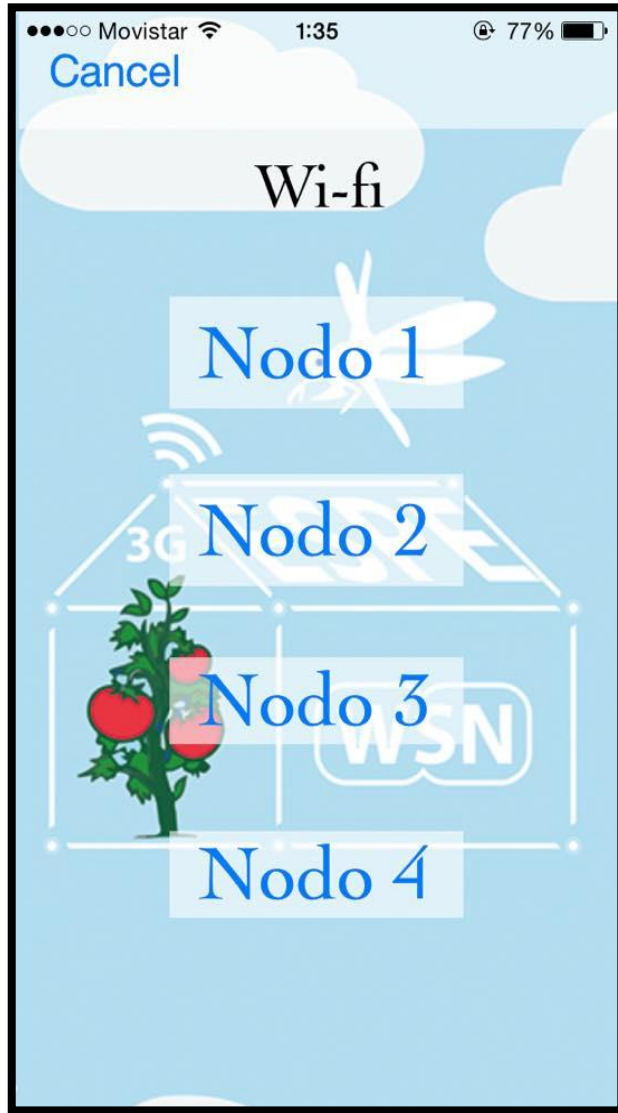
- Gestión de la comunicación y petición de consulta entre aplicación – servidor web.

```
- (void) retrieveData
{
    #define getDataURL @"http://190.99.73.45:8080/wifi/respuesta.php"
    NSURL *url = [NSURL URLWithString:getDataURL];
    NSData *data = [NSData dataWithContentsOfURL:url];
    NSError *error = nil;

    if (data == nil) {
        NSLog(@"ERRO");
        UIAlertView *alert = [[UIAlertView alloc] initWithTitle:@"ERROR"
        message:@"Conexion Fail. Revise el servidor Host" delegate: nil cancelButtonTitle:@"Ok" otherButtonTitles:nil];
        [alert show];
        return;
    }
}
```



Visualización de Datos *TENDENCIA*



Programación en Android

Objetos JSON y recibir datos

```
JSONObject json = jsonArray.getJSONObject(0);  
JSONObject json1 = jsonArray.getJSONObject(1);  
JSONObject json2 = jsonArray.getJSONObject(2);  
JSONObject json3 = jsonArray.getJSONObject(3);
```

```
if (json.getString("TABLA").equals("NODO 1")) {  
    a = a + json.getString("TEMPERATURA")  
    + json.getString("HUMEDAD")  
    + json.getString("LUMINOSIDAD")  
    + json.getString("LLUVIA")  
    + json.getString("LLUVIAH")  
    + json.getString("POVOLTAJE")  
    + json.getString("VOLTAJE")  
    + json.getString("FECHA")  
    + json.getString("TABLA")  
    nodo1t = a.toString();}
```

Imprimir datos

```
public void bt_nodo1wifi(View v) {  
  
    Intent act = new Intent(this, Nodoswifit.class);  
    act.putExtra(NODO1T, nodo1t);  
    startActivity(act);  
}
```



Programación en iOS

Objetos JSON y recibir datos

```
for (int i = 0; i < jsonArray.count; i++) {  
  
    NSString * cTEMPERATURA = [[jsonArray objectAtIndex:i] objectForKey:@"TEMPERATURA"];  
    NSString * cHUMEDAD = [[jsonArray objectAtIndex:i] objectForKey:@"HUMEDAD"];  
    NSString * cLUMINOSIDAD = [[jsonArray objectAtIndex:i] objectForKey:@"LUMINOSIDAD"];  
    NSString * cULTRAVIOLETA = [[jsonArray objectAtIndex:i] objectForKey:@"ULTRAVIOLETA"];  
    NSString * cRADIACION = [[jsonArray objectAtIndex:i] objectForKey:@"LLUVIA"];  
    NSString * cLLUVIA = [[jsonArray objectAtIndex:i] objectForKey:@"LLUVIAH"];  
    NSString * cCO2 = [[jsonArray objectAtIndex:i] objectForKey:@"CO2"];  
    NSString * cCO = [[jsonArray objectAtIndex:i] objectForKey:@"CO"];  
    NSString * cVOLTAJE = [[jsonArray objectAtIndex:i] objectForKey:@"VOLTAJE"];  
    NSString * cPOVOLTAJE = [[jsonArray objectAtIndex:i] objectForKey:@"POVOLTAJE"];  
    NSString * cHORA = [[jsonArray objectAtIndex:i] objectForKey:@"time"];  
    NSString * cTABLA = [[jsonArray objectAtIndex:i] objectForKey:@"TABLA"];  
  
    [datos1Array addObject:[[datos1 alloc] initWithTEMPERATURA:cTEMPERATURA andHUMEDAD:cHUMEDAD  
        andSOLAR:cRADIACION andCO:cCO andVOLTAJE:cVOLTAJE andPOVOLTAJE:cPOVOLTAJE andULTRAVIOLETA  
        cTABLA andLLUVIA:cLLUVIA andVIENTO:cTABLA andDIVIENTO:cTABLA]];  
}
```

