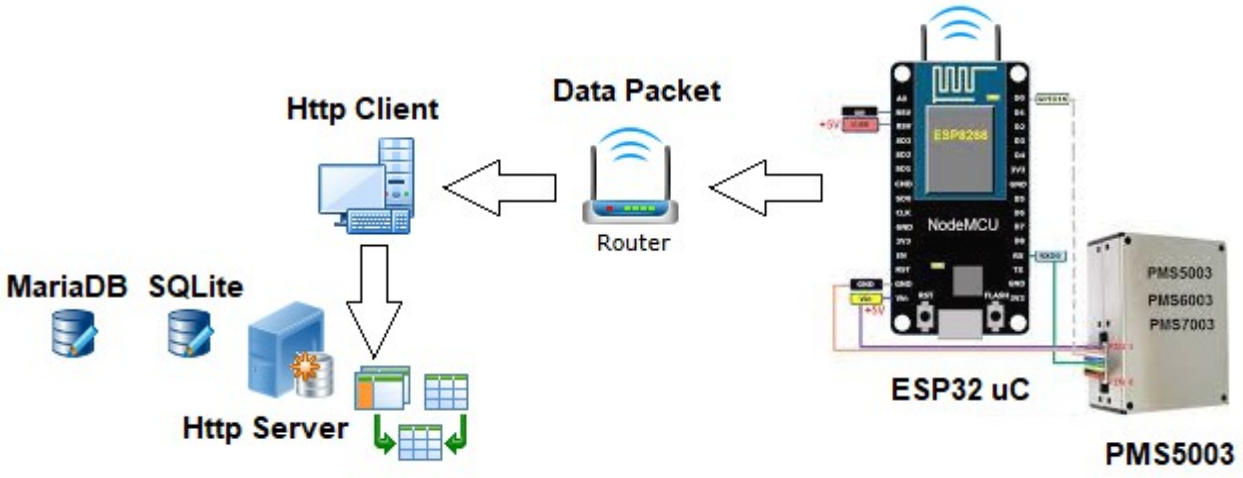
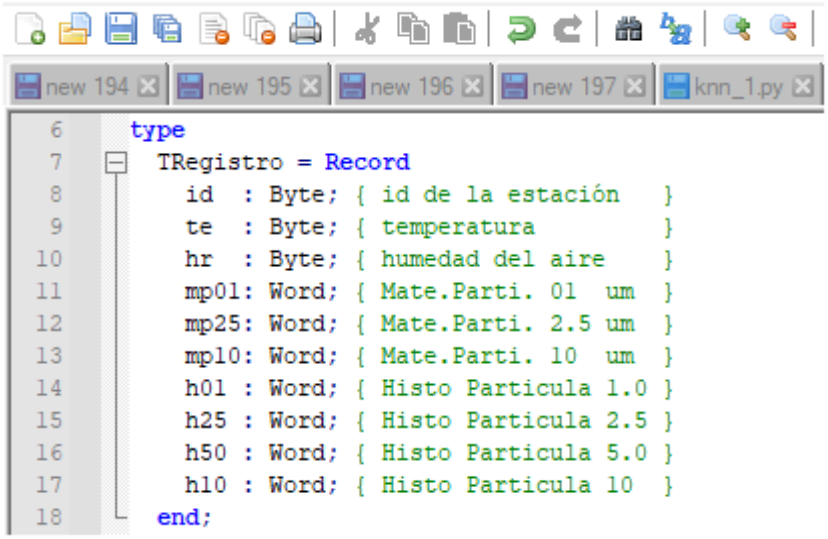
Proyecto #2 – Sistemas Inteligentes
INFO1157
By Alberto Caro

1.- Imagine que usted es un ingeniero civil informático con una especialidad en *Internet de las Cosas* (**IoT**) y *Sistemas Inteligentes* (**Sistemas Embebidos + Sensores**). Hay un problema de **DataLogger** que debe ser resuelto y se necesita que usted desarrolle las aplicaciones del **Server** como del **Cliente** remoto. La empresa necesita que el protocolo de envío de los datos se realice mediante **HTTP Server** y **HTTP Client** en **Mode Console/Terminal** utilizando el famoso y poderoso lenguaje de programación **Lázarus Pascal**. Se dispone de un esquema que aclara mas el desafío:



Usted dispone del archivo binario **data.dat** que fue generado por el micro **ESP32** y enviado al host por **WIFI**. La estructura del archivo es la siguiente:




```
type
  TRegistro = Record
    id  : Byte; { id de la estación   }
    te  : Byte; { temperatura         }
    hr  : Byte; { humedad del aire    }
    mp01: Word; { Mate.Parti. 01  um  }
    mp25: Word; { Mate.Parti. 2.5 um  }
    mp10: Word; { Mate.Parti. 10  um  }
    h01 : Word; { Histo Particula 1.0 }
    h25 : Word; { Histo Particula 2.5 }
    h50 : Word; { Histo Particula 5.0 }
    h10 : Word; { Histo Particula 10  }
  end;
```

Este archivo es de tipo estructurado el cual almacena los sampleos de **10** estaciones ambientales diferentes. Este archivo tiene todos los datos que el **HTTP Cliente** debe enviar al **HTTP Server** para ser procesados.

