

IoT: Medidor de Temperatura

Entrega 2

27/05/2016

**Universidad Tecnológica Nacional**

**Arquitectura de Software**

**IoT: Medidor de Temperatura**

Arquitectura de Software

Integrantes

* Appiano, Fabrizio 50497
* Mena, José Ignacio 60507
* Pattusi, Rodrigo 50590
* Ryser, Carlos 60420

Contenido

[1. Código fuente versionado en Github. 2](#_Toc452133741)

[2. Definición de Requerimientos y Supuestos 2](#_Toc452133742)

[Requerimientos Funcionales 2](#_Toc452133743)

[Supuestos 2](#_Toc452133744)

[Predicción de Evolución de la Arquitectura 2](#_Toc452133745)

[3. Definición de Requerimientos No Funcionales 2](#_Toc452133746)

[Requerimientos no funcionales 2](#_Toc452133747)

[4. Diagrama de Arquitectura 3](#_Toc452133748)

[5. Dependencias 4](#_Toc452133749)

[Configuración de Desarrollo 4](#_Toc452133750)

[Configuración de Testing 5](#_Toc452133751)

[Configuración de Producción 5](#_Toc452133752)

## Código fuente versionado en Github.

[Repositorio Github](https://github.com/rodrigogit45/ADS2016) (https://github.com/mappiano/ADS2016.git)

## Definición de Requerimientos y Supuestos

### Requerimientos Funcionales

* Tomar mediciones de temperatura ambiental
* Informar de temperaturas máximas y mínimas alcanzadas
* Los límites de temperaturas deben poder ser configurables

### Supuestos

* Se dará soporte a los siguientes web browser Chrome (versión 5.0 en adelante, Firefox (versión 3.2 en adelante), Edge (12.0 en adelante).
* Los navegadores antes mencionados tendrán activado el intérprete de JS.

### Predicción de Evolución de la Arquitectura

* Estimamos que en futuras versiones se necesitará almacenar los datos de temperaturas tomadas, en un lugar centralizado o tener algún mecanismo de backup.
* La aplicación deberá considerar la posibilidad de que los navegadores no soportan JS y no permitan el uso de Cookies.

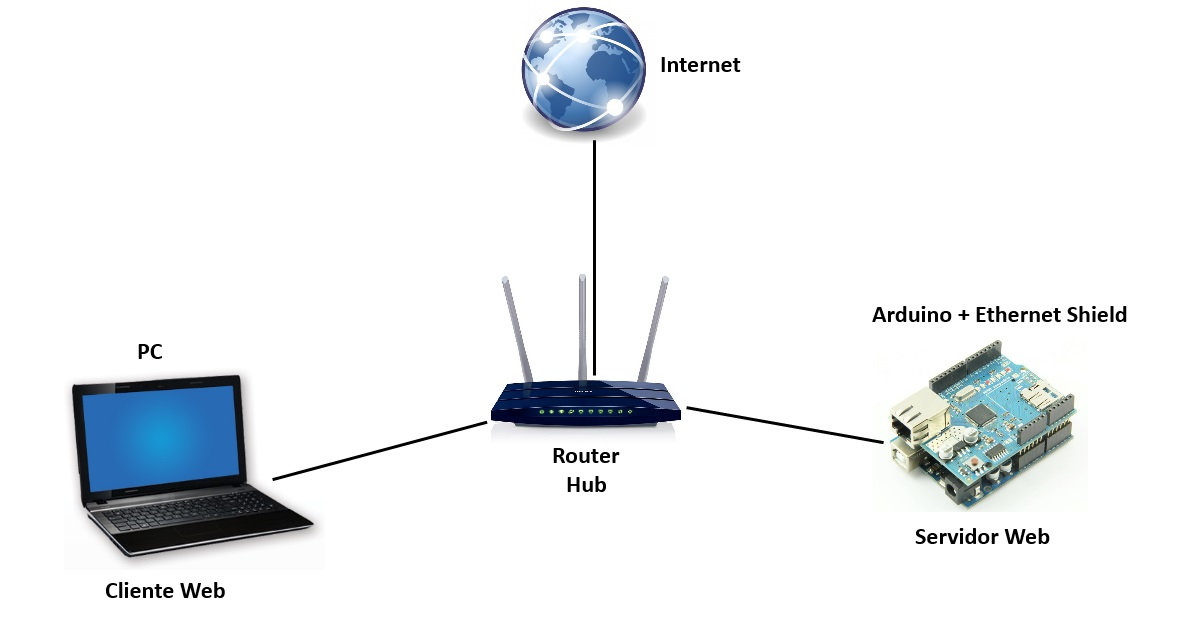
## Definición de Requerimientos No Funcionales