Imprimir datos

```
(void) setLabels  
  
if([currentDato.TABLA isEqualToString:@"NODO 1"]){  
    dato.text= @"TEMPERATURA:";  
    datoLabel.text = currentDato.TEMPERATURA ;  
    d.text=@"°C";  
  
    dato1.text=@"HUMEDAD:";  
    dato1Label.text = currentDato.HUMEDAD;  
    d1.text=@"%RH";  
  
    dato2.text=@"LUMINOSIDAD:";  
    dato2Label.text = currentDato.LUMINOSIDAD;  
    d2.text=@"Lux";  
  
    dato3.text=@"CAN. LLUVIA(mm):";  
    dato3Label.text = currentDato.LLUVIA;  
    d3.text=@"mm";  
  
    dato4.text=@"CAN. LLUVIA(h):";  
    dato4Label.text = currentDato.SOLAR;  
    d4.text=@"mm";  
  
    dato5.text=@"VOLTAJE:";  
    dato5Label.text = currentDato.VOLTAJE;  
    d5.text=@"V";
```



2. Históricos

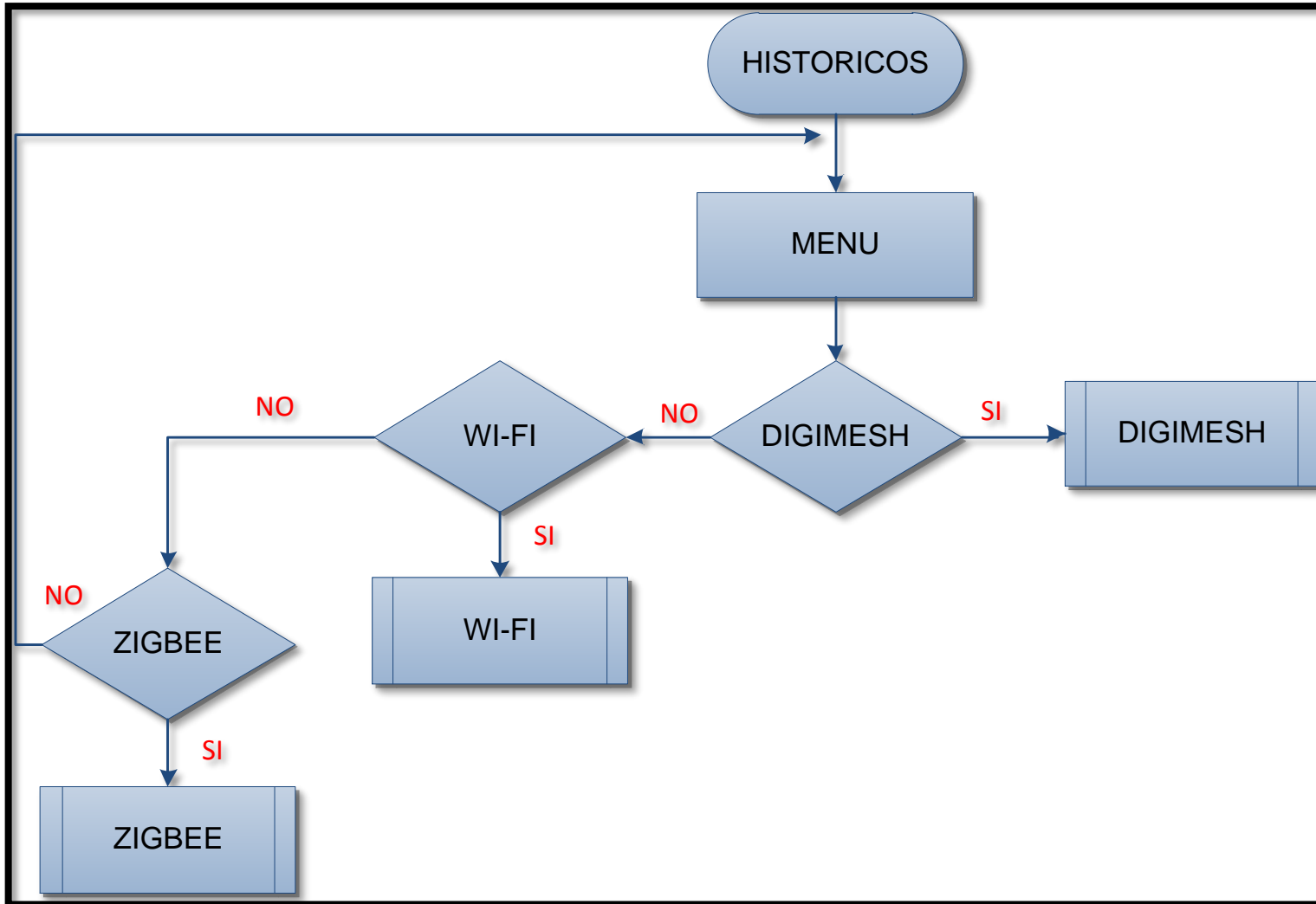


Diagrama de flujo red Wi-Fi

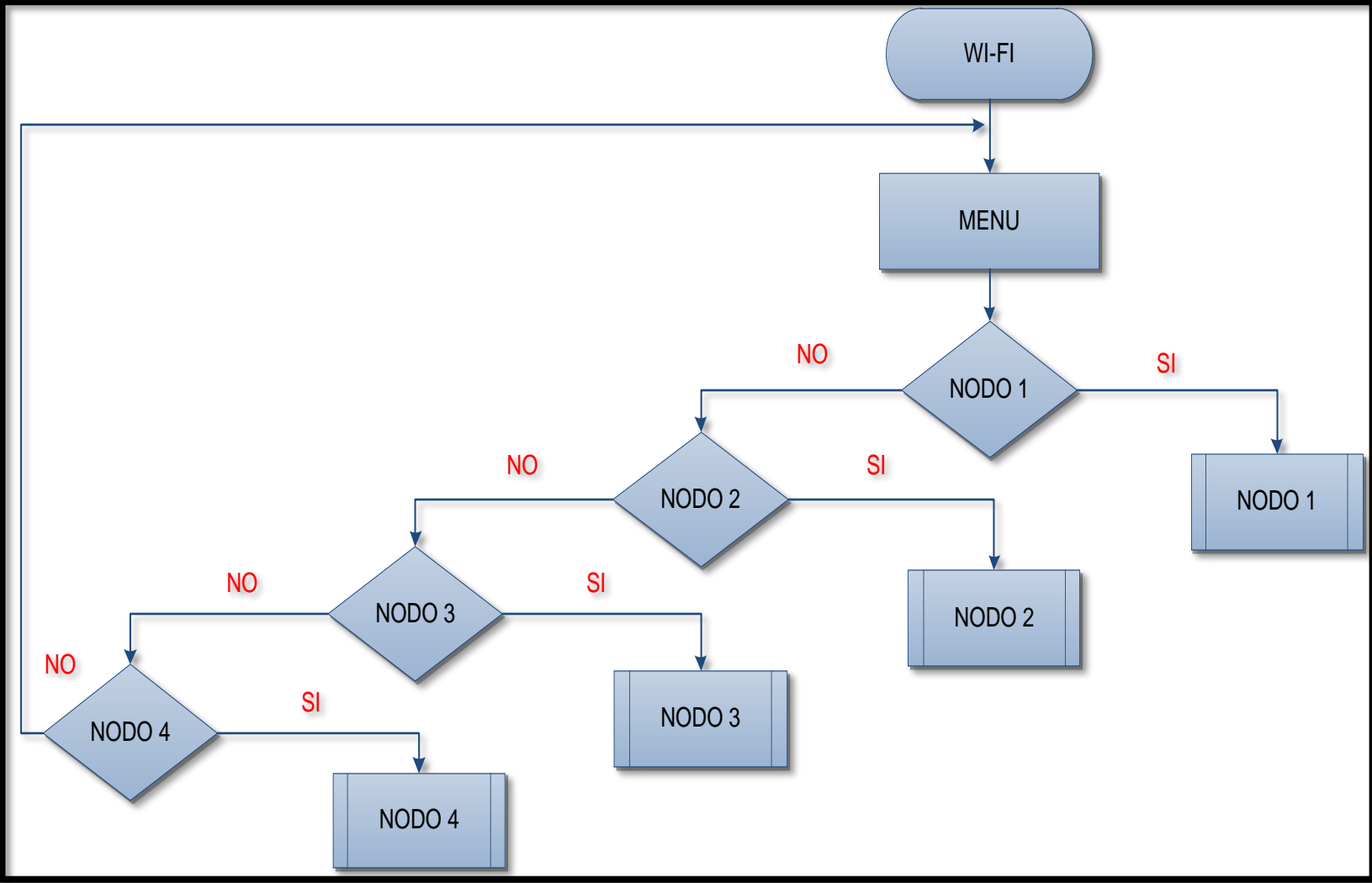


Diagrama de flujo del nodo1

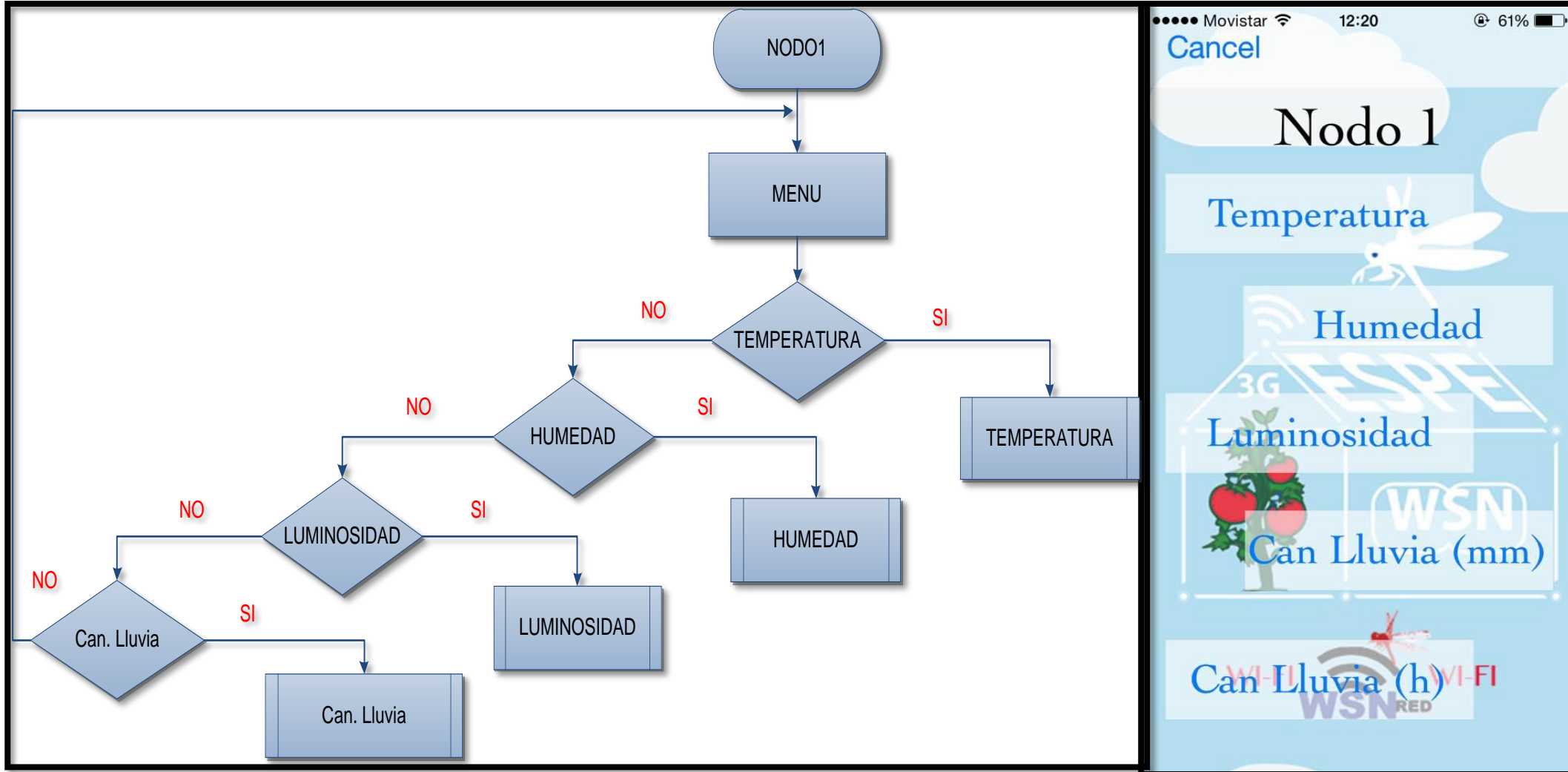
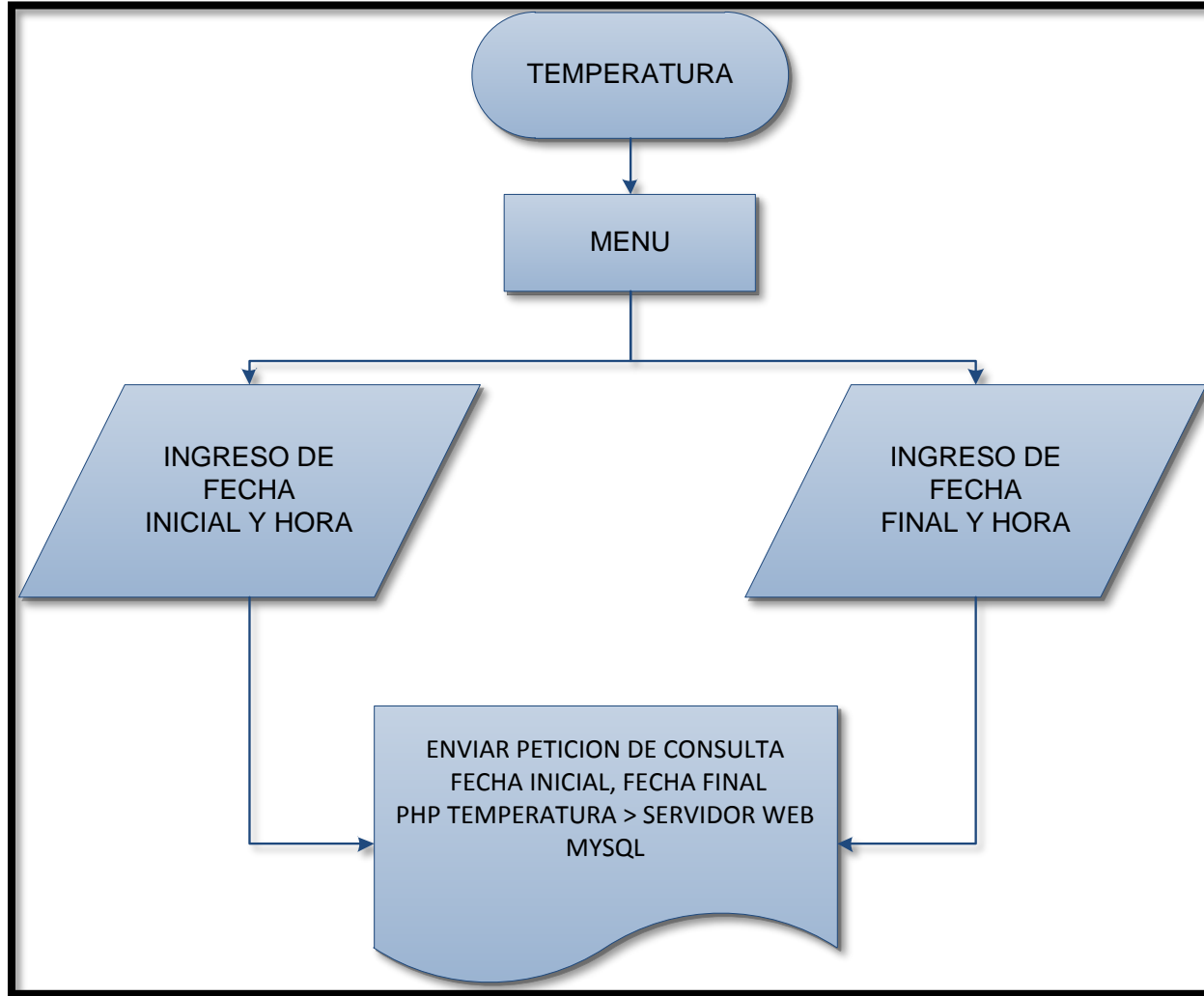