Dado el contexto anterior, usted debera realizar lo siguiente:

1. Programar un **HTTPClient** (Método **POST**) que envíe todos los datos que están almacenados en **data.dat** al **HTTPServer** que los almacenará en: **SQLITE** y **MariaDB**. El envío de los datos se deben configurar según:

    1. **Datos tipo JSON** → Content/Type = 'application/json'
    2. **Datos tipo Stream** →Content/Type = 'application/octet-stream'

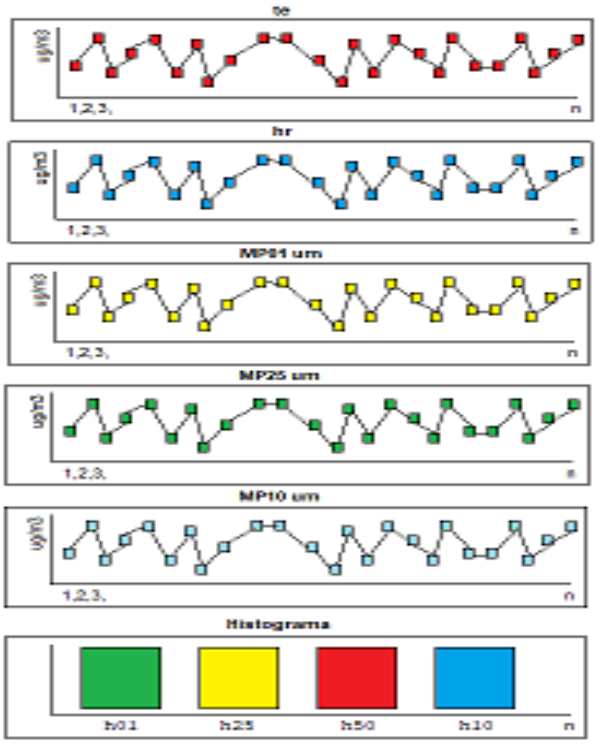
    Es decir, los datos de **data.dat** se deben enviar primero en formato **JSON**. Despues enviar en formato **STREAM** (Conjunto de Bytes, **TMemoryStream**). Debe programar **HTTPServer** y **HTTPClient**.

2. Programar un **HTTPCliente** (Método **POST**) que envie todos los datos que están almacenados en **data.dat** a un **Flask WEB Server**. **Flask** almacenará los datos en **SQLITE** y generará con **pyplot** un **fig , axs = plt.subplots(.)** y lo exportará como un **.PNG**. Cada gráfica samplea los valores de **te, hr, mp01, mp25 y mp10** de manera separada.

    En la **6ta** fila se grafica el **Histograma** de **h01, h25, h50 y h10**. Todas las gráficas con **label**, **unidades de medida**, **títulos** y **Grilla ON**. Además, cada un de las **5** series anteriores deben mostrar una gráfica de **promedio móvil con ventana** (Investigar) de **10** datos históricos.

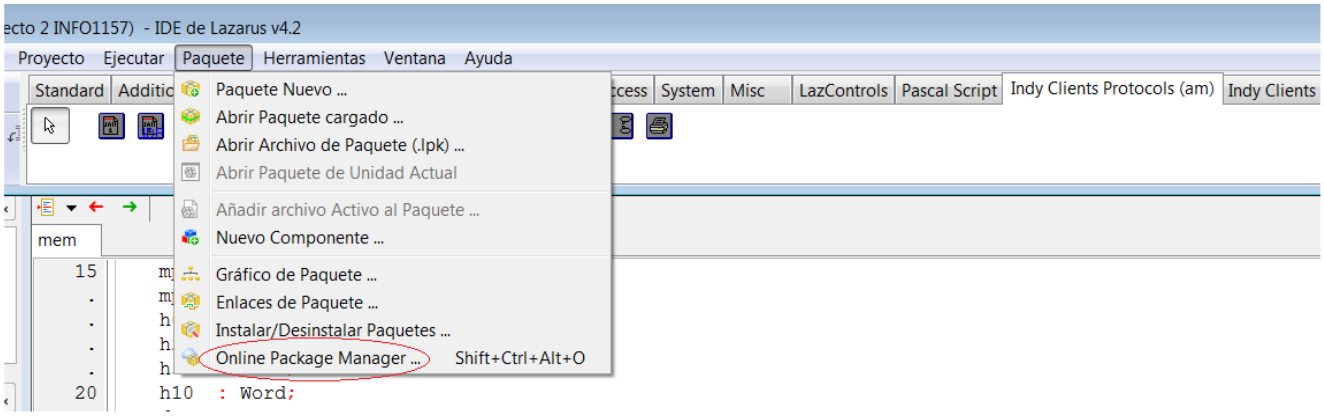
    El **HTTPServer** y **HTTPCliente** se programan en **Mode Consola**. No se aceptan desarrollos de los **HTTPServer** y **Client** utilizando **GUI**.

    Gráfica de **Pyplot** debería vserse así o algo similar:



Conceptos y temas a investigar en Lazarus (Indy Components)

- TidHTTPServer
- TidHTTPClient
- Eventos OnCommandGet
- ARequestInfo.Command
- ARequestInfo.PostStream
- ARequestInfo.PostStream.Position
- ARequestInfo.PostStream.Size
- ARequestInfo.PostStream.ReadBuffer
- ATResponseInfo.ContentText
- ATResponseInfo.ResponseNo
- AContext: TidContext
- TMemoryStream
- TMemoryStream.LoadFromFile
- TMemoryStream.Read
- TMemoryStream.ReadBuffer
- TMemoryStream.WriteBuffer
- TMemoryStream.Free
- TMemoryStream.SaveToFile



--- << Investigar cómo se instalan Paquetes en Lazarus Pascal >> ---

Observaciones

- Trabajo Grupo de 2 o Individual.
- Defensa y presentación fecha y hora por confirmar.