### Requerimientos no funcionales

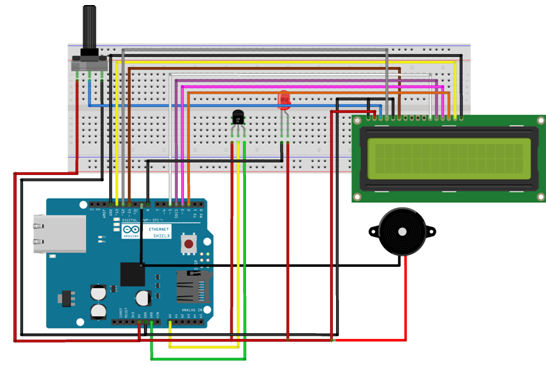
* La medición de la temperatura deberá ser realizada a través de una placa compatible con Arduino.
* Las mediciones deberán ser visualizadas a lo largo del tiempo en un gráfico de barras.
* El servidor web deberá estar dentro de la placa.
* Las mediciones de temperaturas serán realizadas cada 5 segundos.
* Debe brindarse una notificación sonora cuando la temperatura máxima o mínima sea marcada por el sensor de temperatura.

Para lograr el RNF 1 y RNF 2, la arquitectura poseerá una capa de interfaz de usuario independiente del resto del sistema que estará encargada de renderizar los gráficos necesarios para visualizar la información.

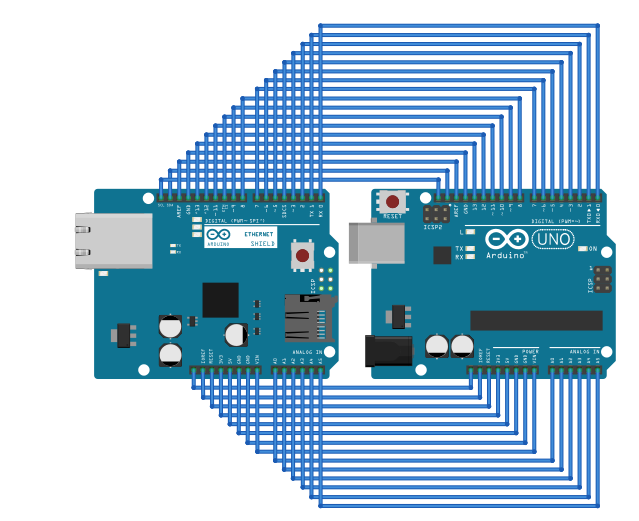
## Diagrama de Arquitectura



Plano del Circuito Ethernet con sensor de temperatura LM35, amplificador y Display de 16 pines.



Conexión Arduino Uno R3 con Arduino Sheld Ethernet R3.



## Dependencias

### Configuración de Desarrollo

#### Lenguajes de programación utilizados

* **C:** este lenguaje nos permite trabajar sobre el código que interpreta la placa Arduino.
* **JavaScript:** es el lenguaje que interpretan los navegadores web y en este caso nos servirá para programar la interfaz del usuario.
* **HTML** **5:** no es un lenguaje propiamente dicho, será utilizado para diseñar la página web donde se mostrarán todas las muestras de temperatura.

#### Herramientas de desarrollo

* **Sublime:** es un editor de texto que permite la escritura de código de múltiples lenguajes. es muy sencillo de utilizar y tiene pocos requerimientos.

#### Librerías de terceros

* **Socket.io**
* **Express**
* **SerialPort**

Entornos de ejecución

* **SQL Server 2014**: esta base de datos relacional nos permitirá almacenar los datos de las temperaturas tomadas por el sensor conectado a la placa Arduino.
* Cualquier navegador web que soporte las últimas actualizaciones de JS, HTML5 y   CSS3.
* Placa Arduino.

### Configuración de Testing

Igual a entorno de desarrollo.

### Configuración de Producción

Igual a entorno de desarrollo con entornos de ejecución

#### Entornos de Ejecución

* Cualquiera de los navegadores web que sean compatibles con las actualizaciones de HTML5, CSS3 y JavaScript.
* Motor de Base de Datos SQL Server 2014.

#### Plan de Seguimiento de Dependencias

Como no se poseen dependencias importantes fuera de la base de datos y las librerías JS se revisarán las dependencias cada  15 días.

#### Plan de Contingencia de Dependencias

* **Sublime**: En caso que nuestra licencia sea suspendida, el código se escribirá en otro editor como WebStorm por ejemplo, el cual podemos contar con licencia gratuita por ser estudiantes.
* **Sql Server 2014**: En caso de que nuestra licencia sea suspendida de manera imprevista, migraremos a MySql que es una base de datos Open Source.