Diagrama de flujo de la variable Temperatura



Cancel Wi-fi

Nodo 1 **GRAFICA CONSULTA**
TEMPERATURA

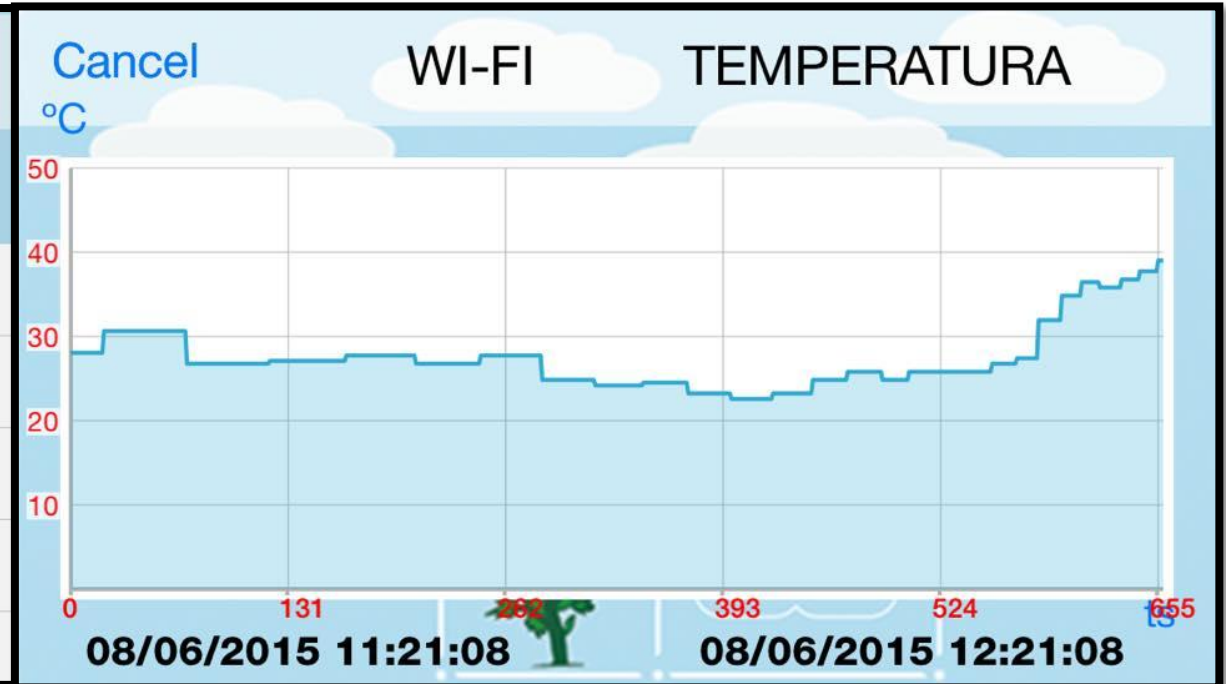
INICIO			FINAL		
FECHA	HORA		FECHA	HORA	
sáb 6 jun	10	18	sáb 6 jun	10	18
dom 7 jun	11	19	dom 7 jun	11	19
hoy	12	20	hoy	12	20
mar 9 jun	13	21	mar 9 jun	13	21
mié 10 jun	14	22	mié 10 jun	14	22



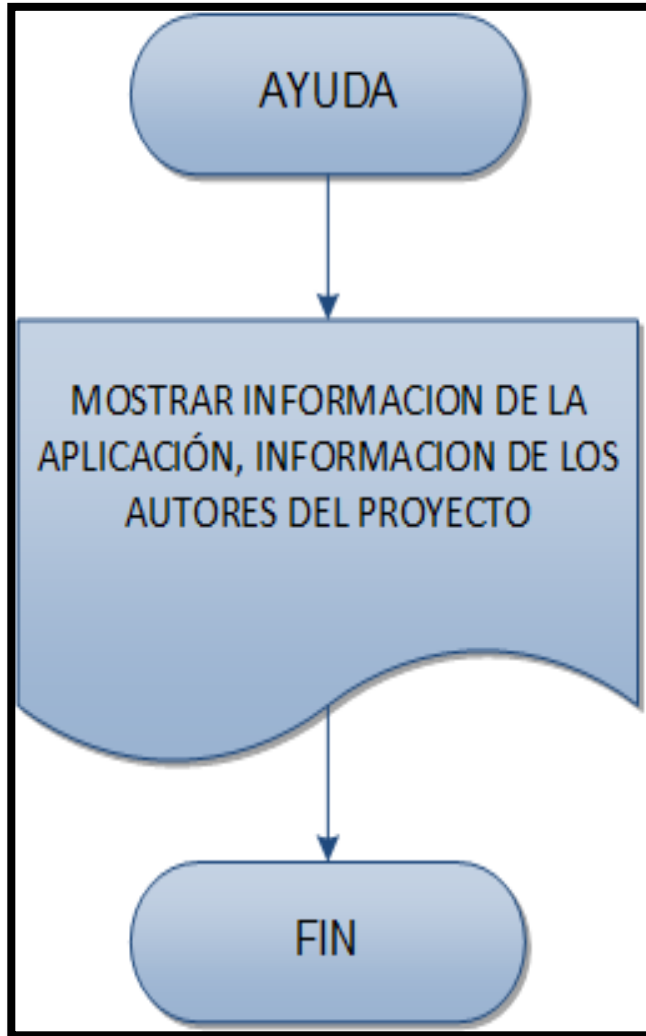
Histórico

Grafica

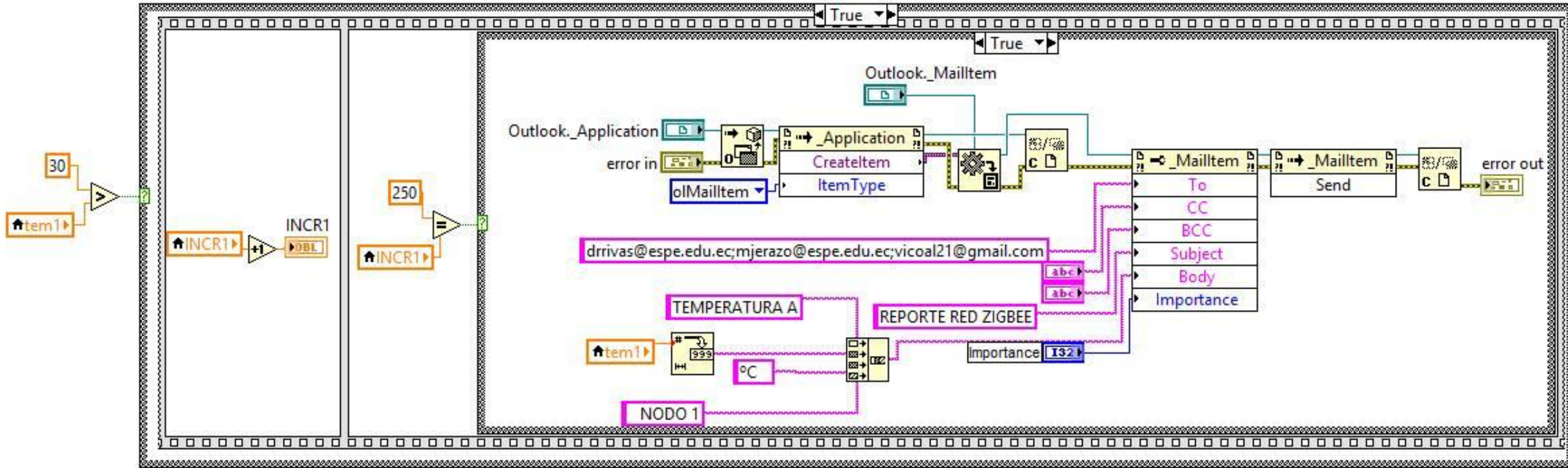
Cancel	Wi-fi
INICIO: 08/06/2015 11:20:49 FINAL: 08/06/2015 12:20:49	
TEMPERATURA	FECHA HORA
28.06 °c	08/06/2015 11:20:49
28.06 °c	08/06/2015 11:20:50
28.06 °c	08/06/2015 11:20:52
28.06 °c	08/06/2015 11:20:53
28.06 °c	08/06/2015 11:20:55



3. Ayuda



Reporte de alarmas por correo

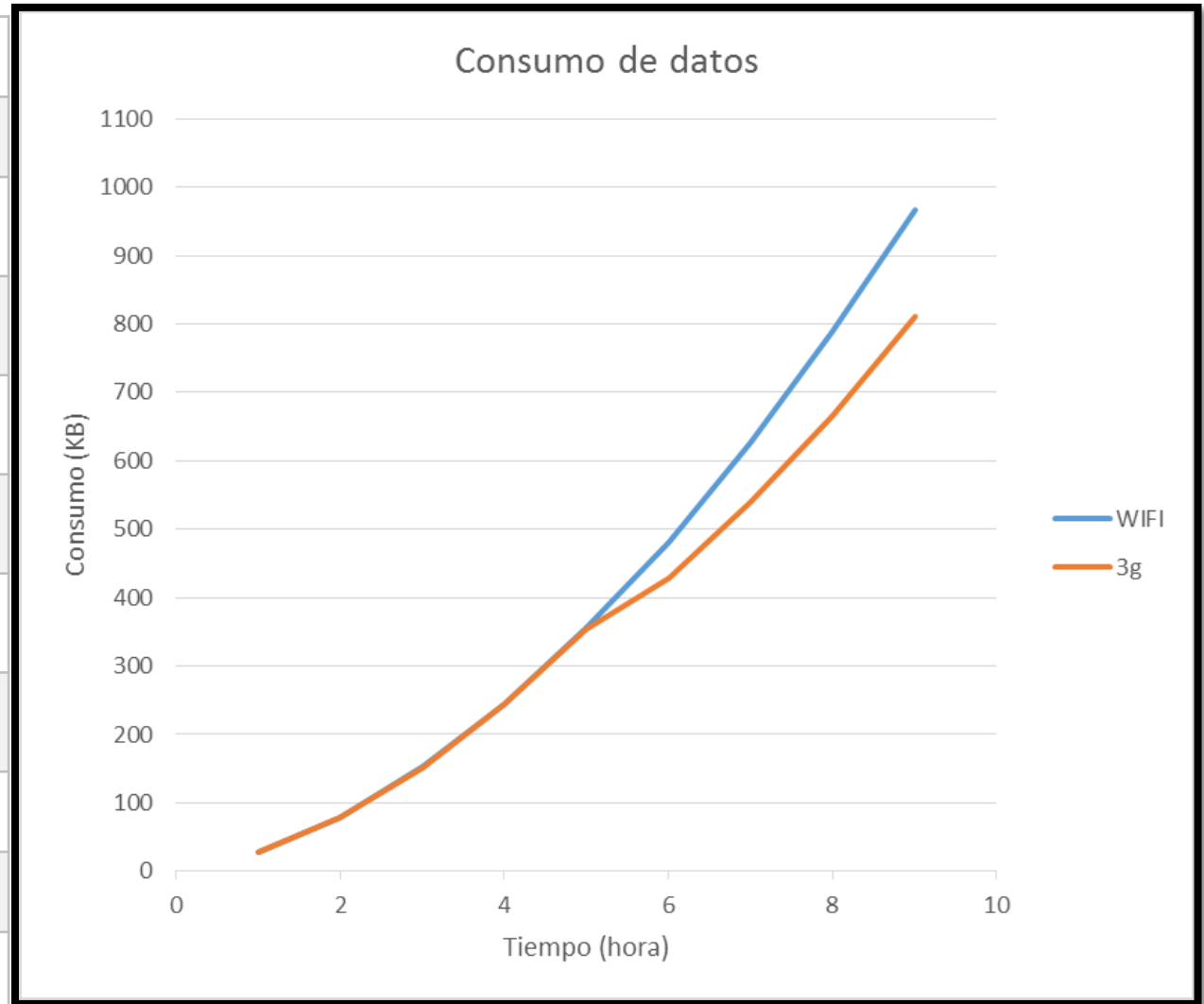


Análisis y Resultados



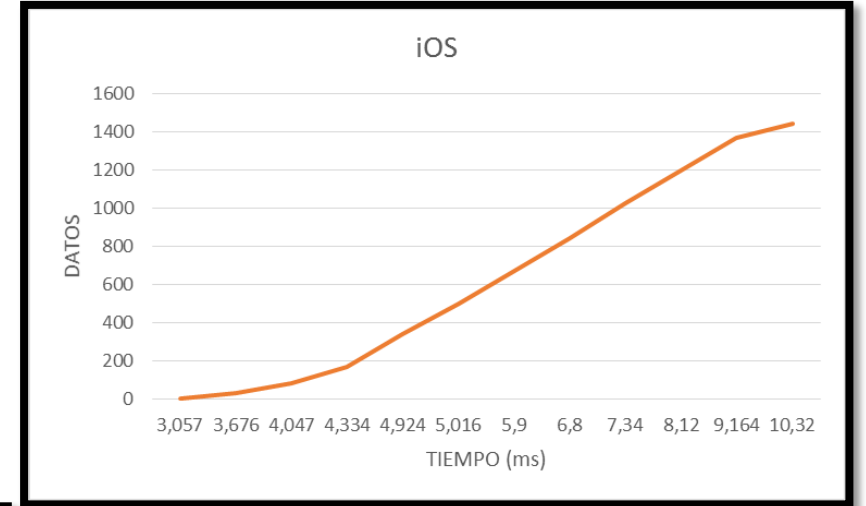
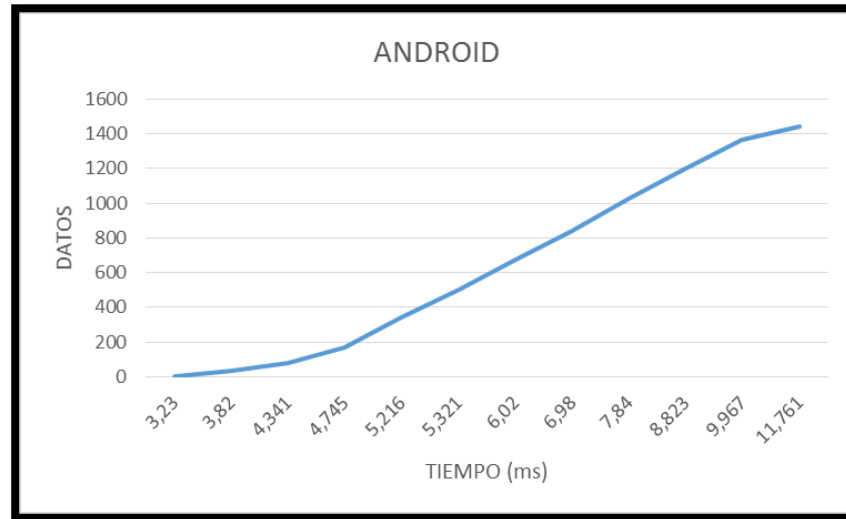
Consumo de datos

TIEMPO (h)	CONSUMO(KB)	
	WIFI	3G
1	28,11	27,95
2	79,07	78,6
3	152,42	151,62
4	245,03	244,92
5	355,62	354,2
6	482,06	428,49
7	627,91	539,13
8	789,8	665,89
9	966,89	810,94



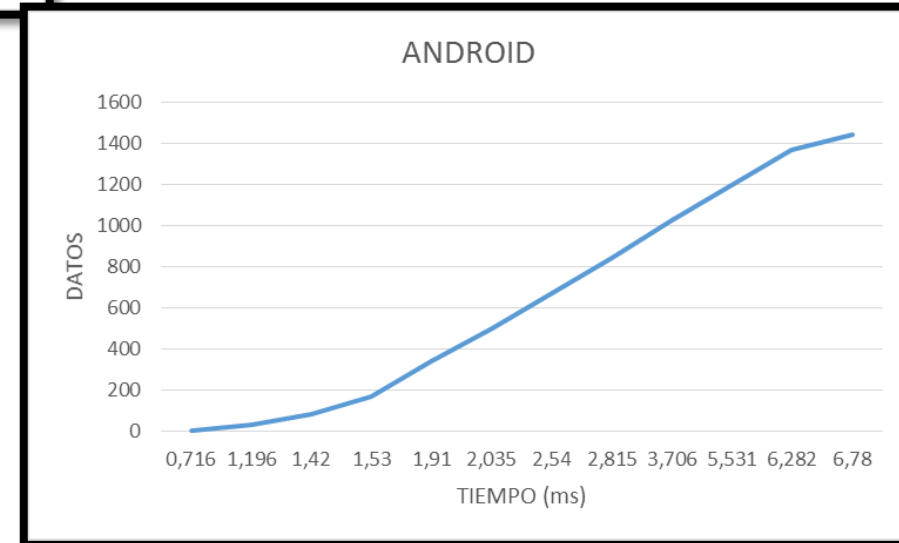
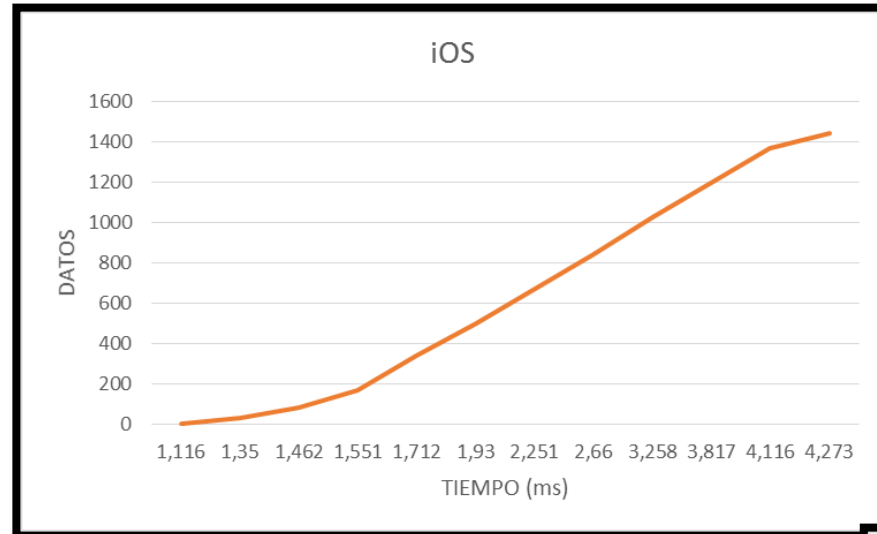
Tiempo de respuesta 3G

DATOS	RESPUESTA(ms)	
	ANDROID	iOs
3	3,23	3,057
31	3,82	3,676
81	4,341	4,047
171	4,745	4,334
339	5,216	4,924
498	5,321	5,016
672	6,02	5,9
843	6,98	6,8
1026	7,84	7,34
1197	8,823	8,12
1368	9,967	9,164
1439	11,761	10,32



Tiempo de respuesta WIFI

DATOS	RESPUESTA(ms)	
	ANDROID	iOS
3	0,716	1,116
31	1,196	1,35
81	1,42	1,462
171	1,53	1,551
339	1,91	1,712
498	2,035	1,93
672	2,54	2,251
843	2,815	2,66
1026	3,706	3,258
1197	5,531	3,817
1368	6,282	4,116
1439	6,78	4,273



Conclusiones

- Se cumplió con el objetivo propuesto en este proyecto, el cual consiste en el desarrollo de un sistema HMI en las plataformas Android e iOS para el monitoreo de la red inalámbrica de sensores agrícolas ubicada en el barrio Rumipamba de Navas cantón Salcedo, provincia de Cotopaxi.
- La aplicación desarrollada en iOS utiliza menor código de programación que la creada en Android; cumpliendo ambas con las mismas funcionalidades.
- La infraestructura de la comunicación de datos desplegada permite que dispositivos móviles con tecnología (Wifi, 3G, 4G). pueden acceder, desde cualquier parte del mundo a la aplicación creada.



Conclusiones

- El consumo de datos en redes móviles está directamente relacionado con el volumen de datos a ser descargados desde de la base, por tal razón a fin de no saturar el servidor se limitó la descarga directa de datos del día, y solo se accederá a datos anteriores si el usuario lo requiere.
- La velocidad de trasmisión de la tecnología de datos móvil está limitado por su área de cobertura e infraestructura de comunicación de la operadora que presta el servicio.
- El análisis de las pruebas realizadas determinó que el tiempo y el consumo de descarga de datos, en la plataforma iOS es más eficiente por su arquitectura cerrada en hardware y software a diferencia de Android al ser una arquitectura abierta.
- El desarrollo del presente proyecto es una iniciativa que para los productores agrícolas a gran escala que implementen un monitoreo móvil mediante los dispositivos Smartphone que permita la visualización del comportamiento de las variables climatológicas que interfieren en la producción agrícola.



Conclusiones

- En la actualidad Epsilon es la única empresa en la región de Latinoamérica que comercializa equipos para procesos agrícolas, su interfaz de monitoreo se restringe a ser visualizado vía web, no disponen de aplicaciones móviles que permitan el monitoreo de dichos procesos.
- La ventaja principal de utilizar software libre es que su costo de licencia es nulo, a diferencia del software privado que necesita de una licencia altamente costosa para poder desarrollar aplicaciones.



Recomendaciones

- El trabajo a futuro en este proyecto sería realizar la etapa de control mediante dispositivos Smartphone y de esta manera conseguir un mayor impacto en la agricultura inteligente.
- Para poder acceder al servidor web desde cualquier lugar geográfico, se requiere conocer su dirección web configurada por un dominio privado.
- En la plataforma Android requiere de una versión de Android 4.0.3 o superior, en cuanto la plataforma iOS requiere de una versión iOS 8.1 o superior.
- Al desarrollar la programación en Eclipse es necesario asignar los permisos de conexión a internet en el archivo manifest.



Recomendaciones

- Para crear un proyecto nuevo de Android e iOS es importante indicar con que versiones va a ser compatible la aplicación, y que estas se encuentre actualizadas mediante el SDK.
- Para depurar la aplicación desarrollada en iOS, se realizó JailBreak al software de desarrollo Xcode y al dispositivo Smartphone.

**GRACIAS POR SU
ATENCIÓN**



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA