

**Universidad Autónoma Metropolitana Unidad Azcapotzalco**  
**División de Ciencias Básicas e Ingeniería**  
**Licenciatura en Ingeniería en Computación**

*Aplicación concurrente para la obtención y almacenamiento de datos a partir de sensores*

Proyecto tecnológico

Pedro Romero Flores  
2113002325  
al2113002325@correo.azc.uam.mx

Asesor:

José Alejandro Reyes Ortiz  
Profesor Asociado  
Departamento de Sistemas

Co-asesor:

Leonardo Daniel Sánchez Martínez  
Profesor Asociado  
Departamento de Sistemas

Trimestre 16 Otoño

Martes 13 diciembre de 2016

Yo, José Alejandro Reyes Ortiz, declaro que aprobé el contenido del presente Reporte de Proyecto de Integración y doy mi autorización para su publicación en la Biblioteca Digital, así como en el Repositorio Institucional de UAM Azcapotzalco.



---

Firma de asesor

Yo, Leonardo Daniel Sánchez Martínez, declaro que aprobé el contenido del presente Reporte de Proyecto de Integración y doy mi autorización para su publicación en la Biblioteca Digital, así como en el Repositorio Institucional de UAM Azcapotzalco.



---

Firma de co-asesor

Yo, Pedro Romero Flores, doy mi autorización a la Coordinación de Servicios de Información de la Universidad Autónoma Metropolitana, Unidad Azcapotzalco, para publicar el presente documento en la Biblioteca Digital, así como en el Repositorio Institucional de UAM Azcapotzalco.



---

Firma de alumno

## **RESUMEN**

La realización de esta propuesta es con el objetivo de capturar información de sensores de humedad, temperatura e intensidad luminosa de forma concurrente. Con los datos de los sensores se realiza el poblado de una ontología. Este proyecto contiene el diseño e implementación de módulos para: la captura de datos de sensores de forma concurrente, para el almacenamiento de datos en la ontología, para la visualización de datos de sensor.

El módulo de lectura se comunica con sensores para obtener su información contenida en un documento HTML, procesa el documento para extraer datos específicos y transmite la información procesada al módulo de escritura. Además recibe peticiones del módulo de visualización para extraer datos de sensores de una ubicación específica y mostrarlos a un usuario a través de una interfaz gráfica. El módulo de escritura, almacena los datos provenientes del módulo de lectura. La comunicación de los módulos desarrollados con los sensores es a través de una placa Arduino.

# CONTENIDO

RESUMEN.....	i
1. INTRODUCCIÓN.....	1
2. ANTECEDENTES.....	1
2.1 Artículos .....	1
2.2 Tesis de otras universidades .....	1
2.3 Proyectos terminales .....	2
3. JUSTIFICACIÓN.....	3
4. OBJETIVOS .....	3
4.1 Objetivo general .....	3
4.2 Objetivos específicos.....	3
5. MARCO TEÓRICO .....	4
5.1 Hardware Arduino.....	4
5.1.1 Sistemas Electrónicos .....	4
5.1.2 Arduino.....	5
5.2 Modelo ontológico.....	6
5.3 Sockets.....	8
5.3.1 Arquitectura Cliente Servidor .....	8
5.3.2 Socket servidor.....	9
5.3.3 Socket cliente .....	9
5.4 Concurrencia .....	9
6. DESARROLLO DEL PROYECTO.....	10
6.1 Sensores .....	10
6.1.1 Sensor de humedad .....	10
6.1.2 Sensor de temperatura .....	10
6.1.3 Sensor de intensidad luminosa.....	10
6.2 Módulo de lectura concurrente.....	12
6.2.1 Captura.....	13
6.2.2 Procesamiento.....	13
6.2.3 Transmisión.....	15
6.2.4 Módulo de visualización.....	16
6.3 Módulo de escritura concurrente.....	17
6.3.1 Servidor concurrente .....	17

6.3.2	Archivos temporales .....	18
6.3.3	Lectura de archivos temporales.....	19
6.3.4	Inserción en el modelo ontológico.....	19
7.	RESULTADOS .....	21
7.1	Resultado prueba 1 .....	21
7.2	Resultado prueba 2 .....	23
7.3	Resultados prueba 3.....	26
8.	ANÁLISIS Y DISCUSIÓN DE RESULTADOS.....	31
9.	CONCLUSIONES .....	33
10.	ANEXOS .....	33
10.1	ANEXO A: Código fuente de un servidor Web Arduino .....	33
10.2	ANEXO B: Módulo de lectura concurrente.....	36
10.2.1	Código fuente de socket servidor de lectura concurrente.....	36
10.2.2	Código fuente de captura concurrente de datos de sensores .....	37
10.2.3	Código fuente de hilo para captura de datos de sensor .....	38
10.2.4	Código fuente de procesamiento de datos.....	39
10.3	ANEXO C: Código fuente de módulo de visualización .....	44
10.4	ANEXO D: Módulo de escritura concurrente.....	48
10.4.1	Código fuente de socket servidor de escritura concurrente.....	48
10.4.2	Código fuente de hilo para creación de archivos temporales .....	49
10.4.3	Código fuente de lectura secuencial de archivos temporales.....	50
10.4.4	Código fuente de métodos para administrar ontologías con Java .....	52
10.4.5	Código fuente de inserción en ontología.....	53
10.5	ANEXO E: Modelo de datos semántico OntologyDevices.....	55
11.	REFERENCIAS BIBLIOGRÁFICAS .....	84

## ÍNDICE DE FIGURAS

Figura 1 Diagrama de sistema electrónico. ....	5
Figura 2 Comparativa entre placas Arduino. ....	5
Figura 3 Interfaz del IDE Arduino. ....	6
Figura 4 Funcionamiento general de sockets. ....	8
Figura 5 Relación entre procesos e hilos. ....	9
Figura 6 Arquitectuta implementada para Aplicación concurrente para la obtención y almacenamiento de datos a partir de sensores. ....	10
Figura 7 Conexiones entre placa Arduino, sensores y SHIELD WIFI 101. ....	11
Figura 8 Página Web resultado de la respuesta del servidor Web Arduino a la petición de un cliente. ....	11
Figura 9 Ejemplo del código de una página HTML resultado de una petición a un servidor Web Arduino. ....	12
Figura 10 Esquema del funcionamiento del módulo de lectura concurrente. ....	13
Figura 11 Estructura de la información de un sensor dentro de una página HTML. ....	15
Figura 12 Ejemplo de extracción de datos de una página HTML. ....	15
Figura 13 Interfaz del módulo de visualización. ....	16
Figura 14 Esquema del funcionamiento del módulo de escritura concurrente. ....	17
Figura 15 Carpeta archivos_temporales. ....	18
Figura 16 Contenido del archivo temporal "temporal278319673424372721". ....	18
Figura 17 Ejemplo de extracción de los elementos. ....	19
Figura 18 Taxonomía de Device. ....	20
Figura 19 Modulo de visualización, prueba de servidor Web Arduino. ....	21
Figura 20 Inserción de datos del individuo DHT11A en la ontología. ....	22
Figura 21 Inserción de datos del individuo DHT11B en la ontología. ....	22
Figura 22 Inserción de datos del individuo LightA en la ontología. ....	23
Figura 23 Inserción de datos del individuo DHT11A en la ontología. ....	23
Figura 24 Módulo de visualización, prueba de servidor Web Apache. ....	24
Figura 25 Inserción de datos del individuo DHT11A en la ontología. ....	24
Figura 26 Inserción de datos del individuo DHT11B en la ontología. ....	25
Figura 27 Inserción de datos del individuo LightA en la ontología. ....	25
Figura 28 Inserción de datos del individuo LightB en la ontología. ....	26
Figura 29 Archivo de configuración direcciones.txt ....	26

Figura 30 Datos capturados de diferentes servidores Web por el módulo de lecturas concurrente.....	27
Figura 31 Inserción de datos de diferentes servidores Web al individuo DHT11A en la ontología.....	28
Figura 32 Inserción de datos diferentes servidores Web al individuo DHT11B en la ontología.....	29
Figura 33 Inserción de datos diferentes servidores Web al individuo Light A en la ontología .....	30
Figura 34 Inserción de datos diferentes servidores Web al individuo Light B en la ontología. ....	31

## ÍNDICE DE TABLAS

Tabla 1 Relación de etiquetas con ontología. ....	14
Tabla 2 Individuos y propiedades de la subclase sensor. ....	20
Tabla 3 Comparación entre resultados. ....	32



# 1. INTRODUCCIÓN

Los entornos inteligentes son espacios físicos, como habitaciones, lugares de trabajo, hogares, etc. en donde existen redes de sensores y elementos computacionales interactuando entre sí. Un entorno inteligente puede automatizar servicios de energía eléctrica, seguridad y bienestar a través de diferentes dispositivos que reaccionan ante acciones de un usuario o diferentes parámetros ambientales; con el objetivo de hacer transparente su interacción con la tecnología; es decir, que diferentes dispositivos tecnológicos interactúen con el usuario sin que él lo perciba.

Se han utilizado diferentes tecnologías en redes de sensores para la captura de variables ambientales y sensores de presencia de objetos obteniendo buenos resultados. La llegada del Internet, ha permitido que los componentes de los entornos inteligentes se puedan distribuir en diferentes zonas geográficas y comunicarse a través de la Web.

Las ontologías, en el dominio de la informática, son una tecnología que sirve para la representación de información de manera clara y, sobre todo, procesable por computadoras. Para los entornos inteligentes, las ontologías juegan un papel importante ya que se pueden utilizar como mecanismo de almacenamiento de la información y con la capacidad de realizar inferencia sobre ella.

Por lo tanto, se propone crear componentes de software que interactúen con múltiples sensores, con el objetivo de capturar y almacenar la información en una ontología para su posterior procesamiento y hacer inferencia con ella en un ambiente inteligente. (Actualmente se encuentran en una fase final de desarrollo y pronto serán ubicados en diferentes cubículos del departamento de Sistemas de la UAM AZC)

## 2. ANTECEDENTES

### 2.1 Artículos

- Control Automático de Condiciones Ambientales en Domótica usando Redes Neuronales Artificiales.[1]

El objetivo fundamental es similar, capturar información relacionada al ambiente a través de sensores: variables de luz, calefacción además de la ventilación y uso de puerta y ventanas de una habitación con el fin de entregar esta información para una red neuronal, para el reconocimiento de patrones de uso de dichas variables. A diferencia de ese trabajo, este proyecto busca obtener y almacenar de forma concurrente en una ontología información de variables similares de diferentes habitaciones para después extraer conocimiento.

### 2.2 Tesis de otras universidades

- Estudio de la plataforma Android para dispositivos móviles y desarrollo de aplicación para la administración de redes de sensores inalámbricos.[2]

Se obtiene la información de una base de datos a través de Internet correspondiente a las coordenadas geográficas de los sensores, y se puede acceder a ellos para monitorear diferentes parámetros. La idea es similar a esta propuesta, se propone recolectar parámetros publicados en diferentes sitios Web por diferentes sensores a través de Internet, que se mostraran en una aplicación cliente que utilizará servicios Web y se almacenaran en una ontología.

Uso del protocolo CoAP para la implementación de una aplicación domótica con redes de sensores inalámbricas.[3]

Es una tesis que propone implementar la automatización de varios dispositivos en el hogar, que reaccionen a los parámetros de temperatura, luminosidad y humedad capturados por sensores, que pueden ser administrados a través de Internet utilizando servicios Web y el protocolo CoAP. Esta tesis tiene objetivos similares a esta propuesta, se capturan estos mismos parámetros y otros tres más, presencia de un objeto, evento entrada-salida y evento de detección de puerta abierta en una habitación, se recupera esta información con servicios Web y se pueden consultar a través de Internet, con la diferencia que se almacenaran en una ontología.

## **2.3 Proyectos terminales**

- Red de monitoreo remoto inalámbrico de sensores con dispositivos ZIGBEE.[4]

Este proyecto terminal resuelve un problema para monitorear campos de temperatura y humedad, se desarrolla una infraestructura de sensores con dispositivos ZigBee para monitorear y almacenar dicha información, esta red dispositivos se implementa en una LAN. Lo que en esta propuesta de proyecto se plantea, es que teniendo múltiples redes establecidas como la desarrollada en el proyecto tecnológico, se pueda monitorear a través de Internet dichas redes, además de almacenarlas en una ontología.

- Monitoreo Remoto de Temperatura Usando Sensores 1- Wire.[5]

En la implementación del proyecto tecnológico se captura la temperatura (en un intervalo determinado de tiempo) por múltiples sensores ubicados en diferentes áreas geográficas, estos parámetros son visualizados una oficina de monitoreo en un dispositivo display LCD, toda esta comunicación se realiza de forma remota a través de radiofrecuencia. El proyecto mencionado y la presente propuesta comparten un objetivo en común, monitorear un parámetro de temperatura de diversos sensores y visualizar el resultado para un usuario, la diferencia entre ambas propuestas, es que se propone la captura la de información de 5 parámetros más (humedad, luminosidad, presencia de un objeto, evento entrada-salida y evento de detección de puerta abierta en una habitación) en diversos sensores de forma concurrente al igual que su almacenamiento en una ontología, a través de servicios Web, además de visualizar los 6 parámetros de los sensores en una aplicación Web, con el fin de realizar pruebas para su correcto funcionamiento.

- Monitoreo Remoto de Temperatura Y Humedad Utilizando Tecnología ZigBee.[6]

El objetivo del proyecto tecnológico es monitorear los parámetros de humedad y temperatura de diversos sensores, capturar los parámetros de forma inalámbrica, visualizar

sus cambios en un cliente y almacenarlos en una base de datos. El objetivo de esta propuesta es similar, se propone 6 parámetros, temperatura, humedad y luminosidad al igual que el proyecto tecnológico, además de las variables presencia de un objeto, evento entrada-salida y evento de detección de puerta abierta en una habitación, que serán capturados por diferentes sensores ubicados en diferentes cubículos del departamento de Sistemas, capturarlos de forma concurrente y almacenarlos en un ontología, a través de servicios Web.

### **3. JUSTIFICACIÓN**

Teniendo una red de sensores que capturan parámetros de temperatura, humedad, luminosidad, presencia de un objeto, evento entrada-salida y evento de detección de puerta abierta de N salas dentro del departamento de Sistemas, se crea la necesidad de recuperar, procesar, analizar y almacenar la información obtenida por los sensores. Es necesario crear componentes de software con la capacidad de realizar dichas tareas en tiempo real a través de Internet y de forma concurrente, con el fin de que un usuario pueda monitorear los sensores simultáneamente, desde cualquier ubicación y utilizando un navegador Web. Donde el almacenamiento de las variables capturadas se realizará en una ontología; para que en trabajos futuros con los datos almacenados, se pueda extraer conocimiento de los parámetros ambientales, realizar tomas de decisiones con respecto a eventos, con el fin de automatizar los dispositivos (ajustar la calefacción, el aire acondicionado, la iluminación, el sistema de ventilación) de las salas del departamento de sistemas de la UAM AZC.

### **4. OBJETIVOS**

#### **4.1 Objetivo general**

Diseñar e implementar un conjunto de servicios para la obtención concurrente y almacenamiento de datos a partir de sensores de humedad, temperatura, intensidad luminosa, presencia de un objeto, evento entrada y salida y evento de detección de puerta abierta.

#### **4.2 Objetivos específicos**

- Integrar el modelo ontológico existente para el almacenamiento de los datos de sensores, específicamente: datos de humedad, temperatura, luminosidad, presencia de un objeto, evento entrada-salida y evento de detección de puerta abierta.
- Diseñar e implementar servicios para la obtención concurrente de datos de los sensores utilizando un analizador de código HTML.
- Diseñar e implementar servicios para el almacenamiento concurrente de los datos obtenidos a partir de los sensores en el modelo ontológico.

- Implementar una aplicación cliente para invocar y probar el funcionamiento de los servicios.

## **5. MARCO TEÓRICO**

### **5.1 Hardware Arduino**

#### **5.1.1 Sistemas Electrónicos**

Un sistema electrónico es un conjunto de: sensores, circuitería de procesamiento, actuadores y fuente de alimentación [7].

- Sensor

Un sensor es un dispositivo con la capacidad de captar una señal de una forma física y convertirla en una señal eléctrica para ser manipulada por la circuitería, con el fin de comunicarse con el exterior y adquirir un conocimiento de cantidades físicas que por su tamaño o su naturaleza no pueden ser percibidos por los sentidos [8]. (Existen 6 tipos de señales: mecánicas, térmicas, magnéticas, eléctricas, ópticas y moleculares)

El uso de sensores para la captura y procesamiento de información se ha incrementado en los últimos años, siendo parte fundamental dentro de diferentes campos como lo son: entornos industriales, entornos inteligentes, entornos militares, detección ambiental; en específico el uso de arreglos de sensores, donde una red de pequeños dispositivos equipados con sensores que colaboran para realizar una tarea en común. Estas redes están por sistemas electrónicos, componentes de software, múltiples sensores con ciertas capacidades sensitivas además de sistemas de comunicación para coordinar estos sensores.

- Circuitos internos

Los circuitos internos de un sistema electrónico, procesan señales eléctricas, la manipulación de dicha señal dependerá de los diferentes componentes de hardware del sistema, como del conjunto de instrucciones lógicas que el hardware tenga pregrabado y que pueda ejecutar de forma interna [7].

- Actuadores

Los actuadores transforman la señal eléctrica manipulada por los circuitos internos en acciones que actúan directamente sobre el mundo físico.

- Fuente de alimentación

Proporciona la energía necesaria para que el sistema electrónico funcione.

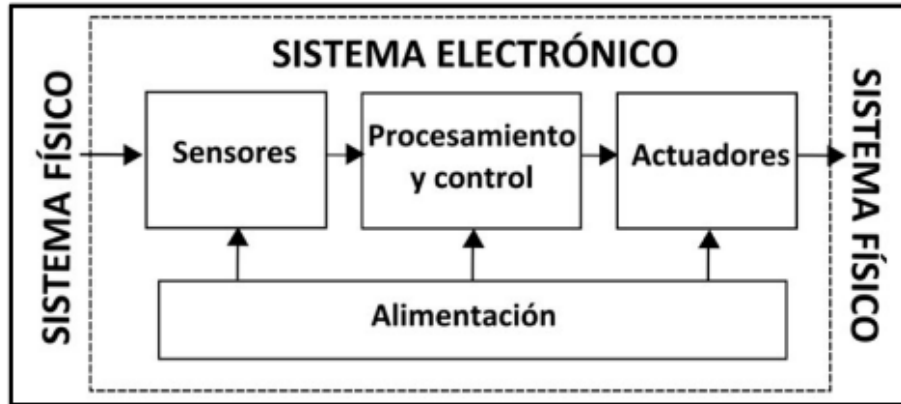


Figura 1 Diagrama de sistema electrónico.

### 5.1.2 Arduino

Arduino es una plataforma electrónica de código abierto basada en hardware y software fáciles de usar [9]. Dicha plataforma está integrada por 3 componentes:

- Placa Arduino.

Es una placa hardware libre que incorpora un microcontrolador re-programable además de una serie de pines tipo hembra y macho que sirven como entradas-salidas de microcontrolador, para conectar de forma sencilla diversos componentes (como los son sensores, actuadores, otras placas arduino, etc.).

Al hablar de placa arduino se refiere a una PCB (placa de circuito impreso, traducción del inglés "printed circuit board"). Las PCB son superficies fabricadas de un metal no conductor sobre las cuales aparecen laminadas pistas de metal conductor. Se utilizan para conectar eléctricamente diversos componentes electrónicos soldados sobre la PCB [7].

Igualmente al referirse a placas arduinos se debe especificar el modelo, porque existe diferentes placas Arduino, en donde varían diferentes especificaciones por ejemplo: el tamaño de la placa, el número de pines, así como el tipo de microcontrolador incorporado.



Figura 2 Comparativa entre placas Arduino.

- IDE Arduino.

Un IDE (Entorno de Desarrollo integrado, en inglés Integrated Development Environment”) es un entorno de programación, en donde se encapsula como un software de aplicación herramientas para el desarrollo, como son: un editor de código fuente, depurador, compilador y una interfaz gráfica. El objetivo es proveer un marco de trabajo que sea amigable con diferentes lenguajes de programación.

El entorno de desarrollo integrado de la plataforma Arduino es libre, gratis y multiplataforma, dicho IDE además del lenguaje de programación Arduino está inspirado en otro IDE y lenguaje de programación de código abierto processing [10], utilizado como medio para la enseñanza y producción de proyectos multimedia e interactivo, que está basado en el lenguaje de programación Java.

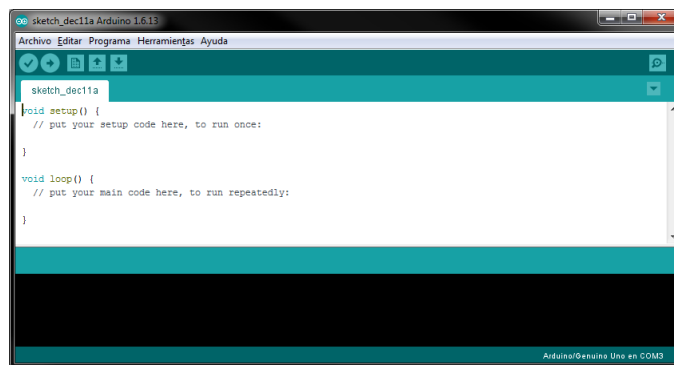


Figura 3 Interfaz del IDE Arduino.

- Lenguaje de programación Arduino.

Un lenguaje de programación consiste en todos los símbolos, caracteres y reglas de uso que permiten a las personas dar instrucciones que son realizadas por una máquina. El lenguaje de programación Arduino es propio de la plataforma. Arduino está inspirado en processing pero el lenguaje de programación utiliza conceptos del paradigma de programación orientada a objetos basado en lenguaje C y C++. El lenguaje de programación Arduino especifica en su sitio web [9] que los programas en Arduino pueden ser divididos en tres componentes principales: estructuras, variables /constantes y funciones.

## 5.2 Modelo ontológico

El ser humano a lo largo de su existencia ha tenido la necesidad de compartir su conocimiento con otras personas. Existen diversos medios de comunicación entre humanos como lo es el lenguaje hablado, escrito así como las artes.

En la era digital, la información juega un papel vital en la búsqueda del conocimiento, para poder compartirlo primero debe ser almacenado para después ser manipulado e interpretado por los humanos, para así alcanzar dicho conocimiento, actualmente las bases de datos son utilizados como almacenes de datos, por definición son "sistema para llevar registros, es decir, es un sistema cuya finalidad general es almacenar información y permitir

a los usuarios recuperar y autorizar peticiones de información.” [11], pero en los últimos años surge la necesidad de compartir conocimiento entre equipos de cómputo, por lo tanto es necesario definir un vocabulario común para compartir dicho conocimiento dentro de un dominio concreto, a través de clases o conceptos, propiedades o atributos de las clases, así como relaciones entre ellas, en conclusión el uso de ontologías.

La principal razón del uso de ontologías se debe en gran medida a que son importantes herramientas para realizar actividades como: procesamiento de lenguaje natural, filtrado y recuperación de información, acceso a datos [12].

El término ontología tiene un origen griego, en donde el concepto fue utilizado para la ciencia de la existencia o del estudio del ser. En el área de la computación, el término de ontología ha recibido diferentes interpretaciones, a continuación se listan las definiciones más relevantes.

- “Una especificación explícita y forma sobre una conceptualización compartida” [13]
- “Taxonomía de conceptos con atributos y relaciones, que proporciona un vocabulario consensuado para definir redes semánticas de unidades de información interrelacionadas” [13]
- “Una ontología es una base de datos donde se describen conceptos del mundo o algún dominio en específico, sus propiedades y como se relaciona los conceptos entre sí” [14]

Continuando con la definición de Thomas Gruber, una conceptualización es una abstracción, una visión simplificada del mundo que se quiere representar, continuando con Gruber una ontología tiene los siguientes componentes para representar el conocimiento de un dominio [15]:

- Conceptos: son las ideas básicas que se intentan formalizar. Los conceptos pueden ser clases de objetos, métodos, planes, estrategias, procesos de razonamiento, etc.
- Relaciones: representan la interacción y enlace entre los conceptos del dominio. Suelen formar la taxonomía del dominio. Por ejemplo: subclase-de, parte-de, parte-exhaustiva-de, conectado-a, etc.
- Funciones: son un tipo concreto de relación donde se identifica un elemento mediante el cálculo de una función que considera varios elementos de la ontología. Por ejemplo, pueden parecer funciones como categorizar-clase, asignar fecha, etc.
- Instancias: se utilizan para representar objetos determinados de un concepto.
- Axiomas: son teoremas que se declaran sobre relaciones que deben cumplir los elementos de la ontología. Por ejemplo: “Si A y B son de la clase C, entonces A no es subclase de B”, “Para todo A que cumpla la condición C1, A es B”.

### 5.3 Sockets

ARPANET fue un proyecto desarrollado por el Departamento de Defensa de los Estados Unidos, es conocido por ser una de las primeras redes de computadora utilizada para comunicar a diferentes instituciones académicas y estatales a los largo de aquel país. Sobre ARPANET y su sucesora la red ARPA Internet, se desarrollaron los protocolos IP, TCP y UDP. Al inicio de la década de los 80, los equipos conetados a estas redes utilizaban el sistemas operativo UNIX, en específico la variante de la universidad de Berkeley (BSD UNIX), donde a partir de la versión BSD 4.2, se incorporó al sistema operativo una serie de llamadas al sistema (llamadas al kernel) con el objetivo de incorporar el diseño de aplicaciones que utilizaran los protocolos de ARPANET. Estas llamadas, junto con algunas bibliotecas estándar de unix, formaron el inicio de la llamada interfaz de programación de sockets.

Un socket es un punto de acceso a los servicios de comunicación entre procesos. Cada socket tiene asociada una dirección que lo identifica. Conociendo esta última, se puede establecer una comunicación con un socket para que actúe como extremo de un canal bidireccional [15].

Los sockets son el método más utilizado para la comunicación entre procesos, los cuales se ejecutan en una misma máquina o maquinas conectadas a través de una red. Dado su origen, los sockets y su programación están fuertemente vinculados con el sistema operativo UNIX, pero, en la actualidad, hay disponibles otros lenguajes de programación que proporcionan llamadas al sistema para trabajar con sockets, por ejemplo: Perl, Java, C++.

#### 5.3.1 Arquitectura Cliente Servidor

La gran mayoría de aplicaciones que usan sockets siguen la arquitectura cliente-servidor. En donde existe un cliente que envía peticiones de servicio a un socket en específico, por otra parte un servidor está en la espera de peticiones y las responde.

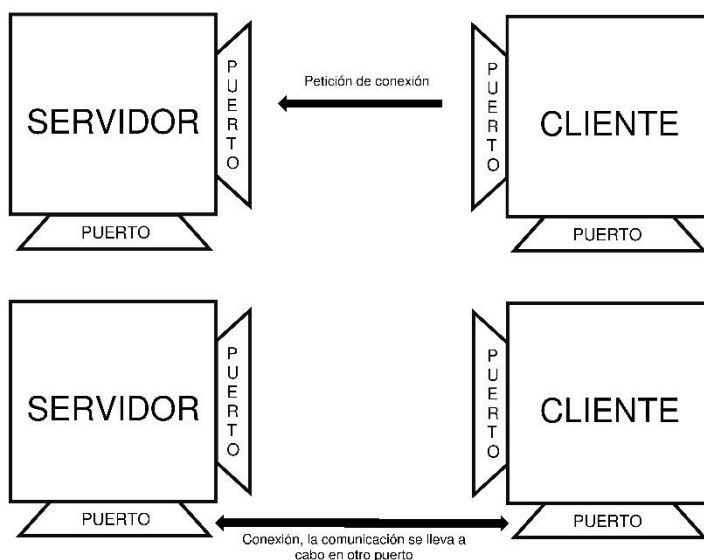


Figura 4 Funcionamiento general de sockets.



### 5.3.2 Socket servidor

Cuando se crea un socket servidor que actúa como destino de la comunicación, es importante asignarle un número de puerto y dirección única, puesto que los sockets clientes la necesitarán para poder establecer la conexión. Cuando se establece la conexión con un cliente el servidor asigna un nuevo puerto para seguir atendiendo al cliente y liberará el puerto establecido para seguir atendiendo a otros clientes. Cuando se asigna un número de puerto a un socket es conveniente considerar que los sistemas operativos tienen definidos rangos de puertos correspondientes a puertos reservados y puertos disponibles, ya que el sistema operativo no permitirá asignar un puerto reservado a un socket.

### 5.3.3 Socket cliente

En la creación de socket cliente para utilizarlo como origen de una comunicación, el número de puerto es irrelevante. Lo más conveniente es dejar que el sistema lo asigne de forma automática, puesto que la dirección del cliente solo es utilizada por el servidor cuando necesita interactuar con el cliente.

## 5.4 Concurrencia

Dentro del área de las ciencias de la computación el concepto de concurrencia se refiere a la propiedad que tienen los sistemas de cómputo donde sus procesos se pueden ejecutar de manera simultánea y pueden interactuar entre ellos además de compartir recursos; en conclusión sistemas multitareas.

Dentro de los lenguajes de programación, la concurrencia está implementada en la programación multihilo, un programa es una secuencia de instrucciones, escritas para realizar un tarea específica en una máquina, un programa multihilo contiene secciones de instrucciones que se pueden ejecutar de forma simultánea. Cada sección es conocida como hilo; los hilos tienen la capacidad de ejecutarse de forma independiente, esto no significa que puedan ejecutarse ellos solos; requieren de la supervisión de un proceso padre (Un proceso es un programa en ejecución). La programación multihilo se puede ver como una forma de multitarea.

Los hilos a menudo son conocidos como procesos ligeros o subprocesos. Un hilo es similar a un proceso con la diferencia de que un hilo se ejecuta dentro del contexto de un programa.

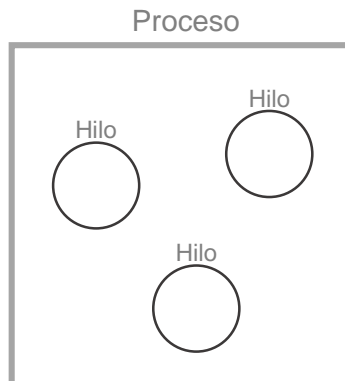


Figura 5 Relación entre procesos e hilos.

## 6. DESARROLLO DEL PROYECTO

El proyecto está desarrollado en diferentes módulos que se muestra en la siguiente ilustración:

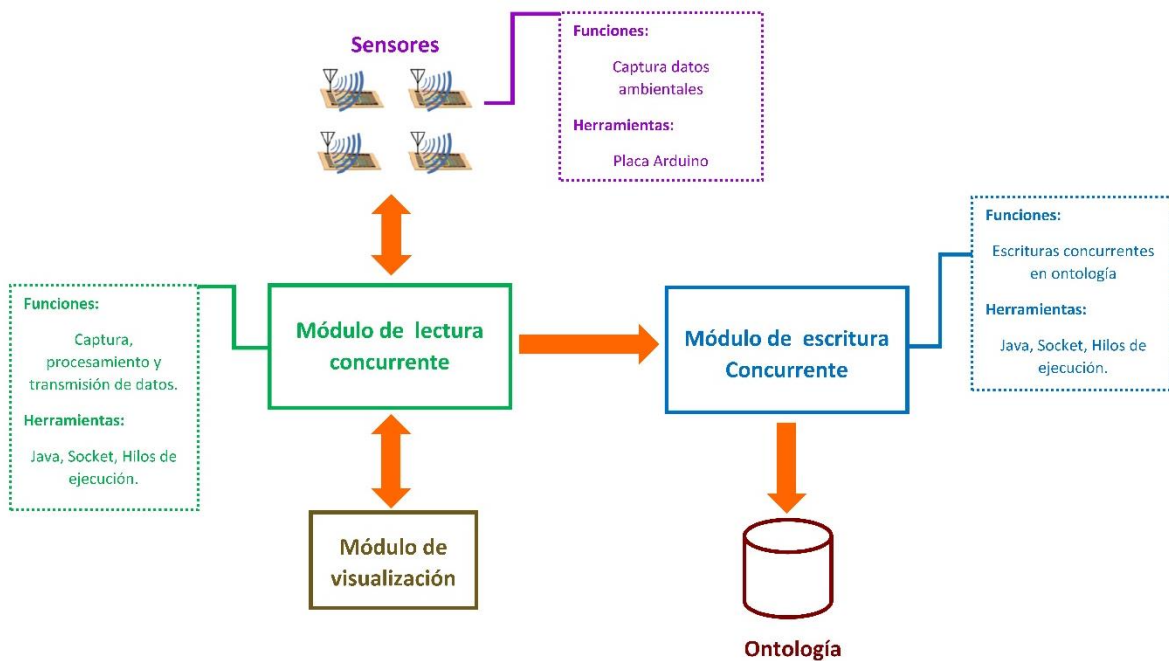


Figura 6 Arquitectuta implementada para Aplicación concurrente para la obtención y almacenamiento de datos a partir de sensores.

### 6.1 Sensores

Antes de iniciar con la explicación de los módulos principales que componen este proyecto de integración, es necesario hablar de los sensores y su relación con la ontología.

#### 6.1.1 Sensor de humedad

Para medir la magnitud física de humedad se utiliza el sensor DHT11 [18], tiene un rango de medición de 20-90% RH (Humedad Relativa), con una resolución de 1% RH y una tolerancia de  $\pm 5\%$  RH.

#### 6.1.2 Sensor de temperatura

Para medir la magnitud física de tempertaura se utiliza el sensor DHT11 [18], tiene un rango de medición de 0-50 °C (Grados centigrados), con una precisión de 1°C y una tolerancia de  $\pm 2$  °C.

#### 6.1.3 Sensor de intensidad luminosa

Para medir la magnitud física de luz se utiliza el sensor IM120710017 [19], tiene un rango de medición de 5 a 10 kohm, con una tolerancia de 0.009 VCC.

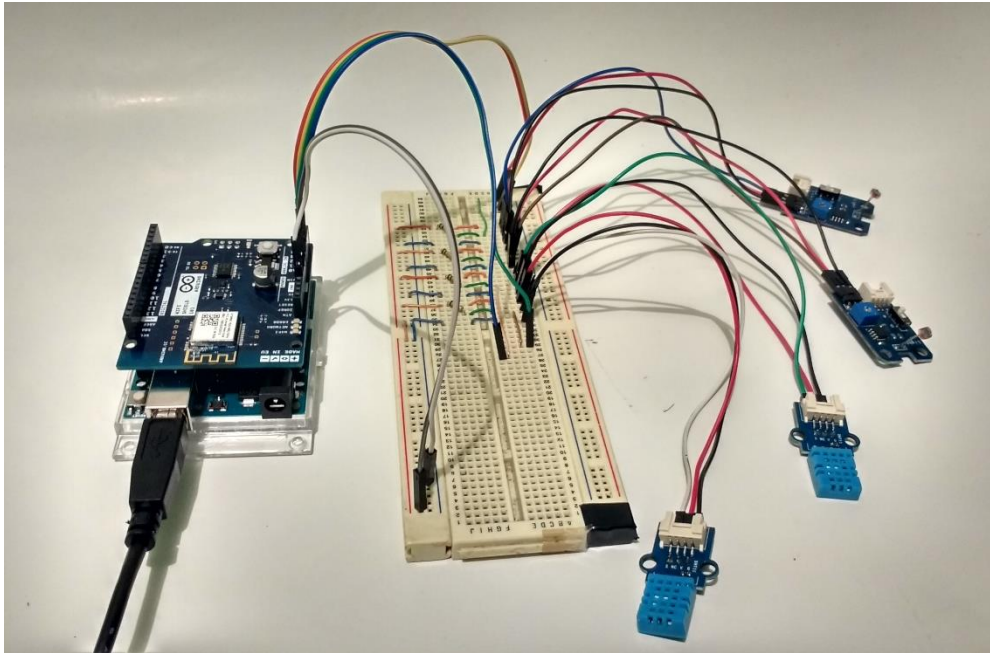


Figura 7 Conexiones entre placa Arduino, sensores y SHIELD WIFI 101.

Los sensores son conectados a una placa Arduino UNO [20] y un SHIELD WIFI 101. A través del SHIELD WIFI 101 [21] se crea un servidor Web en el puerto 80 (se ha seleccionado de forma arbitraria este puerto, porque es el predeterminado en donde los clientes realizan peticiones a un servidor HTTP). Cuando el servidor tiene una petición por parte de un cliente, envía como respuesta una página HTML en donde se incrusta información de los sensores, como es: el tipo de sensor, la magnitud o magnitudes físicas que captura y el valor numérico de estas. Es necesario asociar el SHIELD WIFI 101 a un router inalámbrico para establecer un medio de comunicación con los clientes, además de la asignación de una dirección IP estática o dinámica, con ella se localizará el servidor Web.

El código fuente de un servidor Web Arduino se encuentra en la sección ANEXOS, ANEXO A: Código fuente de un servidor Web Arduino

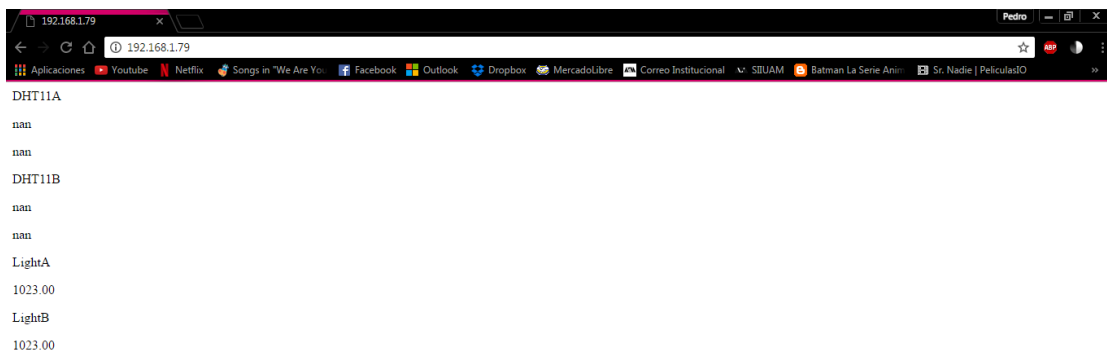


Figura 8 Página Web resultado de la respuesta del servidor Web Arduino a la petición de un cliente.

```

<html><head><meta http-equiv="Content-Type" content="text/html; charset=windows-1252">
</head>
<body>
  <a class="Sensor" id="DHT11A">DHT11A</a>
  <p></p>
  <a name="DHT11A" id="hasTempValue">
    29.00</a>
  <p></p>
  <a name="DHT11A" id="hasHumidityValue">
    37.00</a>
  <p></p>
  <a class="Sensor" id="DHT11B">DHT11B</a>
  <p></p>
  <a name="DHT11B" id="hasTempValue">
    24.00</a>
  <p></p>
  <a name="DHT11B" id="hasHumidityValue">
    30.00
  </a>
  <p></p>
  <a class="Sensor" id="LightA">LightA</a>
  <p></p>
  <a name="LightA" id="hasLightValue">
    124.00
  </a>
  <p></p>
  <a class="Sensor" id="LightB">LightB</a>
  <p></p>
  <a name="LightB" id="hasLightValue">
    130.00
  </a>
  <p></p>
</body>
</html>

```

Figura 9 Ejemplo del código de una página HTML resultado de una petición a un servidor Web Arduino.

## 6.2 Módulo de lectura concurrente

Este módulo realiza las siguientes actividades: la captura, el procesamiento y la transmisión de los datos de provenientes de los sensores. Las tres tareas antes mencionadas son ejecutadas de forma periódica por una tarea programada. Además recibe peticiones por parte del módulo de visualización para realizar la captura de datos.

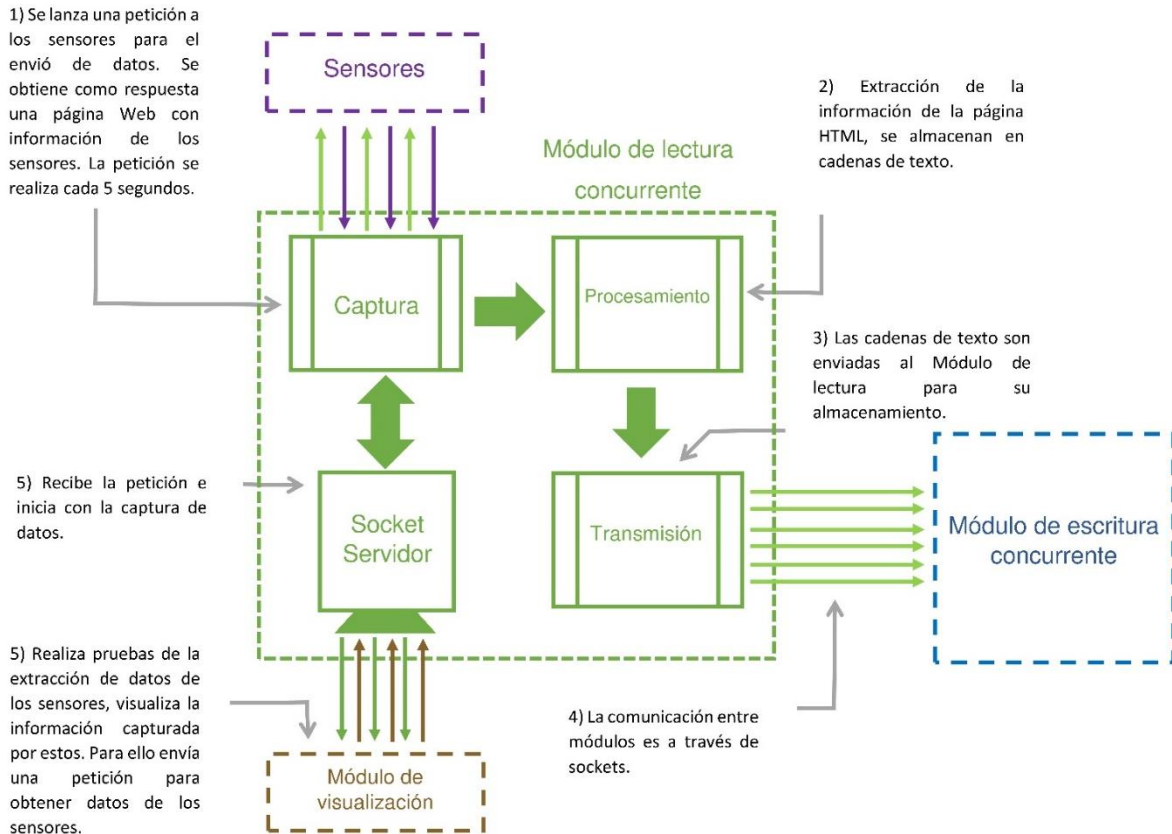


Figura 10 Esquema del funcionamiento del módulo de lectura concurrente.

### 6.2.1 Captura

La captura de datos se realiza de forma periódica en un intervalo de 5 segundos, este valor fue asignado de forma arbitraria, aunque este número puede ser modificado. Para realizar la captura de datos de forma concurrente, el módulo de lectura crea procesos ligeros (hilos), donde cada uno de ellos realiza una petición a un servidor Web. El número de hilos creados depende del contenido de un archivo de configuración llamado "direcciones.txt". El archivo contiene una lista de direcciones IP, donde cada una de ellas corresponde a un servidor Web. La respuesta del servidor a la petición es una página HTML que contiene la información de los sensores, (ver las figuras 8 y 9) después el hilo analiza y procesa el código HTML de la página para extraer información. La implementación en Java de esta sección, se encuentra en: ANEXO B: Módulo de lectura concurrente, Código fuente de captura concurrente de datos de sensores

### 6.2.2 Procesamiento

Antes de explicar la fase de procesamiento es necesario mencionar que el código HTML de la página que contiene la información de los sensores, ver figura 9, contiene información dentro de los atributos de las etiquetas HTML, que está relacionada con la ontología, el objetivo es facilitar el procesamiento de datos, en la tabla 1 se describe la función de cada uno de los posibles valores para los atributos.

Valor de atributos	Tipo de dato en la ontología	Descripción
Sensor	Subclase	Es una subclase de la clase Device, que pertenece a la ontología OntologyDevices.
DHT11A	Individuo	Individuo miembro de la clase sensor, identifica a un sensor DHT11 en la placa arduino.
DHT11B	Individuo	Individuo miembro de la clase sensor, identifica a un sensor DHT11 en la placa arduino.
LightA	Individuo	Individuo miembro de la clase sensor, identifica a un sensor IM120710017 placa arduino.
LightB	Individuo	Individuo miembro de la clase sensor, identifica a un sensor IM120710017 en la placa arduino.
hasTempValue	Propiedad	Relación entre un individuo (sensor DHT11) con un valor numérico de temperatura.
hasHumidityValue	Propiedad	Relación entre un individuo (sensor DHT11) con un valor numérico de humedad.
hasLightValue	Propiedad	Relación entre un individuo (sensor IM120710017) con un numérico valor de luminosidad.

Tabla 1 Relación de etiquetas con ontología.

En esta sección del módulo de lectura concurrente, cada hilo, con la respuesta obtenida extrae información del sensor. Con la ayuda del API JSOUP [16] se realizan consultas sobre etiquetas HTML y sus atributos, a fin de identificar estructuras del tipo mostrado en la figura 10, donde cada estructura corresponde a la información de un sensor. De cada estructura se extraerán los siguientes datos: individuo, además de la relación y el valor por cada magnitud física. Estos datos se concatenan en una cadena de texto, separando cada elemento con el carácter "|". La implementación en Java de esta sección, se encuentra en: ANEXO B: Módulo de lectura concurrente, Código fuente de procesamiento de datos.

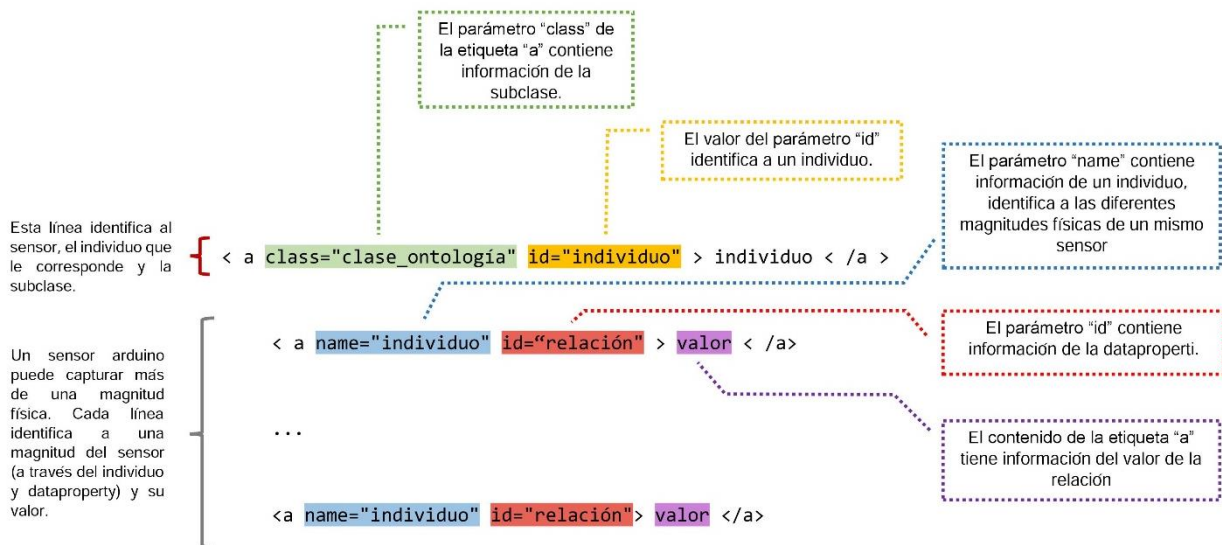


Figura 11 Estructura de la información de un sensor dentro de una página HTML.

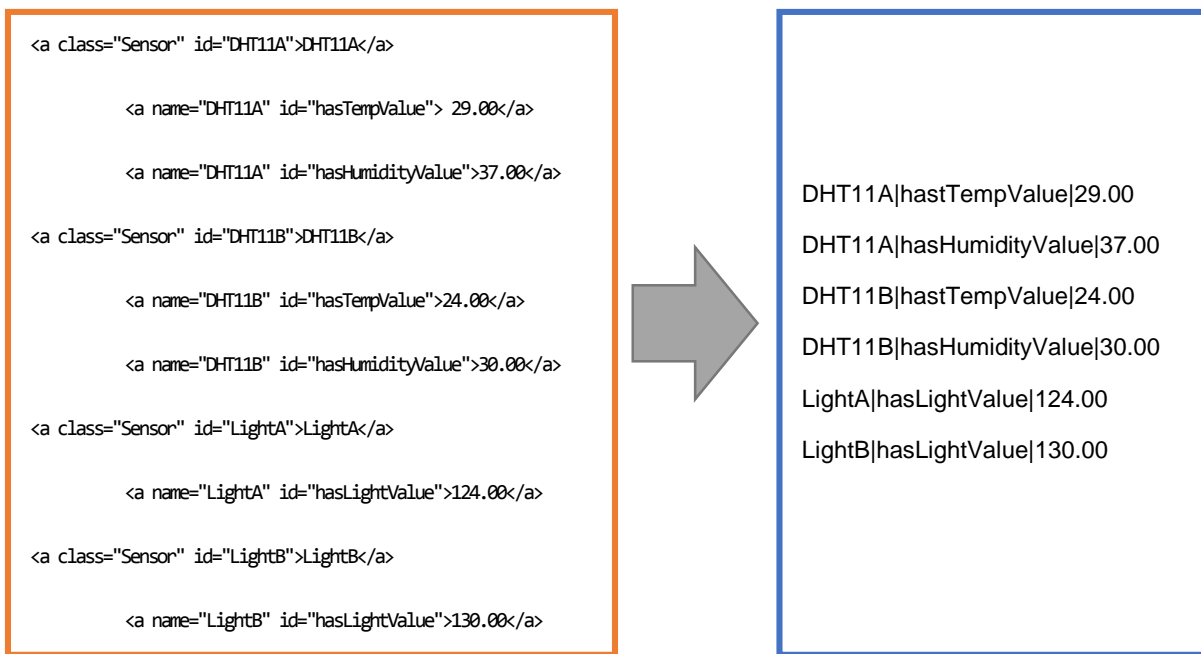


Figura 12 Ejemplo de extracción de datos de una página HTML.

### 6.2.3 Transmisión

Una vez que la sección de Procesamiento ha extraído la información, cada hilo actúa como un socket cliente, que se comunican a un mismo socket servidor que forma parte del módulo de escritura concurrente; en la comunicación cliente–servidor, se envía de forma serializada

la cadena de texto con los datos del sensor o los sensores. Si el socket servidor no está disponible, se captura la excepción "java.net.connectexception connection refused". La implementación en Java de esta sección, se encuentra en: ANEXO B: Módulo de lectura concurrente, Código fuente de hilo para captura de datos de sensor.

#### 6.2.4 Módulo de visualización

El módulo de visualización tiene la función de desplegar la información de los sensores que están en algún servidor Web. Al contrario de la tarea programada, el módulo de visualización es ejecutado por un usuario, para su esto es necesario saber la dirección IP de los sensores que se quieren visualizar.

Cuando el módulo de visualización es ejecutado, a través de un socket cliente envía una petición a un servidor. Cuando el servidor recibe y acepta la petición; el cliente envía la dirección IP que el usuario ha ingresado, se crea un hilo que inicia captura de datos en la IP, en consecuencia ejecutan las secciones de procesamiento y trasmisión. Cuando el hilo ha completado las tres secciones envía sus resultados al módulo de visualización, hasta este punto se termina la comunicación del cliente y servidor.

El servidor a donde se realizan estas peticiones está implementado dentro del módulo de lectura concurrente, además puede atender hasta 100 clientes de forma simultánea.

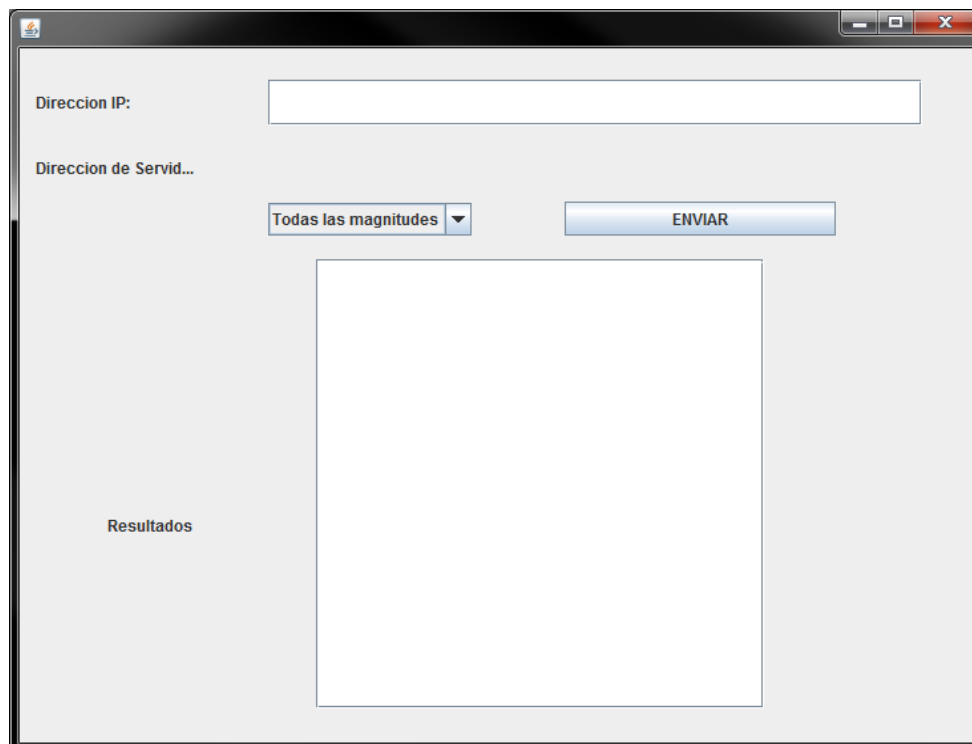


Figura 13 Interfaz del módulo de visualización.



La implementación en Java de esta sección, se encuentra en: ANEXO C: Código fuente de módulo de visualización.

### 6.3 Módulo de escritura concurrente

Esté módulo realiza la escritura de los datos de los sensores en la ontología OntologyDevices.

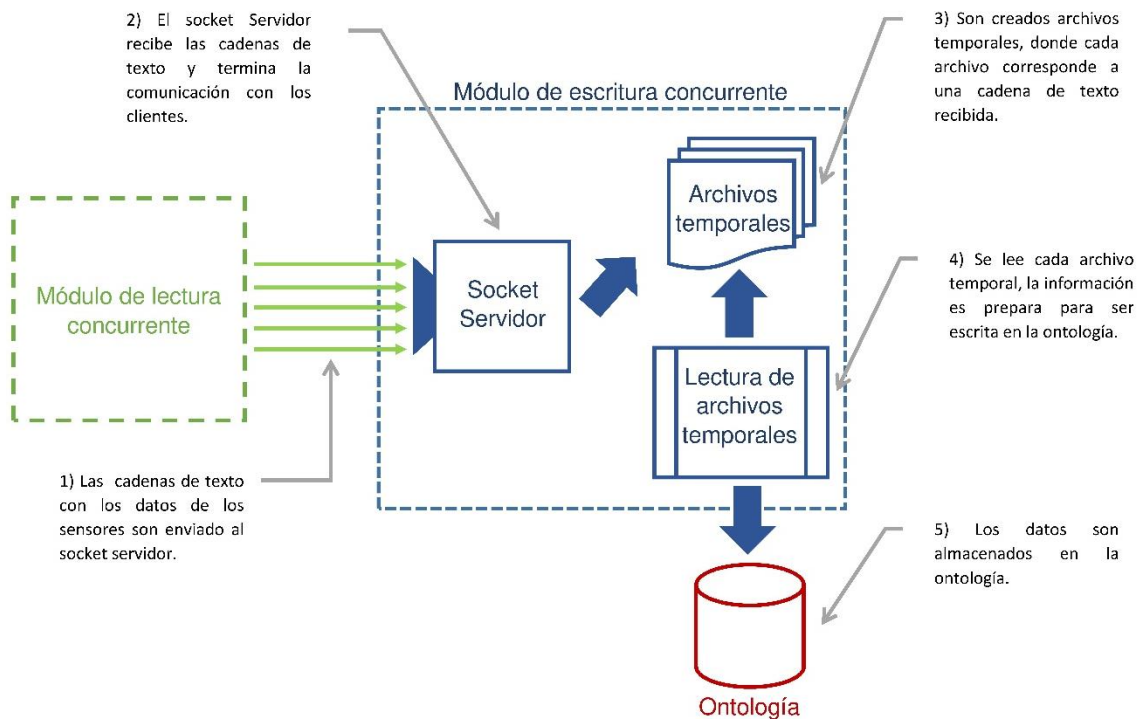


Figura 14 Esquema del funcionamiento del módulo de escritura concurrente.

#### 6.3.1 Servidor concurrente

Este módulo implementa un socket en configuración servidor, él está en la escucha de peticiones por clientes generados en el módulo de lectura concurrente. El servidor es concurrente por lo que puede atender hasta 100 clientes de forma simultánea, para esto utiliza hilos. Cuando la petición del cliente es aceptada por el servidor se inicia la comunicación, donde el cliente envía una cadena de texto con la información del sensor, el servidor recibe la cadena, genera un hilo de ejecución y termina la comunicación con el cliente. El hilo con la cadena de texto crea un archivo temporal y lo almacena en una carpeta llamada archivos\_temporales. La implementación en Java de esta sección, se encuentra en: ANEXO D: Módulo de escritura concurrente, Código fuente de socket servidor de escritura concurrente.

### 6.3.2 Archivos temporales

Debido a que el API OWL [17] para la administración de la ontología desde el lenguaje de programación Java no posee de mecanismo para la inserción de datos de forma concurrente. El objetivo es almacenar de forma temporal la información de los sensores en archivos y después extraer la información de ellos en forma secuencial para almacenarla en la ontología.

En la carpeta archivos\_temporales, el servidor a través de hilos, almacena todos los archivos temporales que contienen las cadenas de texto enviadas por los clientes. Todos los archivos temporales son nombrados con el prefijo “temporal” concatenado con una cadena de números (La cadena de números es asignada por la máquina virtual de Java). La implemetación en Java de esta sección, se encuentra en: ANEXO D: Módulo de escritura concurrente, Código fuente de hilo para creación de archivos temporales.

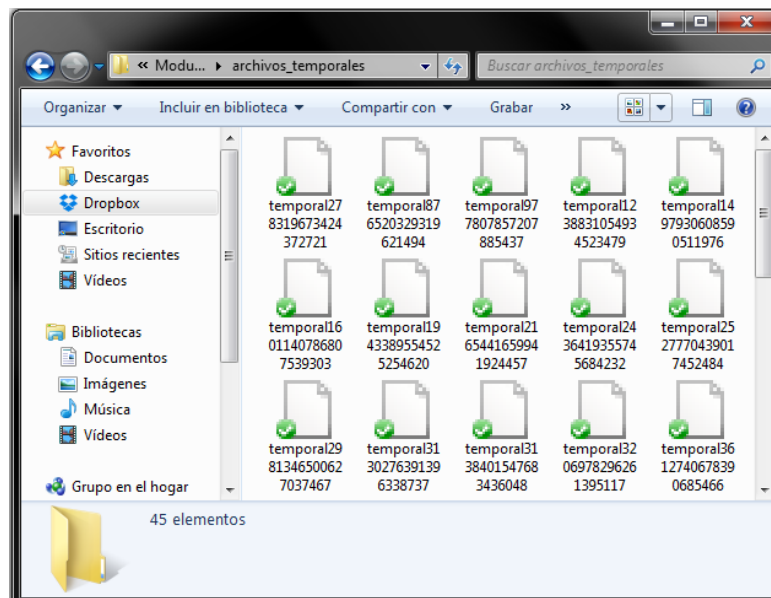


Figura 15 Carpeta archivos\_temporales.

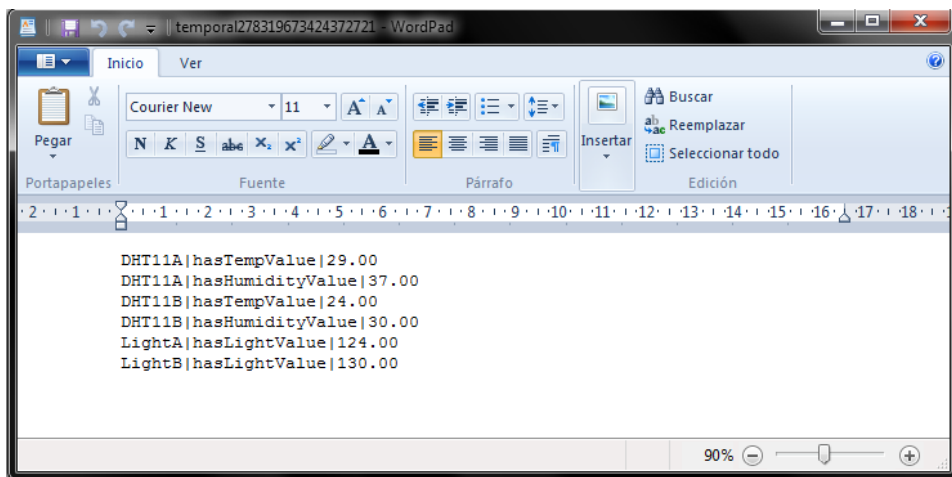


Figura 16 Contenido del archivo temporal “temporal278319673424372721”.

### 6.3.3 Lectura de archivos temporales

Para extraer la información de los archivos temporales, se utiliza una tarea programada para ejecutarse de forma periódica en un intervalo de 2 segundos.

Esta ejecución programada realiza las siguientes operaciones: primero revisa la cantidad de elementos dentro de la carpeta “archivos\_temporales”; si es igual a cero, terminar la ejecución; de lo contrario lee el contenido de cada archivo temporal línea por línea, para extraer los siguientes elementos:

- al individuo
- su relación o *dataproperty*
- y el valor numérico de la relación.

Se separa en tokens cada línea extraída del archivo, el carácter “|” dentro de la cadena funciona como delimitador entre los elementos. Después de esto, se tienen los datos necesarios y se insertan en la ontología. El poblado de la ontología de la ontología se realiza de forma secuencial. Cuando el archivo temporal es leído en su totalidad es destruido, con esto se evita que se sobrescriban datos en la ontología.

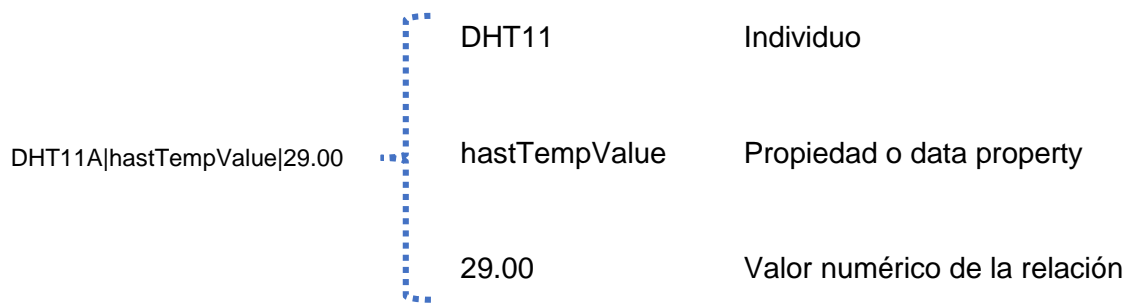


Figura 17 Ejemplo de extracción de los elementos.

La implementación en Java de esta sección, se encuentra en: ANEXO D: Módulo de escritura concurrente, Código fuente de lectura secuencial de archivos temporales.

### 6.3.4 Inserción en el modelo ontológico

La ontología “OntologyDevices.owl” fue utilizada para el almacenamiento de datos de sensores. El modelo ontológico no se elaboró en el desarrollo de este proyecto ya se había implementado con anterioridad. El API OWL [17] para el lenguaje de programación Java proporciona los métodos para escribir en la ontología, pero, para realizar inserciones, es necesario conocer a los individuos con sus propiedades y su valor equivalente en la ontología, por lo que es necesario conocer la taxonomía de OntologyDevices. La ilustración muestra la taxonomía, la cual consiste de la siguiente jerarquía: la clase Device. Dentro de Device se encuentran las subclases SmartPhone, Actuator, Sensor, RFIDCard.

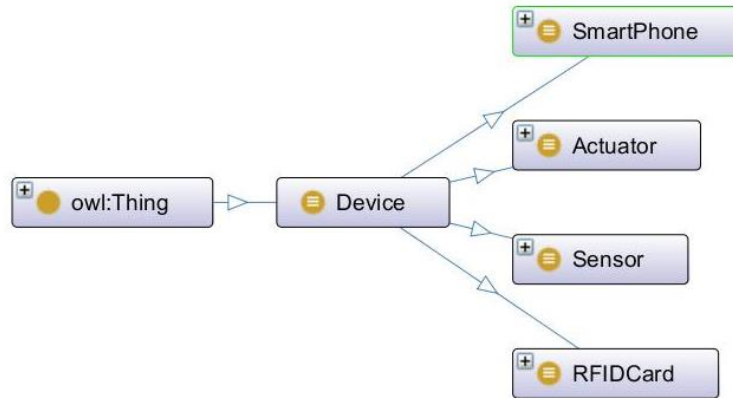


Figura 18 Taxonomía de Device.

En la subclase Sensor que pertenece a la clase Device, es donde se clasifican todos los individuos relacionados con los sensores, los individuos de la subclase son:

Individuos	Propiedades
DHT11A/DHT11B	hasNumber hasModel hasHumidityValue hasTempValue hasNameDevice hasBrand hasKind
LightA/LightB	hasNumber hasModel hasHumidityValue hasTempValue hasNameDevice hasBrand hasKind
RFIDReader	hasNumber hasModel hasHumidityValue hasTempValue hasNameDevice hasBrand hasKind idNumberRead idCardRead

Tabla 2 Individuos y propiedades de la subclase sensor.

Los individuos y propiedades utilizados en la inserción de datos a la ontología son: DHT11A y DHT11B utilizan las propiedades *hasTempValue* y *hasHumidityValue* para los valores de humedad y temperatura, LightA, y LightB utilizan la propiedad *hasLightValue* para el valor de intensidad luminosa.

La implementación en Java de esta sección, se encuentra en: ANEXO D: Módulo de escritura concurrente, Código fuente de inserción en ontología, Código fuente de métodos para administrar ontologías con Java.

## 7. RESULTADOS

### 7.1 Resultado prueba 1

En esta prueba la dirección IP del servidor Web Arduino es: 192.168.1.67, el módulo de lectura y el módulo de escritura se ejecutaron en la misma computadora, el archivo la lectura de sensores fue hecha por el módulo de visualización. El tiempo de ejecución fue de aproximadamente 5 segundos, comienza cuando se envía la petición del servidor y termina cuando se insertan los datos en la ontología. Los datos capturados no sufrieron alteraciones al ser procesados por ambos módulos.

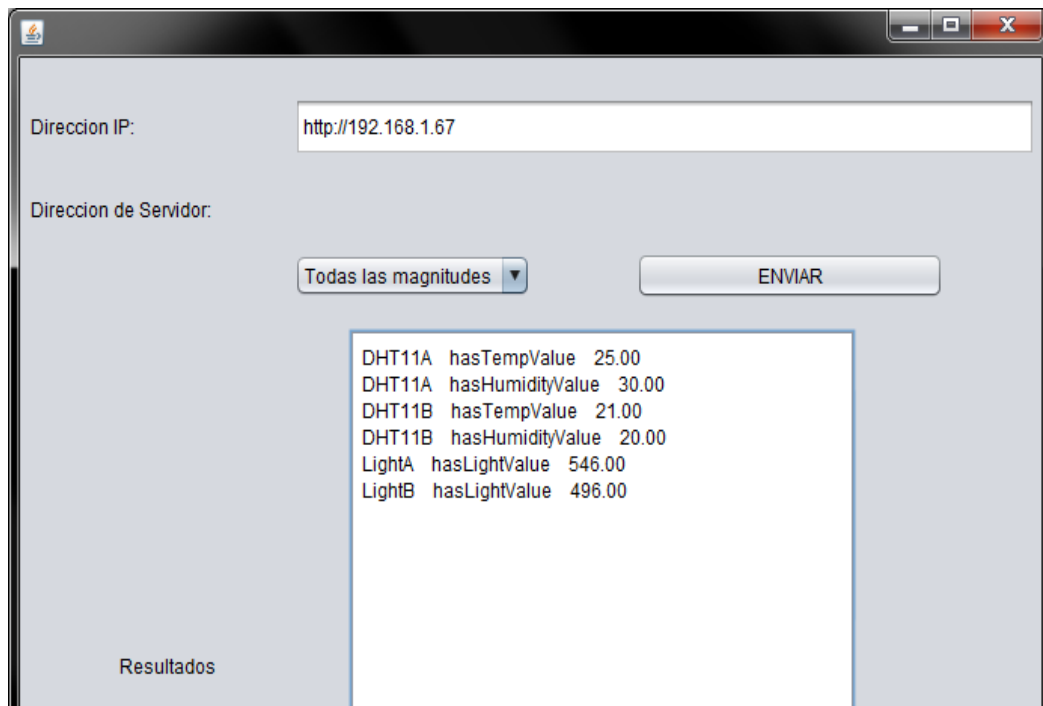


Figura 19 Modulo de visualización, prueba de servidor Web Arduino.

```

<DataProperty IRI="#hasTempValue"/>
  <NamedIndividual IRI="#DHT11A"/>
    <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#float">25.0</Literal>

<DataProperty IRI="#hasHumidityValue"/>
  <NamedIndividual IRI="#DHT11A"/>
    <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#float">30.0</Literal>

```

Figura 20 Inserción de datos del individuo DHT11A en la ontología.

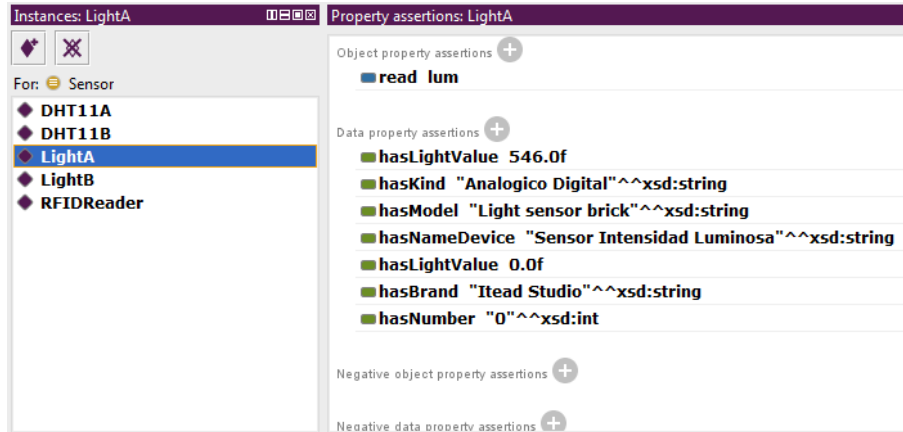
```

<DataProperty IRI="#hasTempValue"/>
  <NamedIndividual IRI="#DHT11B"/>
    <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#float">21.0</Literal>

<DataProperty IRI="#hasHumidityValue"/>
  <NamedIndividual IRI="#DHT11B"/>
    <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#float">20.0</Literal>

```

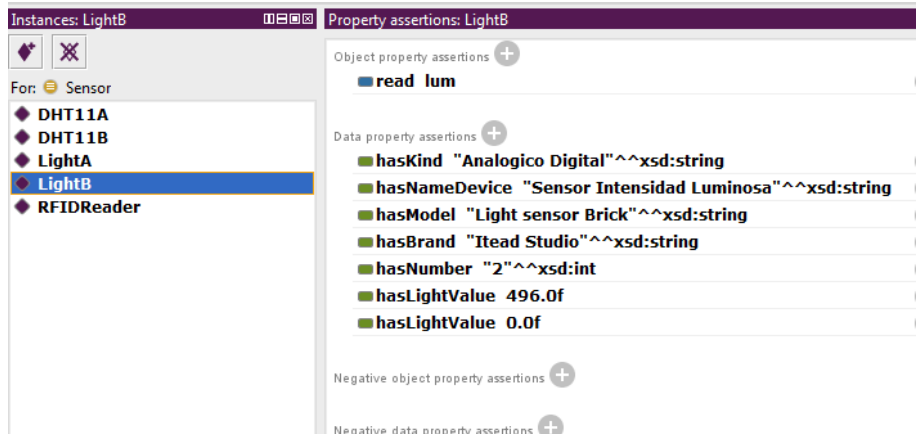
Figura 21 Inserción de datos del individuo DHT11B en la ontología.



```

<DataProperty IRI="#hasLightValue"/>
  <NamedIndividual IRI="#LightA"/>
    <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#float">546.0</Literal>
  
```

Figura 22 Inserción de datos del individuo LightA en la ontología.



```

<DataProperty IRI="#hasLightValue"/>
  <NamedIndividual IRI="#LightB"/>
    <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#float">496.0</Literal>
  
```

Figura 23 Inserción de datos del individuo DHT11A en la ontología.

### 7.2 Resultado prueba 2

En esta prueba se utilizó un servidor Web Apache, el servidor esta implementado en una computadora que tiene la ip: 192.168.1.100, en el servidor se encuentra un página HTML estática simulando la respuesta que genera un servidor Web Arduino. El tiempo de ejecución fue de aproximadamente 2.2 segundos, comienza cuando se envía la petición del

servidor y termina cuando se insertan los datos en la ontología. Los datos capturados no sufrieron alteraciones al ser procesados por ambos módulos.

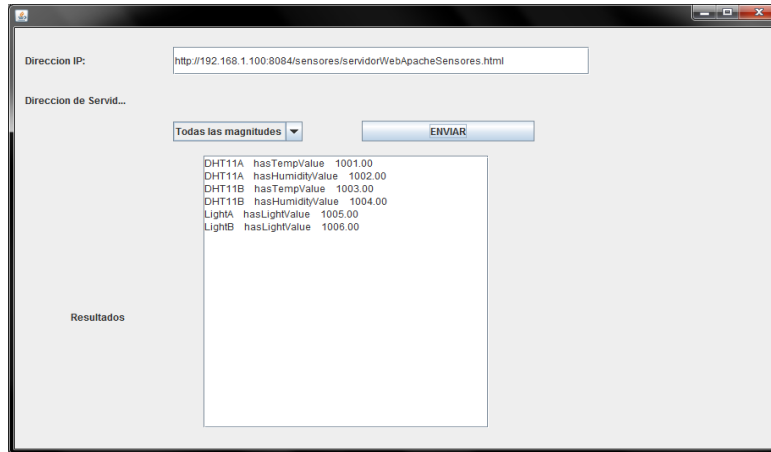
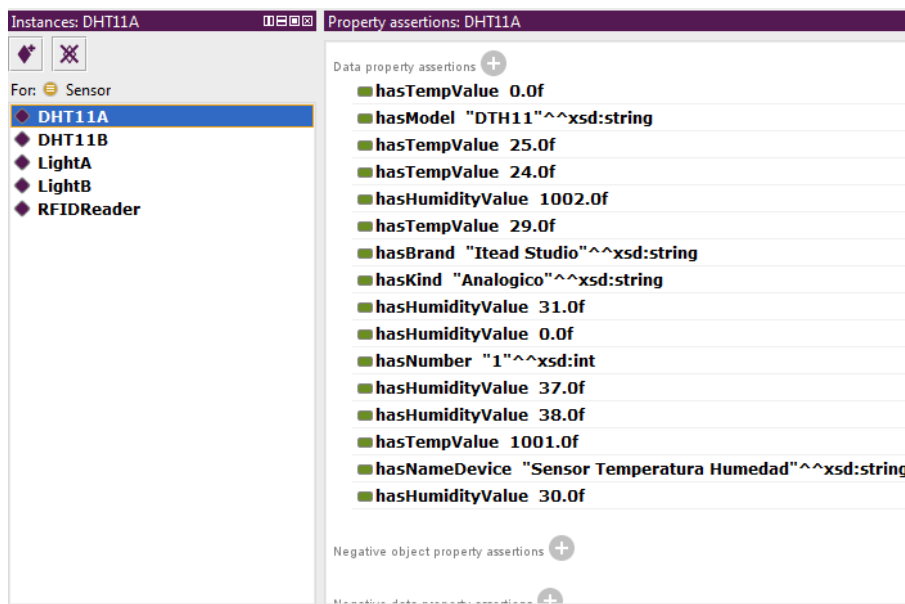


Figura 24 Módulo de visualización, prueba de servidor Web Apache.



```

<DataProperty IRI="#hasTempValue"/>
<NamedIndividual IRI="#DHT11A"/>
<Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#float">1001.0</Literal>

<DataProperty IRI="#hasHumidityValue"/>
  <NamedIndividual IRI="#DHT11A"/>
  <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#float">1002.0</Literal>

```

Figura 25 Inserción de datos del individuo DHT11A en la ontología.



```

<DataProperty IRI="#hasTempValue"/>
  <NamedIndividual IRI="#DHT11B"/>
  <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#float">1003.0</Literal>

<DataProperty IRI="#hasHumidityValue"/>
  <NamedIndividual IRI="#DHT11B"/>
  <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#float">1004.0</Literal>

```

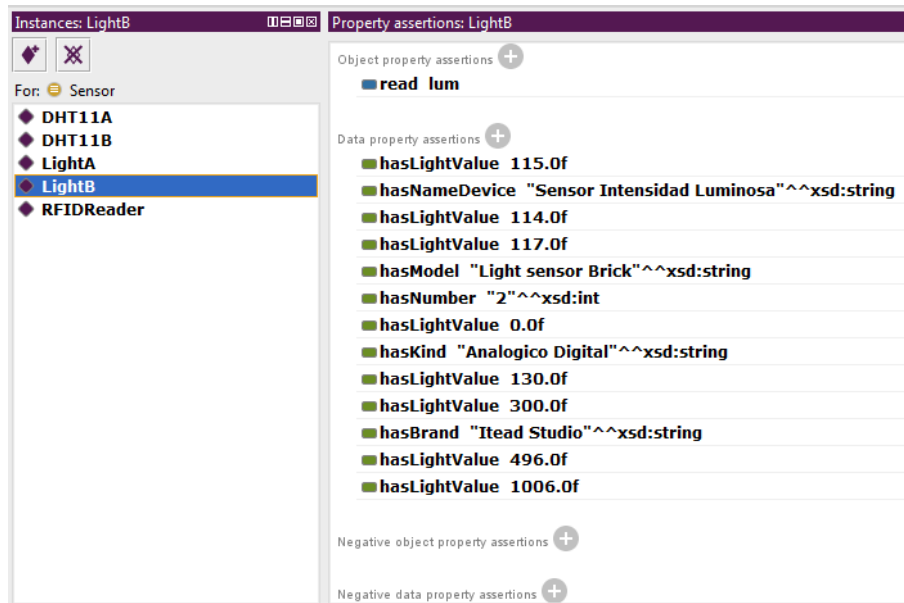
Figura 26 Inserción de datos del individuo DHT11B en la ontología.

```

<DataProperty IRI="#hasLightValue"/>
  <NamedIndividual IRI="#LightA"/>
  <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#float">1005.0</Literal>

```

Figura 27 Inserción de datos del individuo LightA en la ontología.



```

<DataProperty IRI="#hasLightValue"/>
  <NamedIndividual IRI="#LightB"/>
    <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#float">1006.0</Literal>
  
```

Figura 28 Inserción de datos del individuo LightB en la ontología.

### 7.3 Resultados prueba 3

En esta prueba las direcciones de los servidores Web están almacenados en el archivo de configuración "direcciones.txt". El tiempo de ejecución fue de aproximadamente 10 segundos, comienza cuando se envían las peticiones a los sensores y termina cuando se insertan los datos en la ontología. Los datos capturados no sufrieron alteraciones al ser procesados por ambos módulos.

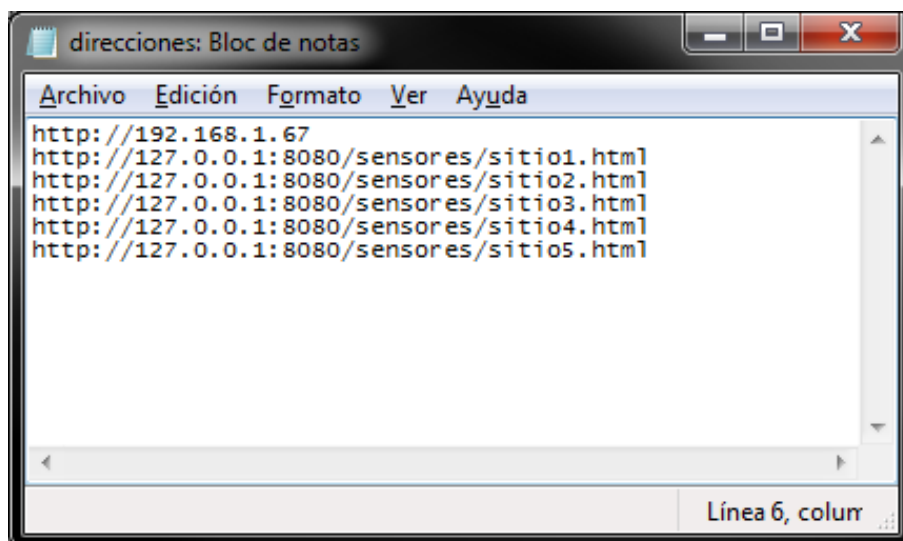


Figura 29 Archivo de configuración direcciones.txt

El módulo de lecturas concurrentes capturo de los sensores los datos que se muestran en la figura 29, al comparar con las ilustraciones de resultados de los individuos, se observa que se hicieron menos inserciones con respecto a número de datos capturados, el motivo es que existen datos de sensores repetidos, la ontología no duplica las data property que tienen el mismo valor.

```
run:
SOCKET SERVIDOR LECTURA : 3445

Extrayendo informacion de: http://192.168.1.67
Extrayendo informacion de: http://127.0.0.1:8080/sensores/sitio1.html
Extrayendo informacion de: http://127.0.0.1:8080/sensores/sitio2.html
Extrayendo informacion de: http://127.0.0.1:8080/sensores/sitio3.html
Extrayendo informacion de: http://127.0.0.1:8080/sensores/sitio4.html
Extrayendo informacion de: http://127.0.0.1:8080/sensores/sitio5.html
DHT11A|hasTempValue|29.00
DHT11A|hasHumidityValue|37.00
DHT11B|hasTempValue|24.00
DHT11B|hasHumidityValue|30.00
LightA|hasLightValue|124.00
LightB|hasLightValue|130.00

DHT11A|hasTempValue|29.00
DHT11A|hasHumidityValue|38.00
DHT11B|hasTempValue|24.00
DHT11B|hasHumidityValue|30.00
LightA|hasLightValue|125.00
LightB|hasLightValue|117.00

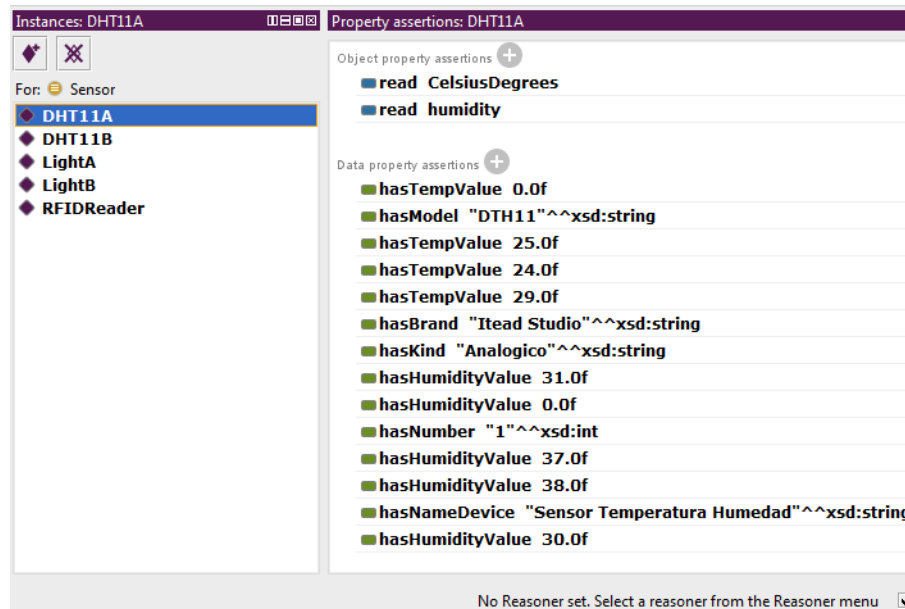
DHT11A|hasTempValue|29.00
DHT11A|hasHumidityValue|37.00
DHT11B|hasTempValue|24.00
DHT11B|hasHumidityValue|30.00
LightA|hasLightValue|124.00
LightB|hasLightValue|130.00

DHT11A|hasTempValue|29.00
DHT11A|hasHumidityValue|37.00
DHT11B|hasTempValue|24.00
DHT11B|hasHumidityValue|30.00
LightA|hasLightValue|125.00
LightB|hasLightValue|115.00

DHT11A|hasTempValue|29.00
DHT11A|hasHumidityValue|37.00
DHT11B|hasTempValue|24.00
DHT11B|hasHumidityValue|30.00
LightA|hasLightValue|124.00
LightB|hasLightValue|114.00

DHT11A|hasTempValue|24.00
DHT11A|hasHumidityValue|31.00
DHT11B|hasTempValue|21.00
DHT11B|hasHumidityValue|20.00
LightA|hasLightValue|288.00
LightB|hasLightValue|300.00
```

Figura 30 Datos capturados de diferentes servidores Web por el módulo de lecturas concurrente.



```

<DataProperty IRI="#hasTempValue"/>
  <NamedIndividual IRI="#DHT11A"/>
    <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#float">25.0</Literal>

<DataProperty IRI="#hasTempValue"/>
  <NamedIndividual IRI="#DHT11A"/>
    <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#float">24.0</Literal>

<DataProperty IRI="#hasTempValue"/>
  <NamedIndividual IRI="#DHT11A"/>
    <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#float">29.0</Literal>

<DataProperty IRI="#hasHumidityValue"/>
  <NamedIndividual IRI="#DHT11A"/>
    <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#float">31.0</Literal>

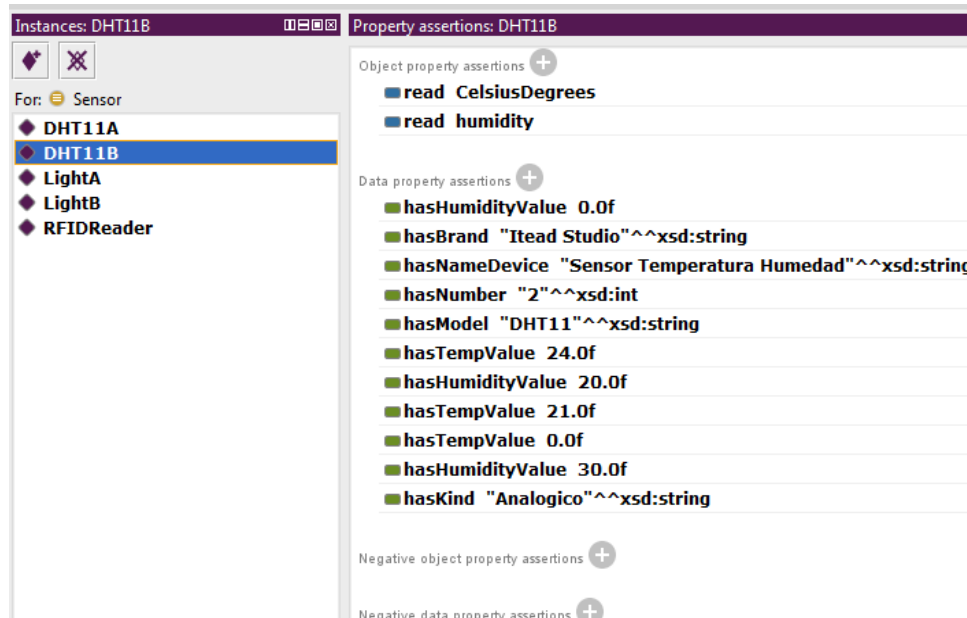
<DataProperty IRI="#hasHumidityValue"/>
  <NamedIndividual IRI="#DHT11A"/>
    <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#float">38.0</Literal>

<DataProperty IRI="#hasHumidityValue"/>
  <NamedIndividual IRI="#DHT11A"/>
    <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#float">30.0</Literal>

<DataProperty IRI="#hasHumidityValue"/>
  <NamedIndividual IRI="#DHT11A"/>
    <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#float">37.0</Literal>

```

Figura 31 Inserción de datos de diferentes servidores Web al individuo DHT11A en la ontología.



```

<DataProperty IRI="#hasTempValue"/>
  <NamedIndividual IRI="#DHT11B"/>
    <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#float">21.0</Literal>

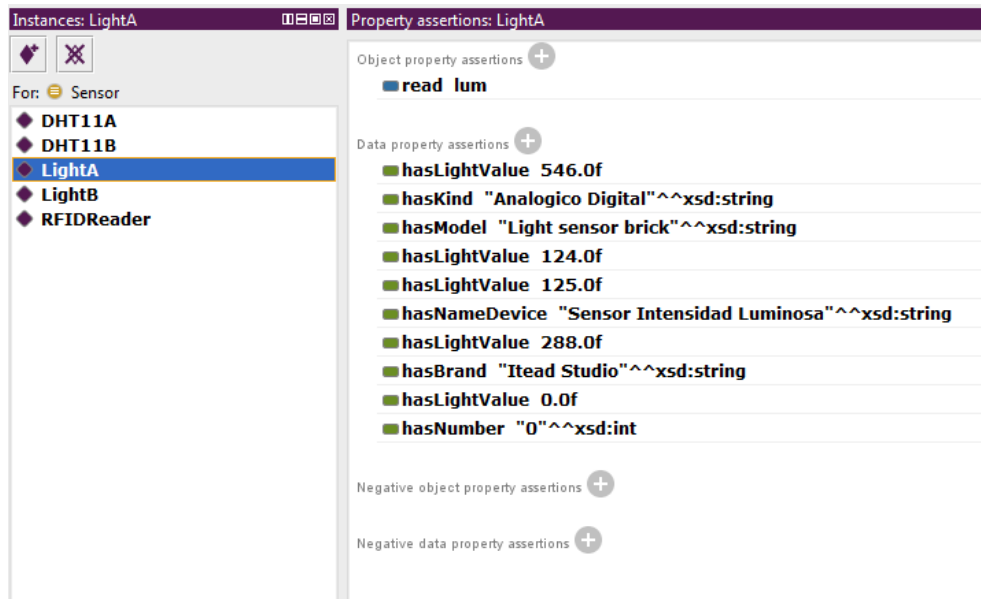
<DataProperty IRI="#hasTempValue"/>
  <NamedIndividual IRI="#DHT11B"/>
    <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#float">24.0</Literal>

<DataProperty IRI="#hasHumidityValue"/>
  <NamedIndividual IRI="#DHT11B"/>
    <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#float">30.0</Literal>

<DataProperty IRI="#hasHumidityValue"/>
  <NamedIndividual IRI="#DHT11B"/>
    <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#float">20.0</Literal>

```

Figura 32 Inserción de datos diferentes servidores Web al individuo DHT11B en la ontología.



```

<DataProperty IRI="#hasLightValue"/>
  <NamedIndividual IRI="#LightA"/>
    <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#float">125.0</Literal>

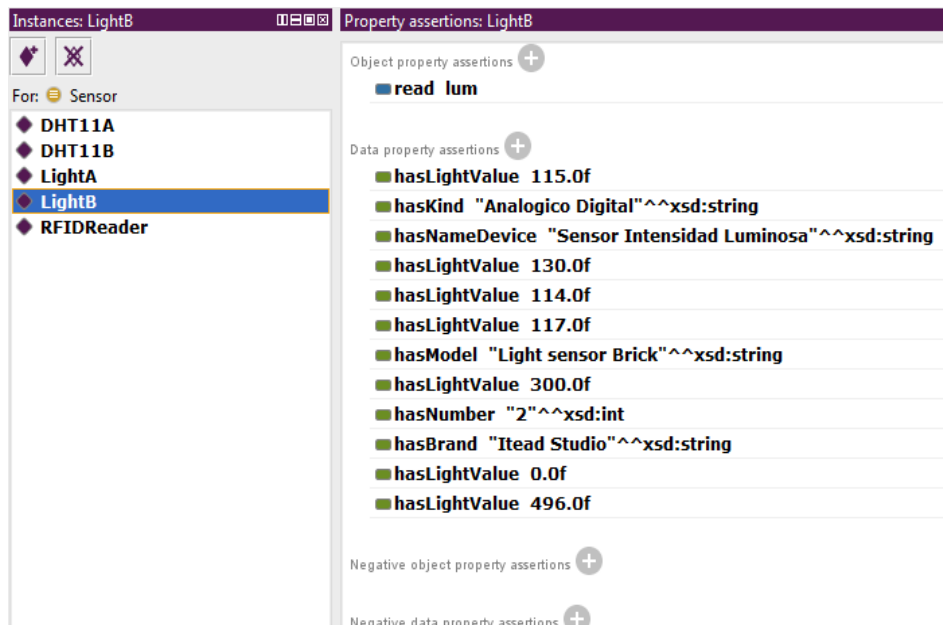
<DataProperty IRI="#hasLightValue"/>
  <NamedIndividual IRI="#LightA"/>
    <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#float">546.0</Literal>

<DataProperty IRI="#hasLightValue"/>
  <NamedIndividual IRI="#LightA"/>
    <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#float">124.0</Literal>

<DataProperty IRI="#hasLightValue"/>
  <NamedIndividual IRI="#LightA"/>
    <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#float">288.0</Literal>

```

Figura 33 Inserción de datos diferentes servidores Web al individuo Light A en la ontología



```

<DataProperty IRI="#hasLightValue"/>
  <NamedIndividual IRI="#LightB"/>
    <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#float">115.0</Literal>

<DataProperty IRI="#hasLightValue"/>
  <NamedIndividual IRI="#LightB"/>
    <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#float">130.0</Literal>

<DataProperty IRI="#hasLightValue"/>
  <NamedIndividual IRI="#LightB"/>
    <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#float">114.0</Literal>

<DataProperty IRI="#hasLightValue"/>
  <NamedIndividual IRI="#LightB"/>
    <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#float">117.0</Literal>

  <DataProperty IRI="#hasLightValue"/><NamedIndividual IRI="#LightB"/><Literal
  datatypeIRI="http://www.w3.org/2001/XMLSchema#float">496.0</Literal>

```

Figura 34 Inserción de datos diferentes servidores Web al individuo Light B en la ontología.

## 8. ANÁLISIS Y DISCUSIÓN DE RESULTADOS

Las pruebas realizadas en los módulos tienen el objetivo de comprobar que las operaciones de captura y almacenamiento de los datos de los sensores se realice correctamente; para ellos se establecieron los siguientes criterios:

Consistencia de los datos: Asegurar que la información requerida esté completa, que los datos se mantiene idénticos durante las operaciones realizadas en los módulos.

- Tiempo de operación: Comparar el tiempo transcurrido entre la captura y su almacenamiento de datos en diferentes escenarios.

Prueba	Tiempo	Persistencia
1	5 segundos aproximadamente	Ningún dato fue alterado
2	2.2 segundos aproximadamente	Ningún dato fue alterado
3	10 segundos aproximadamente	Ningún dato fue alterado

Tabla 3 Comparación entre resultados.

Los módulos de captura y lectura concurrentes garantizan la persistencia de datos, los resultados de las pruebas realizadas capturaron y almacenaron en forma correcta los datos de los sensores, no hubo pérdida ni alteración de datos, la inserción de individuos con su propiedad y valor en la ontología se realizó con éxito.

En las pruebas 1 y 2 el número y tipos de sensores son idénticos, el tiempo de la prueba 1 es casi el doble de la prueba 2, esto es debido al servidor Web a donde realizaron las peticiones, el servidor Web Apache responde aproximadamente en la mitad de tiempo con respecto al servidor Web Arduino, motivo: el hardware; Tanto la placa Arduino UNO y el SHIELD WIFI 101 son del tipo básico, sus recursos son muy limitados además la comunicación con la placa Arduino servidor es de forma inalámbrica. El tiempo de ejecución de la prueba 2 es menor, aunque es conocido que este tipo de comunicación puede llegar a presentar retardos debido a su naturaleza.

La prueba 3 es la que más tiempo consume en su ejecución, las peticiones se realizan a 5 servidores Web Apache y a un Arduino; la lectura y la escritura son concurrentes, por esta razón en teoría la ejecución debería tardar aproximadamente 5 segundos, que es el tiempo más lento de lectura y escritura de los datos de un servidor Web Arduino. La prueba 3 tardo aproximadamente 10 segundos, esto es debido a que el módulo de escritura concurrente atiende a todos cliente del módulo de escritura concurrente simultáneamente pero, la escritura sobre la ontología es secuencial, es ahí donde el módulo demora más tiempo.



## 9. CONCLUSIONES

La lectura de datos de forma concurrente se ha cumplido exitosamente, los datos desde su lectura hasta su escritura en la ontología son persistentes, por esta razón el poblado del modelo ontológico se ha realizado con éxito, a los sensores representados como individuos en la ontología se les asigno un valor capturado por un sensor asociado a un data property; sin corromper el modelo.

Los tiempos de ejecución son aceptables pero pueden mejorar. Como se muestra en la sección de resultados, un servidor Web Arduino tarda un tiempo considerable en responder una petición de un cliente, la razón son los recursos limitados de la placa Arduino UNO por otra parte la lectura secuencial de archivos temporales influye en el tiempo de ejecución, si el número de servidores Web para la captura de datos incrementa, la cantidad de archivos temporales generados aumentara de igual manera, lo que incrementa el tiempo de ejecución en general.

Para trabajos futuros, las placas Arduino podrían reemplazarse por modelos actuales con más recursos para mejorar los tiempos de respuesta a un cliente; otra alternativa seria utilizar dos placas modelo UNO interconectadas, una para lectura de datos, otra para atender peticiones de clientes. La comunicación a través de Ethernet es más rápida en comparación con el WIFI, utilizar este tipo de comunicación disminuiría el tiempo de ejecución, con la desventaja de quitar la portabilidad que el WIFI otorga. Actualmente el API OWL no posee métodos de inserción concurrente. Con la creación y lectura secuencial de Archivos temporales se garantiza la persistencia de los datos, pero eleva el tiempo en la inserción en la ontología, es recomendable buscar un mejor método para la inserción de datos.

## 10. ANEXOS

### 10.1 ANEXO A: Código fuente de un servidor Web Arduino

```
#include <DHT.h>
#include <SPI.h>
#include <WiFi101.h>

#define DHTPIN A0
#define DHTPINB A1

#define DHTTYPE DHT11

DHT dhtA(A0, DHTTYPE);
DHT dhtB(A1, DHTTYPE);

char ssid[] = "INFINITUM7EC7";
char pass[] = "AlphaBetaGammaEpsilon__2016";
int puerto = 80;

float temp1;
float hum1;
```

```

float temp2;
float hum2;

int lum1 = A2;
int lum2 = A3;

float L1val;
float L2val;

IPAddress ip;
String c;

int status = WL_IDLE_STATUS;
WiFiServer server(puerto);

void setup() {
  pinMode(lum1, INPUT);
  pinMode(lum2, INPUT);
  Serial.begin(115200);
  if (WiFi.status() == WL_NO_SHIELD) {
    Serial.println("Verifique la conexion del SHIELD WIFI");
    while (true);
  }
  while (status != WL_CONNECTED) {
    Serial.print("Conectando al SSID: ");
    Serial.println(ssid);
    status = WiFi.begin(ssid, pass);
    delay(3000);
  }
  dhtA.begin();
  dhtB.begin();
  Serial.println();
  server.begin();
  printWifiStatus();
}

void printWifiStatus() {
  Serial.print("SSID: ");
  Serial.println(WiFi.SSID());
  ip = WiFi.localIP();
  Serial.print("IP Address: ");
  Serial.println(ip);
  long rssi = WiFi.RSSI();
  Serial.print("signal strength (RSSI): ");
  Serial.print(rssi);
  Serial.println(" dBm");
}

void loop() {
  Lecturas();
}

```

```

void http() {
  WiFiClient client = server.available();
  if (client)
  {
    if (client.connected() && client.available() > 0)
    {
      Serial.println("Cliente Nuevo");
      c = client.readString();
      Serial.print(c);

      client.println("HTTP/1.1 200 OK");
      client.println("Content-Type: text/html");
      client.println("Connection: close");
      client.println();
      client.println("<!DOCTYPE html>");
      client.println("<html>");
      client.println("<head>");
      client.println("<title></title>");
      client.println("</head>");
      client.println("<body>");
      client.println("<a class=Sensor id = DHT11A>DHT11A</a>");
      client.println("<p></p>");
      client.println("<a name=DHT11A id = hasTempValue>");
      client.print(temp1);
      client.println("</a>");
      client.println("<p></p>");
      client.println("<a name=DHT11A id = hasHumidityValue>");
      client.print(hum1);
      client.println("</a>");
      client.println("<p></p>");
      client.println("<a class=Sensor id = DHT11B>DHT11B</a>");
      client.println("<p></p>");
      client.println("<a name=DHT11B id = hasTempValue>");
      client.print(temp2);
      client.println("</a>");
      client.println("<p></p>");
      client.println("<a name=DHT11B id = hasHumidityValue>");
      client.println(hum2);
      client.println("</a>");
      client.println("<p></p>");
      client.println("<a class=Sensor id = LightA>LightA</a>");
      client.println("<p></p>");
      client.println("<a name=LightA id = hasLightValue>");
      client.println(L1val);
      client.println("</a>");
      client.println("<p></p>");
      client.println("<a class=Sensor id = LightB>LightB</a>");
      client.println("<p></p>");
      client.println("<a name=LightB id = hasLightValue>");
      client.println(L2val);
      client.println("</a>");
      client.println("<p></p>");
      client.println("</body>");
      client.println("</html>");
    }
  }
}

```

```

        client.stop();
    }
}
}

void Lecturas() {
    hum1 = dhtA.readHumidity();
    temp1 = dhtA.readTemperature();
    L1val = analogRead(lum1);
    delay(5);
    hum2 = dhtB.readHumidity();
    temp2 = dhtB.readTemperature();
    L2val = analogRead(lum2);
    http();
}

```

## 10.2 ANEXO B: Módulo de lectura concurrente

### 10.2.1 Código fuente de socket servidor de lectura concurrente

```

public boolean servicioEscrituraConcurrente() {
    try {
        System.out.println("S O C K E T   S E R V I D O R   L E C T U R A   :   3445\n");
        ServerSocket servidor = new ServerSocket(3445, 100);

        lecturaConcurrente lectura = new lecturaConcurrente();
        lectura.servicioLecturaConcurrente();

        while (true) {
            Socket clienteNuevo = servidor.accept();

            ObjectInputStream mensaje = new ObjectInputStream(clienteNuevo.getInputStream());
            String direccion = (String) mensaje.readObject();

            mensaje = new ObjectInputStream(clienteNuevo.getInputStream());
            int i = (Integer) mensaje.readObject();

            System.out.println(clienteNuevo.getInetAddress());
            lecturaSensor sensor = new lecturaSensor(direccion, i);
            sensor.start();
            sensor.join();

            ObjectOutputStream salida = new
ObjectOutputStream(clienteNuevo.getOutputStream());
            salida.writeObject(sensor.DatosSensor);
        }
    } catch (IOException e) {
        System.out.println(e);
    } catch (ClassNotFoundException ex) {
        System.out.println(ex);
    } catch (InterruptedException ex) {
        Logger.getLogger(servidorLecturaConcurrente.class.getName()).log(Level.SEVERE, null,
ex);
    }
}

```

```

    }
    return true;
}
}

```

## 10.2.2 Código fuente de captura concurrente de datos de sensores

```

package LecturaConcurrente;

import java.io.BufferedReader;
import java.io.File;
import java.io.FileReader;
import java.util.Vector;
import java.util.concurrent.Executors;
import java.util.concurrent.ScheduledExecutorService;
import java.util.concurrent.TimeUnit;

/**
 *
 * @author Pedro Romero Flores
 */
public class lecturaConcurrente {

    public Vector leerArchivoConfiguracion() {
        Vector direcciones = new Vector();
        try {
            /*Si existe el fichero*/
            File Ffichero = new File("archivos_configuracion\\direcciones.txt");

            if (Ffichero.exists()) {
                /*Abre un flujo de lectura a el fichero*/
                BufferedReader Flee = new BufferedReader(new FileReader(Ffichero));
                String Slinea;
                while ((Slinea = Flee.readLine()) != null) {
                    direcciones.add(Slinea);
                }
                Flee.close();
            } else {
                System.out.println("Fichero No Existe");
            }

        } catch (Exception ex) {
            /*Captura un posible error y le imprime en pantalla*/
            System.out.println(ex.getMessage());
        }

        return direcciones;
    }

    public boolean servicioLecturaConcurrente() {
        final Runnable tarea = new Runnable() {
            public void run() {
                Vector direcciones;
            }
        };
    }
}

```

```

        direcciones = leerArchivoConfiguracion();
        for (int i = 0; i < direcciones.size(); i++) {
            System.out.println("Extrayendo informacion de: "+(String)
direcciones.elementAt(i));
            lecturaSensor lectura = new lecturaSensor((String)
direcciones.elementAt(i),0);
            lectura.start();
        }
    }
};
ScheduledExecutorService timer = Executors.newSingleThreadScheduledExecutor();
timer.scheduleAtFixedRate(tarea, 0, 15, TimeUnit.SECONDS);

return true;
}
}

```

### 10.2.3 Código fuente de hilo para captura de datos de sensor

```

package LecturaConcurrente;

import java.io.IOException;
import java.io.ObjectOutputStream;
import java.net.Socket;

/**
 *
 * @author Pedro Romero Flores
 */

public class lecturaSensor extends Thread {

    String direccionIP;
    int tipoSensor;
    String DatosSensor;

    public lecturaSensor(String direccionIP, int tipoSensor) {
        this.direccionIP = direccionIP;
        this.tipoSensor = tipoSensor;
    }

    public String getDireccionIP() {
        return direccionIP;
    }

    public void setDireccionIP(String direccionIP) {
        this.direccionIP = direccionIP;
    }

    public int getTipoSensor() {
        return tipoSensor;
    }
}

```

```

public void setTipoSensor(int tipoSensor) {
    this.tipoSensor = tipoSensor;
}

public void enviarDatos() {

    try {

        Socket cliente = new Socket("127.0.0.1", 3444);
        ObjectOutputStream mensaje = new ObjectOutputStream(cliente.getOutputStream());
        mensaje.writeObject(this.DatosSensor);
        cliente.close();
    } catch (IOException ex) {
        System.out.println(ex);
    }

}

public void run() {
    parser parser = new parser(getDireccionIP());
    DatosSensor = parser.Parser(getTipoSensor());
    if (this.DatosSensor.length() > 0)
    {
        enviarDatos();
    }
}
}

```

#### 10.2.4 Código fuente de procesamiento de datos

```

import java.io.IOException;
import java.io.ObjectOutputStream;
import java.net.Socket;
import java.util.Vector;
import org.jsoup.Connection;
import org.jsoup.Jsoup;
import org.jsoup.nodes.Document;
import org.jsoup.nodes.Element;
import org.jsoup.select.Elements;

/**
 *
 * @author Pedro Romero Flores
 */

public class parser {

    String Direccion;
    String DatosSensor;

    public parser(String Direccion) {
        this.Direccion = Direccion;
    }
}

```

```

        this.DatosSensor="";
    }

    public String getDireccion() {
        return Direccion;
    }

    public boolean setDireccion(String Direccion) {
        this.Direccion = Direccion;
        return true;
    }

    public boolean enviar(String individuo, String relacion, String valor) {
        this.DatosSensor = DatosSensor +individuo + "|" + relacion + "|" + valor +"\n";
        return true;
    }

    //public boolean Parser(int tipoSensor)
    public String Parser(int tipoSensor)
    {
        // Compruebo si me da un 200 al hacer la petición

        boolean retorno = false;

        //Faltaba # al final del IRI, solamente así comenzo a escribir en la ontología;
        try {

            if (getStatusConnectionCode(Direccion) == 200) {

                //Obtiene el html de la direccion
                Document documentHTML = getHtmlDocument(Direccion);

                //Dato Tipo Elements en donde se almacenaran el resultado de las consultas para
                buscar "Clase e individuo" y la(s) "DataPropierties del individuo"
                Elements ClassIndividual, DataPropierties;

                //a[Class] - > Parametro que identifica a la clase      ****      a[id] -> Al
                estar junto del parametro Class identifica al individuo
                //a[Name] - > Parametro que identifica al individuo    ****      a[id] -> al
                estar junto del parametro Name idnetifica la data propierties, el value HTML es el value de la data
                propierties
                //ClassIndividual = documentHTML.select("a[class]");
                ClassIndividual = documentHTML.select("a[class]");

                if (ClassIndividual.size() > 0) {
                    for (Element elem : ClassIndividual) {
                        DataPropierties = documentHTML.select(" a[name^=" + valorTagParam("id",
                        elem.toString()) + "]");

                        for (Element elem2 : DataPropierties) {
                            if (tipoSensor == 0 && (obtenerTemperatura(valorTagParam("id",
                            elem2.toString()))
                                || obtenerHumedad(valorTagParam("id", elem2.toString()))

```



```

        || obtenerIntensidadLuminosa(valorTagParam("id",
elem2.toString())))) {

        enviar(
            valorTagParam("id", cleanString(elem.toString())),
            valorTagParam("id", elem2.toString()),
            cleanString(

elem2.toString().substring(elem2.toString().indexOf(">") + 1,
                                                                    elem2.toString().indexOf("<",
(elem2.toString().indexOf(">"))))
                )
        );
    } else if (tipoSensor == 1 && obtenerTemperatura(valorTagParam("id",
elem2.toString())) {
        enviar(
            valorTagParam("id", cleanString(elem.toString())),
            valorTagParam("id", elem2.toString()),
            cleanString(

elem2.toString().substring(elem2.toString().indexOf(">") + 1,
                                                                    elem2.toString().indexOf("<",
(elem2.toString().indexOf(">"))))
                )
        );
    } else if (tipoSensor == 2 && obtenerHumedad(valorTagParam("id",
elem2.toString())) {
        enviar(
            valorTagParam("id", cleanString(elem.toString())),
            valorTagParam("id", elem2.toString()),
            cleanString(

elem2.toString().substring(elem2.toString().indexOf(">") + 1,
                                                                    elem2.toString().indexOf("<",
(elem2.toString().indexOf(">"))))
                )
        );
    } else if (tipoSensor == 3 &&
obtenerIntensidadLuminosa(valorTagParam("id", elem2.toString())) {
        enviar(
            valorTagParam("id", cleanString(elem.toString())),
            valorTagParam("id", elem2.toString()),
            cleanString(

elem2.toString().substring(elem2.toString().indexOf(">") + 1,
                                                                    elem2.toString().indexOf("<",
(elem2.toString().indexOf(">"))))
                )
        );
    } else if (tipoSensor == 4 &&
obtenerPresenciaObjeto(valorTagParam("id", elem2.toString())) {
        enviar(
            valorTagParam("id", cleanString(elem.toString())),
            valorTagParam("id", elem2.toString()),

```

```

                cleanString(
elem2.toString().substring(elem2.toString().indexOf(">") + 1,
                                                                    elem2.toString().indexOf("<"),
(elem2.toString().indexOf(">"))))
            )
        );
    }
}
}
returno = true;
}
} else {
    System.out.println("El Status Code no es OK es: " +
getStatusConnectionCode(Direccion));
    returno = false;
}
} catch (NullPointerException e) {
    returno = false;
}

return this.DatosSensor;
}

public boolean obtenerTemperatura(String dataProperty) {
    String[] arreglo = {"temp", "Temp", "TEMP"};
    boolean returno = false;
    for (int i = 0; i < arreglo.length; i++) {
        if (dataProperty.contains(arreglo[i])) {
            returno = true;
            break;
        }
    }
    return returno;
}

public boolean obtenerHumedad(String dataProperty) {
    String[] arreglo = {"humidity", "HUMIDITY", "Humidity"};
    boolean returno = false;
    for (int i = 0; i < arreglo.length; i++) {
        if (dataProperty.contains(arreglo[i])) {
            returno = true;
            break;
        }
    }
    return returno;
}

public boolean obtenerIntensidadLuminosa(String dataProperty) {
    String[] arreglo = {"light", "LIGHT", "Light"};
    boolean returno = false;

```

```

        for (int i = 0; i < arreglo.length; i++) {
            if (dataProperty.contains(arreglo[i])) {
                retorno = true;
                break;
            }
        }
        return retorno;
    }

    public boolean obtenerPresenciaObjeto(String dataProperty) {
        String[] arreglo = {"RFID", "", "zzz"};
        boolean retorno = false;

        for (int i = 0; i < arreglo.length; i++) {
            if (dataProperty.contains(arreglo[i])) {
                retorno = true;
                break;
            }
        }
        return false;
    }

    public String cleanString(String cadena) {
        return cadena.replace(" ", "");
    }

    public String valorTagParam(String TagParam, String cadena) {
        int posTagParam = 0, posIni = 0, posFin = 0;

        posTagParam = cadena.indexOf(TagParam);
        posIni = cadena.indexOf("'", posTagParam + 1);
        posFin = cadena.indexOf("'", posIni + 1);

        return cadena.substring(posIni + 1, posFin);
    }

    public int getStatusConnectionCode(String url) {

        Connection.Response response = null;

        try {
            response =
Jsoup.connect(url).userAgent("Mozilla/5.0").timeout(100000).ignoreHttpErrors(true).execute();
        } catch (IOException ex) {
            System.out.println("Excepción al obtener el Status Code: " + ex.getMessage());
        }
        catch (IllegalArgumentException ex){
            System.out.println("Direccion invalida: " + ex.getMessage());
        }

        return response.statusCode();
    }
}

```

```

public Document getHtmlDocument(String url) {

    Document doc = null;
    try {
        doc = Jsoup.connect(url).userAgent("Mozilla/5.0").timeout(100000).get();
    } catch (IOException ex) {
        System.out.println("Excepción al obtener el HTML de la página" + ex.getMessage());
    }
    return doc;
}
}

```

### 10.3 ANEXO C: Código fuente de módulo de visualización

```

package ModuloVisualizacion;

import java.io.IOException;
import java.io.ObjectInputStream;
import java.io.ObjectOutputStream;
import java.net.Socket;
import java.util.StringTokenizer;
import java.util.logging.Level;

/**
 *
 * @author Pedro Romero Flores
 */
public class ModuloVisualizacion extends javax.swing.JFrame {

    /**
     * Creates new form ModuloVisualizacion
     */
    public ModuloVisualizacion() {
        initComponents();
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        direccion = new javax.swing.JTextField();
        magnitudFisica = new javax.swing.JComboBox<>();
        enviar = new javax.swing.JButton();
        jLabel1 = new javax.swing.JLabel();
        jLabel2 = new javax.swing.JLabel();
        jScrollPane1 = new javax.swing.JScrollPane();
        resultados = new javax.swing.JTextArea();
    }
}

```

```

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

magnitudFisica.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] { "Todas las
magnitudes", "Temperatura", "Humedad", "Intensidad Luminosa" }));

enviar.setText("ENVIAR");
enviar.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        enviarMouseClicked(evt);
    }
});

jLabel1.setText("Direccion de Servidor:");

jLabel2.setText("Resultados");

resultados.setColumns(20);
resultados.setRows(5);
jScrollPane1.setViewportView(resultados);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addContainerGap()
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 123,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGap(66, 66, 66)
            .addComponent(jLabel2)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addContainerGap()
                .addComponent(magnitudFisica, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(70, 70, 70)
                .addComponent(enviar, javax.swing.GroupLayout.PREFERRED_SIZE, 204,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGroup(layout.createSequentialGroup()
                .addContainerGap()
                .addComponent(direccion, javax.swing.GroupLayout.PREFERRED_SIZE, 492,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGroup(layout.createSequentialGroup()
                .addContainerGap()
                .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 337,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGap(271, 271, 271))
    );

```

```

        layout.setVerticalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(layout.createSequentialGroup()
                    .addGap(24, 24, 24)
                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                        .addComponent(direccion, javax.swing.GroupLayout.PREFERRED_SIZE, 35,
                            javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addComponent(jLabel1, javax.swing.GroupLayout.DEFAULT_SIZE,
                            javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
                    .addGap(18, 18, 18)
                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                        .addComponent(enviar)
                        .addComponent(magnitudFisica, javax.swing.GroupLayout.PREFERRED_SIZE,
                            javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                        .addGroup(layout.createSequentialGroup()
                            .addGap(213, 213, 213)
                            .addComponent(jLabel2))
                        .addGroup(layout.createSequentialGroup()
                            .addGap(18, 18, 18)
                            .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 343,
                                javax.swing.GroupLayout.PREFERRED_SIZE)))
                    .addContainerGap(64, Short.MAX_VALUE))
        );

        pack();
    } // </editor-fold>

    private void enviarMouseClicked(java.awt.event.MouseEvent evt) {
        // TODO add your handling code here:
        int tipoMagnitud = magnitudFisica.getSelectedIndex();
        String ip = this.direccion.getText();

        try {

            Socket cliente = new Socket("127.0.0.1", 3445);

            ObjectOutputStream direccionIPParser = new
                ObjectOutputStream(cliente.getOutputStream());
            direccionIPParser.writeObject(ip);

            ObjectOutputStream tipoSensor = new ObjectOutputStream(cliente.getOutputStream());
            tipoSensor.writeObject(tipoMagnitud);

            ObjectInputStream mensaje = new ObjectInputStream(cliente.getInputStream());

            StringTokenizer token = new StringTokenizer((String)mensaje.readObject(), "|");
            String aux="";
            cliente.close();

            while (token.hasMoreTokens())
            {
                aux = aux + token.nextToken()+" ";
            }
        }
    }

```

```

        this.resultados.setText((String) aux);

    } catch (IOException ex) {
        this.resultados.setText(ex.getMessage());
    } catch (ClassNotFoundException ex) {
        this.resultados.setText(ex.getMessage());
    }
}

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and
feel.
    * For details see
http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
    */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {

        java.util.logging.Logger.getLogger(ModuloVisualizacion.class.getName()).log(java.util.logging.Level
1.SEVERE, null, ex);
    } catch (InstantiationException ex) {

        java.util.logging.Logger.getLogger(ModuloVisualizacion.class.getName()).log(java.util.logging.Level
1.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {

        java.util.logging.Logger.getLogger(ModuloVisualizacion.class.getName()).log(java.util.logging.Level
1.SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {

        java.util.logging.Logger.getLogger(ModuloVisualizacion.class.getName()).log(java.util.logging.Level
1.SEVERE, null, ex);
    }
}
//</editor-fold>

    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new ModuloVisualizacion().setVisible(true);
        }
    }
}

```

```

    });
}

// Variables declaration - do not modify
private javax.swing.JTextField direccion;
private javax.swing.JButton enviar;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JComboBox<String> magnitudFisica;
private javax.swing.JTextArea resultados;
// End of variables declaration
}

```

## 10.4 ANEXO D: Módulo de escritura concurrente

### 10.4.1 Código fuente de socket servidor de escritura concurrente

```

package EscrituraConcurrente;

import java.io.IOException;
import java.io.ObjectInputStream;
import java.net.ServerSocket;
import java.net.Socket;
import java.util.concurrent.Executors;
import java.util.concurrent.ScheduledExecutorService;
import java.util.concurrent.TimeUnit;

/**
 *
 * @author Pedro Romero Flores
 */
public class servidorEscrituraConcurrente {

    public boolean servicioEscrituraOntologia() {
        final Runnable escribirEnOntologia = new Runnable() {
            operacionesOntologia ontologia = new operacionesOntologia("OntologyDevices.owl",
"http://www.semanticweb.org/usuario/ontologies/2016/0/OntologyDevices");

            public void run() {
                lecturaArchivosTemporales lectura = new lecturaArchivosTemporales(ontologia);
                lectura.servicioLecturaArchivosTemporales();
            }
        };

        ScheduledExecutorService timer = Executors.newSingleThreadScheduledExecutor();
        timer.scheduleAtFixedRate(escribirEnOntologia, 1, 1, TimeUnit.SECONDS);
        return true;
    }
}

```



```

public boolean servicioEscrituraConcurrente() {
    try {
        System.out.println("S O C K E T   S E R V I D O R   E S C R I T U R A   :   3446\n");
        ServerSocket servidor = new ServerSocket(3446, 100);

        servicioEscrituraOntologia();

        while (true) {
            Socket clienteNuevo = servidor.accept();
            ObjectInputStream mensaje = new ObjectInputStream(clienteNuevo.getInputStream());
            System.out.println(clienteNuevo.getInetAddress());
            System.out.println("Recibiendo Informacion");
            escrituraArchivosTemporales h = new escrituraArchivosTemporales((String)
mensaje.readObject());
            h.start();
        }

        } catch (IOException e) {
            System.out.println(e);
        } catch (ClassNotFoundException ex) {
            System.out.println(ex);
        }
        }
        return true;
    }
}
}

```

#### 10.4.2 Código fuente de hilo para creación de archivos temporales

```

package EscrituraConcurrente;

import java.io.BufferedWriter;
import java.io.File;
import java.io.FileWriter;
import java.io.IOException;

/**
 *
 * @author Pedro Romero Flores
 */
public class escrituraArchivosTemporales extends Thread {
    String cadenaTexto;

    public escrituraArchivosTemporales(String individuo) {
        this.cadenaTexto = individuo;
    }

    public String getCadenaTexto() {
        return cadenaTexto;
    }
}

```

```

public void setCadenaTexto(String cadenaTexto) {
    this.cadenaTexto = cadenaTexto;
}

public void run(){
    escribirArchivoTemporal(getCadenaTexto());
}

public void escribirArchivoTemporal(String cadena) {
    try {

        File Ffichero = new File("archivos_temporales");
        File tempFile = File.createTempFile("temporal","",Ffichero);
        BufferedWriter out = new BufferedWriter(new FileWriter(tempFile));
        out.write(cadena);
        out.close();
    } catch (IOException ex) {
        System.out.println(ex);
    }
}
}
}

```

### 10.4.3 Código fuente de lectura secuencial de archivos temporales

```

package EscrituraConcurrente;

import java.io.BufferedReader;
import java.io.File;
import java.io.FileReader;
import java.util.StringTokenizer;

/**
 *
 * @author Pedro Romero Flores
 */
public class lecturaArchivosTemporales {

    operacionesOntologia operador;

    public lecturaArchivosTemporales(operacionesOntologia load) {
        // this.operador = new operacionesOntologia("OntologyDevices.owl",
"http://www.semanticweb.org/usuario/ontologies/2016/0/OntologyDevices");
        this.operador= load;
    }

    public boolean servicioLecturaArchivosTemporales() {
        String path = "archivos_temporales";
        String files;
        File folder = new File(path);
        File[] listOffiles = folder.listFiles();
    }
}

```

```

if (listOfFiles.length > 0) {
    //operador.load();
    for (int i = 0; i < listOfFiles.length; i++) {
        if (listOfFiles[i].isFile()) {
            files = listOfFiles[i].getName();
            if (files.startsWith("temporal")) {
                leerArchivoTemporal(path + "\\\" + files);
            }
        }
    }
}

} else {
    //System.out.println("Directorio vacio :(");
}
return true;
}

public void leerArchivoTemporal(String archivo) {
    try {
        /*Si existe el fichero*/
        File Ffichero = new File(archivo);
        String aux,aux1,aux2;
        if (Ffichero.exists()) {
            /*Abre un flujo de lectura a el fichero*/
            BufferedReader Flee = new BufferedReader(new FileReader(Ffichero));
            String Slinea;
            StringTokenizer token;
            while ((Slinea = Flee.readLine()) != null) {
                token = new StringTokenizer(Slinea, "|");
                aux=token.nextToken();
                aux1=token.nextToken();
                aux2=token.nextToken();

                System.out.println(aux+" "+aux1+" "+aux2);
                operador.write(aux, aux1, Float.parseFloat(aux2));

            }
            Flee.close();
            Ffichero.delete();
        } else {
            System.out.println("Fichero No Existe");
        }
    } catch (Exception ex) {
        /*Captura un posible error y le imprime en pantalla*/
        System.out.println(ex.getMessage());
    }
}
}

```

#### 10.4.4 Código fuente de métodos para administrar ontologías con Java

```
package EscrituraConcurrente;

import java.io.File;
import org.semanticweb.owlapi.apibinding.OWLManager;
import org.semanticweb.owlapi.model.IRI;
import org.semanticweb.owlapi.model.OWLDataFactory;
import org.semanticweb.owlapi.model.OWLDataProperty;
import org.semanticweb.owlapi.model.OWLDataPropertyAssertionAxiom;
import org.semanticweb.owlapi.model.OWLLiteral;
import org.semanticweb.owlapi.model.OWLNamedIndividual;
import org.semanticweb.owlapi.model.OWLOntology;
import org.semanticweb.owlapi.model.OWLOntologyCreationException;
import org.semanticweb.owlapi.model.OWLOntologyManager;
import org.semanticweb.owlapi.model.OWLOntologyStorageException;

/**
 *
 * @author Pedro Romero Flores
 */
public class operacionesOntologia {

    OWLOntology ontologia;
    IRI ontologyIRI;
    String iri;
    String ontoFile;
    OWLOntologyManager manager;

    public operacionesOntologia(String ontoFile, String iri) {
        this.iri = iri+"#";
        this.ontoFile = ontoFile;
        this.load();
    }

    public boolean load() {
        boolean retorno = false;
        try {
            manager = OWLManager.createOWLOntologyManager();
            ontologyIRI = IRI.create(iri);
            IRI ontoIri = IRI.create(new File(ontoFile));
            ontologia = manager.loadOntologyFromOntologyDocument(ontoIri);
            retorno = true;
        } catch (OWLOntologyCreationException | java.lang.NullPointerException e) {
            System.out.println(e);
            retorno = false;
        }
        return retorno;
    }

    public boolean write(String objName, String property, float value) {
        boolean retorno = false;

        try {
```

```

        OWLDataFactory factory = manager.getOWLDataFactory();
        IRI dataPropertyName = generateIRI(property, ontologyIRI);

        OWLDataProperty owlDatatype = factory.getOWLDataProperty(dataPropertyName); /*Obtiene
una data property de un IRI*/
        IRI namedIndividual = generateIRI(objName, ontologyIRI); /*Regresa un individuo a
partir de una IRI*/
        OWLNamedIndividual individual =
factory.getOWLNamedIndividual(namedIndividual); /*Regresa un individuo a partir de una IRI*/
        //OWLLiteral owlLiteral = factory.getOWLStringLiteral(value);
        OWLLiteral owlLiteral = factory.getOWLTypedLiteral(value);
        OWLDataPropertyAssertionAxiom dataProperty =
factory.getOWLDataPropertyAssertionAxiom(owlDatatype, individual, owlLiteral); /*Genera un Axioma
para una dataPropertyAssertion*/

        manager.addAxiom(ontologia, dataProperty);
        manager.saveOntology(ontologia);

        retorno = true;
    } catch (OWLOntologyStorageException e) {
        System.out.println(retorno);
        retorno = false;
    }
    return retorno;
}

private IRI generateIRI(String clase, IRI ontologyIRI) {
    IRI ontologyI = IRI.create(ontologyIRI + clase);
    return ontologyI;
}
}
}

```

#### 10.4.5 Código fuente de inserción en ontología

```

package EscrituraConcurrente;

import java.io.BufferedReader;
import java.io.File;
import java.io.FileReader;
import java.util.StringTokenizer;

/**
 *
 * @author Pedro Romero Flores
 */
public class lecturaArchivosTemporales {

    operacionesOntologia operador;

    public lecturaArchivosTemporales(operacionesOntologia load) {

```

```

        // this.operador = new operacionesOntologia("OntologyDevices.owl",
"http://www.semanticweb.org/usuario/ontologies/2016/0/OntologyDevices");
        this.operador= load;
    }

    public boolean servicioLecturaArchivosTemporales() {
        String path = "archivos_temporales";
        String files;
        File folder = new File(path);
        File[] listOffFiles = folder.listFiles();
        if (listOffFiles.length > 0) {
            //operador.load();
            for (int i = 0; i < listOffFiles.length; i++) {
                if (listOffFiles[i].isFile()) {
                    files = listOffFiles[i].getName();
                    if (files.startsWith("temporal")) {
                        leerArchivoTemporal(path + "\\\" + files);
                    }
                }
            }
        }
        } else {
            //System.out.println("Directorio vacio :(");
        }
        return true;
    }

    public void leerArchivoTemporal(String archivo) {
        try {
            /*Si existe el fichero*/
            File Ffichero = new File(archivo);
            String aux,aux1,aux2;
            if (Ffichero.exists()) {
                /*Abre un flujo de lectura a el fichero*/
                BufferedReader Flee = new BufferedReader(new FileReader(Ffichero));
                String Slinea;
                StringTokenizer token;
                while ((Slinea = Flee.readLine()) != null) {
                    token = new StringTokenizer(Slinea, "|");
                    aux=token.nextToken();
                    aux1=token.nextToken();
                    aux2=token.nextToken();

                    System.out.println(aux+" "+aux1+" "+aux2);
                    operador.write(aux, aux1, Float.parseFloat(aux2));

                }
                Flee.close();
                Ffichero.delete();
            } else {
                System.out.println("Fichero No Existe");
            }
        } catch (Exception ex) {
            /*Captura un posible error y le imprime en pantalla*/

```

```

        System.out.println(ex.getMessage());
    }
}
}

```

## 10.5 ANEXO E: Modelo de datos semántico OntologyDevices

```

<?xml version="1.0"?>
<Ontology xmlns="http://www.w3.org/2002/07/owl#"
  xmlns:base="http://www.semanticweb.org/usuario/ontologies/2016/0/OntologyDevices"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:xml="http://www.w3.org/XML/1998/namespace"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  ontologyIRI="http://www.semanticweb.org/usuario/ontologies/2016/0/OntologyDevices">
  <Prefix name="swrl" IRI="http://www.w3.org/2003/11/swrl#" />
  <Prefix name="owl" IRI="http://www.w3.org/2002/07/owl#" />
  <Prefix name="rdf" IRI="http://www.w3.org/1999/02/22-rdf-syntax-ns#" />
  <Prefix name="swrla" IRI="http://swrl.stanford.edu/ontologies/3.3/swrla.owl#" />
  <Prefix name="protege" IRI="http://protege.stanford.edu/plugins/owl/protege#" />
  <Prefix name="xsd" IRI="http://www.w3.org/2001/XMLSchema#" />
  <Prefix name="swrlb" IRI="http://www.w3.org/2003/11/swrlb#" />
  <Prefix name="" IRI="http://www.semanticweb.org/usuario/ontologies/2016/0/OntologyDevices" />
  <Prefix name="xml" IRI="http://www.w3.org/XML/1998/namespace" />
  <Prefix name="xsp" IRI="http://www.owl-ontologies.com/2005/08/07/xsp.owl#" />
  <Prefix name="rdfs" IRI="http://www.w3.org/2000/01/rdf-schema#" />
  <Declaration>
    <Class IRI="#Actuator" />
  </Declaration>
  <Declaration>
    <Class IRI="#Device" />
  </Declaration>
  <Declaration>
    <Class IRI="#Distance" />
  </Declaration>
  <Declaration>
    <Class IRI="#Humidity" />
  </Declaration>
  <Declaration>
    <Class IRI="#LightIntensity" />
  </Declaration>
  <Declaration>
    <Class IRI="#PhysicalMeasurement" />
  </Declaration>
  <Declaration>
    <Class IRI="#RFIDCard" />
  </Declaration>
  <Declaration>
    <Class IRI="#Sensor" />
  </Declaration>
  <Declaration>
    <Class IRI="#SmartPhone" />
  </Declaration>

```

```

</Declaration>
<Declaration>
  <Class IRI="#Temperature"/>
</Declaration>
<Declaration>
  <ObjectProperty IRI="#hasOwner"/>
</Declaration>
<Declaration>
  <ObjectProperty IRI="#itIsLocatedIn"/>
</Declaration>
<Declaration>
  <ObjectProperty IRI="#read"/>
</Declaration>
<Declaration>
  <DataProperty IRI="#avgHum"/>
</Declaration>
<Declaration>
  <DataProperty IRI="#avgLight"/>
</Declaration>
<Declaration>
  <DataProperty IRI="#avgTemp"/>
</Declaration>
<Declaration>
  <DataProperty IRI="#hasBrand"/>
</Declaration>
<Declaration>
  <DataProperty IRI="#hasHumidityValue"/>
</Declaration>
<Declaration>
  <DataProperty IRI="#hasIDCard"/>
</Declaration>
<Declaration>
  <DataProperty IRI="#hasIMEI"/>
</Declaration>
<Declaration>
  <DataProperty IRI="#hasIPAddress"/>
</Declaration>
<Declaration>
  <DataProperty IRI="#hasKind"/>
</Declaration>
<Declaration>
  <DataProperty IRI="#hasLightValue"/>
</Declaration>
<Declaration>
  <DataProperty IRI="#hasMacAddress"/>
</Declaration>
<Declaration>
  <DataProperty IRI="#hasModel"/>
</Declaration>
<Declaration>
  <DataProperty IRI="#hasNameDevice"/>
</Declaration>
<Declaration>
  <DataProperty IRI="#hasNumber"/>

```



```

</Declaration>
<Declaration>
  <DataProperty IRI="#hasNumberOfIdentification"/>
</Declaration>
<Declaration>
  <DataProperty IRI="#hasPhoneNumber"/>
</Declaration>
<Declaration>
  <DataProperty IRI="#hasState"/>
</Declaration>
<Declaration>
  <DataProperty IRI="#hasSymbol"/>
</Declaration>
<Declaration>
  <DataProperty IRI="#hasTempValue"/>
</Declaration>
<Declaration>
  <DataProperty IRI="#idCardRead"/>
</Declaration>
<Declaration>
  <DataProperty IRI="#idNumberRead"/>
</Declaration>
<Declaration>
  <DataProperty IRI="#itIsActive"/>
</Declaration>
<Declaration>
  <DataProperty IRI="#readA"/>
</Declaration>
<Declaration>
  <NamedIndividual IRI="#AC"/>
</Declaration>
<Declaration>
  <NamedIndividual IRI="#CelsiusDegrees"/>
</Declaration>
<Declaration>
  <NamedIndividual IRI="#Centimeters"/>
</Declaration>
<Declaration>
  <NamedIndividual IRI="#DHT11A"/>
</Declaration>
<Declaration>
  <NamedIndividual IRI="#DHT11B"/>
</Declaration>
<Declaration>
  <NamedIndividual IRI="#FahrenheitDegrees"/>
</Declaration>
<Declaration>
  <NamedIndividual IRI="#Fan1"/>
</Declaration>
<Declaration>
  <NamedIndividual IRI="#Inches"/>
</Declaration>
<Declaration>
  <NamedIndividual IRI="#LightA"/>

```

```

</Declaration>
<Declaration>
  <NamedIndividual IRI="#LightB"/>
</Declaration>
<Declaration>
  <NamedIndividual IRI="#Lum1"/>
</Declaration>
<Declaration>
  <NamedIndividual IRI="#RFIDReader"/>
</Declaration>
<Declaration>
  <NamedIndividual IRI="#S5"/>
</Declaration>
<Declaration>
  <NamedIndividual IRI="#SonyX1"/>
</Declaration>
<Declaration>
  <NamedIndividual IRI="#Tarjeta00"/>
</Declaration>
<Declaration>
  <NamedIndividual IRI="#humidity"/>
</Declaration>
<Declaration>
  <NamedIndividual IRI="#lum"/>
</Declaration>
<Declaration>
  <AnnotationProperty abbreviatedIRI="swrla:isRuleEnabled"/>
</Declaration>
<EquivalentClasses>
  <Class IRI="#Actuator"/>
  <DataExactCardinality cardinality="1">
    <DataProperty IRI="#hasState"/>
    <Datatype abbreviatedIRI="xsd:int"/>
  </DataExactCardinality>
</EquivalentClasses>
<EquivalentClasses>
  <Class IRI="#Actuator"/>
  <DataExactCardinality cardinality="1">
    <DataProperty IRI="#itIsActive"/>
    <Datatype abbreviatedIRI="xsd:boolean"/>
  </DataExactCardinality>
</EquivalentClasses>
<EquivalentClasses>
  <Class IRI="#Device"/>
  <DataExactCardinality cardinality="1">
    <DataProperty IRI="#hasBrand"/>
    <Datatype abbreviatedIRI="xsd:string"/>
  </DataExactCardinality>
</EquivalentClasses>
<EquivalentClasses>
  <Class IRI="#Device"/>
  <DataExactCardinality cardinality="1">
    <DataProperty IRI="#hasModel"/>
    <Datatype abbreviatedIRI="xsd:string"/>
  </DataExactCardinality>
</EquivalentClasses>

```

```

    </DataExactCardinality>
</EquivalentClasses>
<EquivalentClasses>
  <Class IRI="#RFIDCard"/>
  <DataExactCardinality cardinality="1">
    <DataProperty IRI="#hasIDCard"/>
    <Datatype abbreviatedIRI="xsd:string"/>
  </DataExactCardinality>
</EquivalentClasses>
<EquivalentClasses>
  <Class IRI="#Sensor"/>
  <ObjectMinCardinality cardinality="1">
    <ObjectProperty IRI="#read"/>
  <Class IRI="#PhysicalMeasurement"/>
</ObjectMinCardinality>
</EquivalentClasses>
<EquivalentClasses>
  <Class IRI="#SmartPhone"/>
  <DataExactCardinality cardinality="1">
    <DataProperty IRI="#hasIMEI"/>
    <Datatype abbreviatedIRI="xsd:string"/>
  </DataExactCardinality>
</EquivalentClasses>
<EquivalentClasses>
  <Class IRI="#SmartPhone"/>
  <DataExactCardinality cardinality="1">
    <DataProperty IRI="#hasMacAddress"/>
    <Datatype abbreviatedIRI="xsd:string"/>
  </DataExactCardinality>
</EquivalentClasses>
<SubClassOf>
  <Class IRI="#Actuator"/>
  <Class IRI="#Device"/>
</SubClassOf>
<SubClassOf>
  <Class IRI="#Distance"/>
  <Class IRI="#PhysicalMeasurement"/>
</SubClassOf>
<SubClassOf>
  <Class IRI="#Humidity"/>
  <Class IRI="#PhysicalMeasurement"/>
</SubClassOf>
<SubClassOf>
  <Class IRI="#LightIntensity"/>
  <Class IRI="#PhysicalMeasurement"/>
</SubClassOf>
<SubClassOf>
  <Class IRI="#RFIDCard"/>
  <Class IRI="#Device"/>
</SubClassOf>
<SubClassOf>
  <Class IRI="#Sensor"/>
  <Class IRI="#Device"/>
</SubClassOf>

```

```

<SubClassOf>
  <Class IRI="#SmartPhone"/>
  <Class IRI="#Device"/>
</SubClassOf>
<SubClassOf>
  <Class IRI="#Temperature"/>
  <Class IRI="#PhysicalMeasurement"/>
</SubClassOf>
<ClassAssertion>
  <Class IRI="#Actuator"/>
  <NamedIndividual IRI="#AC"/>
</ClassAssertion>
<ClassAssertion>
  <Class IRI="#Temperature"/>
  <NamedIndividual IRI="#CelsiusDegrees"/>
</ClassAssertion>
<ClassAssertion>
  <Class IRI="#Distance"/>
  <NamedIndividual IRI="#Centimeters"/>
</ClassAssertion>
<ClassAssertion>
  <Class IRI="#Sensor"/>
  <NamedIndividual IRI="#DHT11A"/>
</ClassAssertion>
<ClassAssertion>
  <Class IRI="#Sensor"/>
  <NamedIndividual IRI="#DHT11B"/>
</ClassAssertion>
<ClassAssertion>
  <Class IRI="#Temperature"/>
  <NamedIndividual IRI="#FahrenheitDegrees"/>
</ClassAssertion>
<ClassAssertion>
  <Class IRI="#Actuator"/>
  <NamedIndividual IRI="#Fan1"/>
</ClassAssertion>
<ClassAssertion>
  <Class IRI="#Distance"/>
  <NamedIndividual IRI="#Inches"/>
</ClassAssertion>
<ClassAssertion>
  <Class IRI="#Sensor"/>
  <NamedIndividual IRI="#LightA"/>
</ClassAssertion>
<ClassAssertion>
  <Class IRI="#Sensor"/>
  <NamedIndividual IRI="#LightB"/>
</ClassAssertion>
<ClassAssertion>
  <Class IRI="#Actuator"/>
  <NamedIndividual IRI="#Lum1"/>
</ClassAssertion>
<ClassAssertion>
  <Class IRI="#Sensor"/>

```

```

    <NamedIndividual IRI="#RFIDReader"/>
</ClassAssertion>
<ClassAssertion>
  <Class IRI="#SmartPhone"/>
  <NamedIndividual IRI="#S5"/>
</ClassAssertion>
<ClassAssertion>
  <Class IRI="#SmartPhone"/>
  <NamedIndividual IRI="#SonyX1"/>
</ClassAssertion>
<ClassAssertion>
  <Class IRI="#RFIDCard"/>
  <NamedIndividual IRI="#Tarjeta00"/>
</ClassAssertion>
<ClassAssertion>
  <Class IRI="#Humidity"/>
  <NamedIndividual IRI="#humidity"/>
</ClassAssertion>
<ClassAssertion>
  <Class IRI="#LightIntensity"/>
  <NamedIndividual IRI="#lum"/>
</ClassAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#read"/>
  <NamedIndividual IRI="#DHT11A"/>
  <NamedIndividual IRI="#humidity"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#read"/>
  <NamedIndividual IRI="#DHT11A"/>
  <NamedIndividual IRI="#CelsiusDegrees"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#read"/>
  <NamedIndividual IRI="#DHT11B"/>
  <NamedIndividual IRI="#humidity"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#read"/>
  <NamedIndividual IRI="#DHT11B"/>
  <NamedIndividual IRI="#CelsiusDegrees"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#read"/>
  <NamedIndividual IRI="#LightA"/>
  <NamedIndividual IRI="#lum"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#read"/>
  <NamedIndividual IRI="#LightB"/>
  <NamedIndividual IRI="#lum"/>
</ObjectPropertyAssertion>
<DataPropertyAssertion>
  <DataProperty IRI="#hasBrand"/>

```

```

    <NamedIndividual IRI="#AC"/>
    <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">air plus</Literal>
</DataPropertyAssertion>
<DataPropertyAssertion>
    <DataProperty IRI="#hasIPAddress"/>
    <NamedIndividual IRI="#AC"/>
    <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">192.168.2.20</Literal>
</DataPropertyAssertion>
<DataPropertyAssertion>
    <DataProperty IRI="#hasModel"/>
    <NamedIndividual IRI="#AC"/>
    <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">gsm01223</Literal>
</DataPropertyAssertion>
<DataPropertyAssertion>
    <DataProperty IRI="#hasNameDevice"/>
    <NamedIndividual IRI="#AC"/>
    <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">Aire
Acondicionado</Literal>
</DataPropertyAssertion>
<DataPropertyAssertion>
    <DataProperty IRI="#hasState"/>
    <NamedIndividual IRI="#AC"/>
    <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#int">0</Literal>
</DataPropertyAssertion>
<DataPropertyAssertion>
    <DataProperty IRI="#hasSymbol"/>
    <NamedIndividual IRI="#CelsiusDegrees"/>
    <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">C</Literal>
</DataPropertyAssertion>
<DataPropertyAssertion>
    <DataProperty IRI="#readA"/>
    <NamedIndividual IRI="#CelsiusDegrees"/>
    <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">celcius</Literal>
</DataPropertyAssertion>
<DataPropertyAssertion>
    <DataProperty IRI="#hasSymbol"/>
    <NamedIndividual IRI="#Centimeters"/>
    <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">cm</Literal>
</DataPropertyAssertion>
<DataPropertyAssertion>
    <DataProperty IRI="#readA"/>
    <NamedIndividual IRI="#Centimeters"/>
    <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">Centimetros</Literal>
</DataPropertyAssertion>
<DataPropertyAssertion>
    <DataProperty IRI="#hasBrand"/>
    <NamedIndividual IRI="#DHT11A"/>
    <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">Itead Studio</Literal>
</DataPropertyAssertion>
<DataPropertyAssertion>
    <DataProperty IRI="#hasHumidityValue"/>
    <NamedIndividual IRI="#DHT11A"/>
    <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#float">1002.0</Literal>
</DataPropertyAssertion>

```

```

<DataPropertyAssertion>
  <DataProperty IRI="#hasHumidityValue"/>
  <NamedIndividual IRI="#DHT11A"/>
  <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#float">31.0</Literal>
</DataPropertyAssertion>
<DataPropertyAssertion>
  <DataProperty IRI="#hasHumidityValue"/>
  <NamedIndividual IRI="#DHT11A"/>
  <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#float">0.0</Literal>
</DataPropertyAssertion>
<DataPropertyAssertion>
  <DataProperty IRI="#hasHumidityValue"/>
  <NamedIndividual IRI="#DHT11A"/>
  <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#float">38.0</Literal>
</DataPropertyAssertion>
<DataPropertyAssertion>
  <DataProperty IRI="#hasHumidityValue"/>
  <NamedIndividual IRI="#DHT11A"/>
  <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#float">30.0</Literal>
</DataPropertyAssertion>
<DataPropertyAssertion>
  <DataProperty IRI="#hasHumidityValue"/>
  <NamedIndividual IRI="#DHT11A"/>
  <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#float">37.0</Literal>
</DataPropertyAssertion>
<DataPropertyAssertion>
  <DataProperty IRI="#hasKind"/>
  <NamedIndividual IRI="#DHT11A"/>
  <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">Analógico</Literal>
</DataPropertyAssertion>
<DataPropertyAssertion>
  <DataProperty IRI="#hasModel"/>
  <NamedIndividual IRI="#DHT11A"/>
  <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">DTH11</Literal>
</DataPropertyAssertion>
<DataPropertyAssertion>
  <DataProperty IRI="#hasNameDevice"/>
  <NamedIndividual IRI="#DHT11A"/>
  <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">Sensor Temperatura
Humedad</Literal>
</DataPropertyAssertion>
<DataPropertyAssertion>
  <DataProperty IRI="#hasNumber"/>
  <NamedIndividual IRI="#DHT11A"/>
  <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#int">1</Literal>
</DataPropertyAssertion>
<DataPropertyAssertion>
  <DataProperty IRI="#hasTempValue"/>
  <NamedIndividual IRI="#DHT11A"/>
  <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#float">25.0</Literal>
</DataPropertyAssertion>
<DataPropertyAssertion>
  <DataProperty IRI="#hasTempValue"/>
  <NamedIndividual IRI="#DHT11A"/>

```

```

    <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#float">0.0</Literal>
</DataPropertyAssertion>
<DataPropertyAssertion>
  <DataProperty IRI="#hasTempValue"/>
  <NamedIndividual IRI="#DHT11A"/>
  <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#float">1001.0</Literal>
</DataPropertyAssertion>
<DataPropertyAssertion>
  <DataProperty IRI="#hasTempValue"/>
  <NamedIndividual IRI="#DHT11A"/>
  <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#float">24.0</Literal>
</DataPropertyAssertion>
<DataPropertyAssertion>
  <DataProperty IRI="#hasTempValue"/>
  <NamedIndividual IRI="#DHT11A"/>
  <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#float">29.0</Literal>
</DataPropertyAssertion>
<DataPropertyAssertion>
  <DataProperty IRI="#hasBrand"/>
  <NamedIndividual IRI="#DHT11B"/>
  <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">Itead Studio</Literal>
</DataPropertyAssertion>
<DataPropertyAssertion>
  <DataProperty IRI="#hasHumidityValue"/>
  <NamedIndividual IRI="#DHT11B"/>
  <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#float">1004.0</Literal>
</DataPropertyAssertion>
<DataPropertyAssertion>
  <DataProperty IRI="#hasHumidityValue"/>
  <NamedIndividual IRI="#DHT11B"/>
  <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#float">30.0</Literal>
</DataPropertyAssertion>
<DataPropertyAssertion>
  <DataProperty IRI="#hasHumidityValue"/>
  <NamedIndividual IRI="#DHT11B"/>
  <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#float">20.0</Literal>
</DataPropertyAssertion>
<DataPropertyAssertion>
  <DataProperty IRI="#hasHumidityValue"/>
  <NamedIndividual IRI="#DHT11B"/>
  <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#float">0.0</Literal>
</DataPropertyAssertion>
<DataPropertyAssertion>
  <DataProperty IRI="#hasKind"/>
  <NamedIndividual IRI="#DHT11B"/>
  <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">Analogico</Literal>
</DataPropertyAssertion>
<DataPropertyAssertion>
  <DataProperty IRI="#hasModel"/>
  <NamedIndividual IRI="#DHT11B"/>
  <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">DHT11</Literal>
</DataPropertyAssertion>
<DataPropertyAssertion>
  <DataProperty IRI="#hasNameDevice"/>

```



```

    <NamedIndividual IRI="#DHT11B"/>
    <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">Sensor Temperatura
Humedad</Literal>
  </DataPropertyAssertion>
  <DataPropertyAssertion>
    <DataProperty IRI="#hasNumber"/>
    <NamedIndividual IRI="#DHT11B"/>
    <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#int">2</Literal>
  </DataPropertyAssertion>
  <DataPropertyAssertion>
    <DataProperty IRI="#hasTempValue"/>
    <NamedIndividual IRI="#DHT11B"/>
    <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#float">0.0</Literal>
  </DataPropertyAssertion>
  <DataPropertyAssertion>
    <DataProperty IRI="#hasTempValue"/>
    <NamedIndividual IRI="#DHT11B"/>
    <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#float">21.0</Literal>
  </DataPropertyAssertion>
  <DataPropertyAssertion>
    <DataProperty IRI="#hasTempValue"/>
    <NamedIndividual IRI="#DHT11B"/>
    <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#float">24.0</Literal>
  </DataPropertyAssertion>
  <DataPropertyAssertion>
    <DataProperty IRI="#hasTempValue"/>
    <NamedIndividual IRI="#DHT11B"/>
    <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#float">1003.0</Literal>
  </DataPropertyAssertion>
  <DataPropertyAssertion>
    <DataProperty IRI="#hasSymbol"/>
    <NamedIndividual IRI="#FahrenheitDegrees"/>
    <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">F</Literal>
  </DataPropertyAssertion>
  <DataPropertyAssertion>
    <DataProperty IRI="#readA"/>
    <NamedIndividual IRI="#FahrenheitDegrees"/>
    <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">fahrenheit</Literal>
  </DataPropertyAssertion>
  <DataPropertyAssertion>
    <DataProperty IRI="#hasBrand"/>
    <NamedIndividual IRI="#Fan1"/>
    <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">Generico</Literal>
  </DataPropertyAssertion>
  <DataPropertyAssertion>
    <DataProperty IRI="#hasIPAddress"/>
    <NamedIndividual IRI="#Fan1"/>
    <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">192.168.2.20</Literal>
  </DataPropertyAssertion>
  <DataPropertyAssertion>
    <DataProperty IRI="#hasModel"/>
    <NamedIndividual IRI="#Fan1"/>
    <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">Generico</Literal>
  </DataPropertyAssertion>

```

```

<DataPropertyAssertion>
  <DataProperty IRI="#hasNameDevice"/>
  <NamedIndividual IRI="#Fan1"/>
  <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">Fan1</Literal>
</DataPropertyAssertion>
<DataPropertyAssertion>
  <DataProperty IRI="#hasState"/>
  <NamedIndividual IRI="#Fan1"/>
  <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#int">0</Literal>
</DataPropertyAssertion>
<DataPropertyAssertion>
  <DataProperty IRI="#hasSymbol"/>
  <NamedIndividual IRI="#Inches"/>
  <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">in</Literal>
</DataPropertyAssertion>
<DataPropertyAssertion>
  <DataProperty IRI="#readA"/>
  <NamedIndividual IRI="#Inches"/>
  <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">pulgadas</Literal>
</DataPropertyAssertion>
<DataPropertyAssertion>
  <DataProperty IRI="#hasBrand"/>
  <NamedIndividual IRI="#LightA"/>
  <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">Itead Studio</Literal>
</DataPropertyAssertion>
<DataPropertyAssertion>
  <DataProperty IRI="#hasKind"/>
  <NamedIndividual IRI="#LightA"/>
  <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">Analógico Digital</Literal>
</DataPropertyAssertion>
<DataPropertyAssertion>
  <DataProperty IRI="#hasLightValue"/>
  <NamedIndividual IRI="#LightA"/>
  <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#float">125.0</Literal>
</DataPropertyAssertion>
<DataPropertyAssertion>
  <DataProperty IRI="#hasLightValue"/>
  <NamedIndividual IRI="#LightA"/>
  <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#float">1005.0</Literal>
</DataPropertyAssertion>
<DataPropertyAssertion>
  <DataProperty IRI="#hasLightValue"/>
  <NamedIndividual IRI="#LightA"/>
  <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#float">0.0</Literal>
</DataPropertyAssertion>
<DataPropertyAssertion>
  <DataProperty IRI="#hasLightValue"/>
  <NamedIndividual IRI="#LightA"/>
  <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#float">546.0</Literal>
</DataPropertyAssertion>
<DataPropertyAssertion>
  <DataProperty IRI="#hasLightValue"/>
  <NamedIndividual IRI="#LightA"/>
  <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#float">124.0</Literal>

```

```

</DataPropertyAssertion>
<DataPropertyAssertion>
  <DataProperty IRI="#hasLightValue"/>
  <NamedIndividual IRI="#LightA"/>
  <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#float">288.0</Literal>
</DataPropertyAssertion>
<DataPropertyAssertion>
  <DataProperty IRI="#hasModel"/>
  <NamedIndividual IRI="#LightA"/>
  <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">Light sensor
brick</Literal>
</DataPropertyAssertion>
<DataPropertyAssertion>
  <DataProperty IRI="#hasNameDevice"/>
  <NamedIndividual IRI="#LightA"/>
  <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">Sensor Intensidad
Luminosa</Literal>
</DataPropertyAssertion>
<DataPropertyAssertion>
  <DataProperty IRI="#hasNumber"/>
  <NamedIndividual IRI="#LightA"/>
  <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#int">0</Literal>
</DataPropertyAssertion>
<DataPropertyAssertion>
  <DataProperty IRI="#hasBrand"/>
  <NamedIndividual IRI="#LightB"/>
  <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">Itead Studio</Literal>
</DataPropertyAssertion>
<DataPropertyAssertion>
  <DataProperty IRI="#hasKind"/>
  <NamedIndividual IRI="#LightB"/>
  <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">Analogico Digital</Literal>
</DataPropertyAssertion>
<DataPropertyAssertion>
  <DataProperty IRI="#hasLightValue"/>
  <NamedIndividual IRI="#LightB"/>
  <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#float">300.0</Literal>
</DataPropertyAssertion>
<DataPropertyAssertion>
  <DataProperty IRI="#hasLightValue"/>
  <NamedIndividual IRI="#LightB"/>
  <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#float">496.0</Literal>
</DataPropertyAssertion>
<DataPropertyAssertion>
  <DataProperty IRI="#hasLightValue"/>
  <NamedIndividual IRI="#LightB"/>
  <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#float">114.0</Literal>
</DataPropertyAssertion>
<DataPropertyAssertion>
  <DataProperty IRI="#hasLightValue"/>
  <NamedIndividual IRI="#LightB"/>
  <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#float">0.0</Literal>
</DataPropertyAssertion>
<DataPropertyAssertion>

```

```

    <DataProperty IRI="#hasLightValue"/>
    <NamedIndividual IRI="#LightB"/>
    <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#float">1006.0</Literal>
</DataPropertyAssertion>
<DataPropertyAssertion>
    <DataProperty IRI="#hasLightValue"/>
    <NamedIndividual IRI="#LightB"/>
    <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#float">130.0</Literal>
</DataPropertyAssertion>
<DataPropertyAssertion>
    <DataProperty IRI="#hasLightValue"/>
    <NamedIndividual IRI="#LightB"/>
    <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#float">117.0</Literal>
</DataPropertyAssertion>
<DataPropertyAssertion>
    <DataProperty IRI="#hasLightValue"/>
    <NamedIndividual IRI="#LightB"/>
    <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#float">115.0</Literal>
</DataPropertyAssertion>
<DataPropertyAssertion>
    <DataProperty IRI="#hasModel"/>
    <NamedIndividual IRI="#LightB"/>
    <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">Light sensor
Brick</Literal>
</DataPropertyAssertion>
<DataPropertyAssertion>
    <DataProperty IRI="#hasNameDevice"/>
    <NamedIndividual IRI="#LightB"/>
    <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">Sensor Intensidad
Luminosa</Literal>
</DataPropertyAssertion>
<DataPropertyAssertion>
    <DataProperty IRI="#hasNumber"/>
    <NamedIndividual IRI="#LightB"/>
    <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#int">2</Literal>
</DataPropertyAssertion>
<DataPropertyAssertion>
    <DataProperty IRI="#hasBrand"/>
    <NamedIndividual IRI="#Lum1"/>
    <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">Philips</Literal>
</DataPropertyAssertion>
<DataPropertyAssertion>
    <DataProperty IRI="#hasIPAddress"/>
    <NamedIndividual IRI="#Lum1"/>
    <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">192.168.2.20</Literal>
</DataPropertyAssertion>
<DataPropertyAssertion>
    <DataProperty IRI="#hasModel"/>
    <NamedIndividual IRI="#Lum1"/>
    <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">Tubular123</Literal>
</DataPropertyAssertion>
<DataPropertyAssertion>
    <DataProperty IRI="#hasNameDevice"/>
    <NamedIndividual IRI="#Lum1"/>

```

```

    <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">Lum1</Literal>
</DataPropertyAssertion>
<DataPropertyAssertion>
  <DataProperty IRI="#hasState"/>
  <NamedIndividual IRI="#Lum1"/>
  <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#int">0</Literal>
</DataPropertyAssertion>
<DataPropertyAssertion>
  <DataProperty IRI="#hasBrand"/>
  <NamedIndividual IRI="#RFIDReader"/>
  <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">Adafruit</Literal>
</DataPropertyAssertion>
<DataPropertyAssertion>
  <DataProperty IRI="#hasKind"/>
  <NamedIndividual IRI="#RFIDReader"/>
  <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">Digital</Literal>
</DataPropertyAssertion>
<DataPropertyAssertion>
  <DataProperty IRI="#hasModel"/>
  <NamedIndividual IRI="#RFIDReader"/>
  <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">PN532</Literal>
</DataPropertyAssertion>
<DataPropertyAssertion>
  <DataProperty IRI="#hasNameDevice"/>
  <NamedIndividual IRI="#RFIDReader"/>
  <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">Lector de
RFID/NFC</Literal>
</DataPropertyAssertion>
<DataPropertyAssertion>
  <DataProperty IRI="#hasNumber"/>
  <NamedIndividual IRI="#RFIDReader"/>
  <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#int">1</Literal>
</DataPropertyAssertion>
<DataPropertyAssertion>
  <DataProperty IRI="#idCardRead"/>
  <NamedIndividual IRI="#RFIDReader"/>
  <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">4A3B5C</Literal>
</DataPropertyAssertion>
<DataPropertyAssertion>
  <DataProperty IRI="#idNumberRead"/>
  <NamedIndividual IRI="#RFIDReader"/>
  <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#int">12345</Literal>
</DataPropertyAssertion>
<DataPropertyAssertion>
  <DataProperty IRI="#hasBrand"/>
  <NamedIndividual IRI="#S5"/>
  <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">Samsung</Literal>
</DataPropertyAssertion>
<DataPropertyAssertion>
  <DataProperty IRI="#hasIMEI"/>
  <NamedIndividual IRI="#S5"/>
  <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">353411069103168</Literal>
</DataPropertyAssertion>
<DataPropertyAssertion>

```

```

    <DataProperty IRI="#hasMacAddress"/>
    <NamedIndividual IRI="#S5"/>
    <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">FC:C2:DE:57:5F:31</Literal>
</DataPropertyAssertion>
<DataPropertyAssertion>
    <DataProperty IRI="#hasModel"/>
    <NamedIndividual IRI="#S5"/>
    <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">S5</Literal>
</DataPropertyAssertion>
<DataPropertyAssertion>
    <DataProperty IRI="#hasNameDevice"/>
    <NamedIndividual IRI="#S5"/>
    <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">Galaxy S5</Literal>
</DataPropertyAssertion>
<DataPropertyAssertion>
    <DataProperty IRI="#hasPhoneNumber"/>
    <NamedIndividual IRI="#S5"/>
    <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">5520895431</Literal>
</DataPropertyAssertion>
<DataPropertyAssertion>
    <DataProperty IRI="#hasBrand"/>
    <NamedIndividual IRI="#SonyX1"/>
    <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">Sony</Literal>
</DataPropertyAssertion>
<DataPropertyAssertion>
    <DataProperty IRI="#hasIMEI"/>
    <NamedIndividual IRI="#SonyX1"/>
    <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">12345610654321</Literal>
</DataPropertyAssertion>
<DataPropertyAssertion>
    <DataProperty IRI="#hasMacAddress"/>
    <NamedIndividual IRI="#SonyX1"/>
    <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">00:01:02:03:04:05</Literal>
</DataPropertyAssertion>
<DataPropertyAssertion>
    <DataProperty IRI="#hasModel"/>
    <NamedIndividual IRI="#SonyX1"/>
    <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">SonyX1</Literal>
</DataPropertyAssertion>
<DataPropertyAssertion>
    <DataProperty IRI="#hasNameDevice"/>
    <NamedIndividual IRI="#SonyX1"/>
    <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">Movil de Romualdo</Literal>
</DataPropertyAssertion>
<DataPropertyAssertion>
    <DataProperty IRI="#hasPhoneNumber"/>
    <NamedIndividual IRI="#SonyX1"/>
    <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">5501234567</Literal>
</DataPropertyAssertion>
<DataPropertyAssertion>
    <DataProperty IRI="#hasBrand"/>
    <NamedIndividual IRI="#Tarjeta00"/>
    <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">Mifare</Literal>
</DataPropertyAssertion>

```

```

<DataPropertyAssertion>
  <DataProperty IRI="#hasIDCard"/>
  <NamedIndividual IRI="#Tarjeta00"/>
  <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">4A3B5C</Literal>
</DataPropertyAssertion>
<DataPropertyAssertion>
  <DataProperty IRI="#hasModel"/>
  <NamedIndividual IRI="#Tarjeta00"/>
  <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">Classic 1K</Literal>
</DataPropertyAssertion>
<DataPropertyAssertion>
  <DataProperty IRI="#readA"/>
  <NamedIndividual IRI="#humidity"/>
  <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">pocentaje</Literal>
</DataPropertyAssertion>
<DataPropertyAssertion>
  <DataProperty IRI="#hasSymbol"/>
  <NamedIndividual IRI="#lum"/>
  <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">Lv</Literal>
</DataPropertyAssertion>
<DataPropertyAssertion>
  <DataProperty IRI="#readA"/>
  <NamedIndividual IRI="#lum"/>
  <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">candela</Literal>
</DataPropertyAssertion>
<AsymmetricObjectProperty>
  <ObjectProperty IRI="#read"/>
</AsymmetricObjectProperty>
<ObjectPropertyDomain>
  <ObjectProperty IRI="#read"/>
  <Class IRI="#Sensor"/>
</ObjectPropertyDomain>
<ObjectPropertyRange>
  <ObjectProperty IRI="#read"/>
  <Class IRI="#PhysicalMeasurement"/>
</ObjectPropertyRange>
<FunctionalDataProperty>
  <DataProperty IRI="#avgHum"/>
</FunctionalDataProperty>
<FunctionalDataProperty>
  <DataProperty IRI="#avgLight"/>
</FunctionalDataProperty>
<FunctionalDataProperty>
  <DataProperty IRI="#avgTemp"/>
</FunctionalDataProperty>
<FunctionalDataProperty>
  <DataProperty IRI="#hasBrand"/>
</FunctionalDataProperty>
<FunctionalDataProperty>
  <DataProperty IRI="#hasHumidityValue"/>
</FunctionalDataProperty>
<FunctionalDataProperty>
  <DataProperty IRI="#hasIDCard"/>
</FunctionalDataProperty>

```

```

<FunctionalDataProperty>
  <DataProperty IRI="#hasIMEI"/>
</FunctionalDataProperty>
<FunctionalDataProperty>
  <DataProperty IRI="#hasIPAddress"/>
</FunctionalDataProperty>
<FunctionalDataProperty>
  <DataProperty IRI="#hasLightValue"/>
</FunctionalDataProperty>
<FunctionalDataProperty>
  <DataProperty IRI="#hasMacAddress"/>
</FunctionalDataProperty>
<FunctionalDataProperty>
  <DataProperty IRI="#hasNameDevice"/>
</FunctionalDataProperty>
<FunctionalDataProperty>
  <DataProperty IRI="#hasNumberOfIdentification"/>
</FunctionalDataProperty>
<FunctionalDataProperty>
  <DataProperty IRI="#hasPhoneNumber"/>
</FunctionalDataProperty>
<FunctionalDataProperty>
  <DataProperty IRI="#hasState"/>
</FunctionalDataProperty>
<FunctionalDataProperty>
  <DataProperty IRI="#hasSymbol"/>
</FunctionalDataProperty>
<FunctionalDataProperty>
  <DataProperty IRI="#hasTempValue"/>
</FunctionalDataProperty>
<FunctionalDataProperty>
  <DataProperty IRI="#idCardRead"/>
</FunctionalDataProperty>
<FunctionalDataProperty>
  <DataProperty IRI="#idNumberRead"/>
</FunctionalDataProperty>
<FunctionalDataProperty>
  <DataProperty IRI="#itIsActive"/>
</FunctionalDataProperty>
<FunctionalDataProperty>
  <DataProperty IRI="#readA"/>
</FunctionalDataProperty>
<DataPropertyDomain>
  <DataProperty IRI="#hasBrand"/>
  <Class IRI="#Device"/>
</DataPropertyDomain>
<DataPropertyDomain>
  <DataProperty IRI="#hasHumidityValue"/>
  <Class IRI="#Sensor"/>
</DataPropertyDomain>
<DataPropertyDomain>
  <DataProperty IRI="#hasIDCard"/>
  <Class IRI="#RFIDCard"/>
</DataPropertyDomain>

```



```

<DataPropertyDomain>
  <DataProperty IRI="#hasIMEI"/>
  <Class IRI="#SmartPhone"/>
</DataPropertyDomain>
<DataPropertyDomain>
  <DataProperty IRI="#hasIPAddress"/>
  <Class IRI="#Actuator"/>
</DataPropertyDomain>
<DataPropertyDomain>
  <DataProperty IRI="#hasKind"/>
  <Class IRI="#Device"/>
</DataPropertyDomain>
<DataPropertyDomain>
  <DataProperty IRI="#hasLightValue"/>
  <Class IRI="#Sensor"/>
</DataPropertyDomain>
<DataPropertyDomain>
  <DataProperty IRI="#hasMacAddress"/>
  <Class IRI="#Device"/>
</DataPropertyDomain>
<DataPropertyDomain>
  <DataProperty IRI="#hasModel"/>
  <Class IRI="#Device"/>
</DataPropertyDomain>
<DataPropertyDomain>
  <DataProperty IRI="#hasNameDevice"/>
  <Class IRI="#Device"/>
</DataPropertyDomain>
<DataPropertyDomain>
  <DataProperty IRI="#hasNumber"/>
  <Class IRI="#Device"/>
</DataPropertyDomain>
<DataPropertyDomain>
  <DataProperty IRI="#hasPhoneNumber"/>
  <Class IRI="#SmartPhone"/>
</DataPropertyDomain>
<DataPropertyDomain>
  <DataProperty IRI="#hasState"/>
  <Class IRI="#Actuator"/>
</DataPropertyDomain>
<DataPropertyDomain>
  <DataProperty IRI="#hasSymbol"/>
  <Class IRI="#PhysicalMeasurement"/>
</DataPropertyDomain>
<DataPropertyDomain>
  <DataProperty IRI="#hasTempValue"/>
  <Class IRI="#Sensor"/>
</DataPropertyDomain>
<DataPropertyDomain>
  <DataProperty IRI="#idCardRead"/>
  <Class IRI="#Sensor"/>
</DataPropertyDomain>
<DataPropertyDomain>
  <DataProperty IRI="#idNumberRead"/>

```

```

    <Class IRI="#Sensor"/>
</DataPropertyDomain>
<DataPropertyDomain>
    <DataProperty IRI="#itIsActive"/>
    <Class IRI="#Actuator"/>
</DataPropertyDomain>
<DataPropertyDomain>
    <DataProperty IRI="#readA"/>
    <Class IRI="#PhysicalMeasurement"/>
</DataPropertyDomain>
<DataPropertyRange>
    <DataProperty IRI="#avgHum"/>
    <Datatype abbreviatedIRI="xsd:float"/>
</DataPropertyRange>
<DataPropertyRange>
    <DataProperty IRI="#avgLight"/>
    <Datatype abbreviatedIRI="xsd:float"/>
</DataPropertyRange>
<DataPropertyRange>
    <DataProperty IRI="#avgTemp"/>
    <Datatype abbreviatedIRI="xsd:float"/>
</DataPropertyRange>
<DataPropertyRange>
    <DataProperty IRI="#hasBrand"/>
    <Datatype abbreviatedIRI="xsd:string"/>
</DataPropertyRange>
<DataPropertyRange>
    <DataProperty IRI="#hasHumidityValue"/>
    <Datatype abbreviatedIRI="xsd:float"/>
</DataPropertyRange>
<DataPropertyRange>
    <DataProperty IRI="#hasIDCard"/>
    <Datatype abbreviatedIRI="xsd:string"/>
</DataPropertyRange>
<DataPropertyRange>
    <DataProperty IRI="#hasIMEI"/>
    <Datatype abbreviatedIRI="xsd:string"/>
</DataPropertyRange>
<DataPropertyRange>
    <DataProperty IRI="#hasIPAddress"/>
    <Datatype abbreviatedIRI="xsd:string"/>
</DataPropertyRange>
<DataPropertyRange>
    <DataProperty IRI="#hasKind"/>
    <Datatype abbreviatedIRI="xsd:string"/>
</DataPropertyRange>
<DataPropertyRange>
    <DataProperty IRI="#hasLightValue"/>
    <Datatype abbreviatedIRI="xsd:float"/>
</DataPropertyRange>
<DataPropertyRange>
    <DataProperty IRI="#hasMacAddress"/>
    <Datatype abbreviatedIRI="xsd:string"/>
</DataPropertyRange>

```

```

<DataPropertyRange>
  <DataProperty IRI="#hasModel"/>
  <Datatype abbreviatedIRI="xsd:string"/>
</DataPropertyRange>
<DataPropertyRange>
  <DataProperty IRI="#hasNameDevice"/>
  <Datatype abbreviatedIRI="xsd:string"/>
</DataPropertyRange>
<DataPropertyRange>
  <DataProperty IRI="#hasNumber"/>
  <Datatype abbreviatedIRI="xsd:int"/>
</DataPropertyRange>
<DataPropertyRange>
  <DataProperty IRI="#hasNumberOfIdentification"/>
  <Datatype abbreviatedIRI="xsd:int"/>
</DataPropertyRange>
<DataPropertyRange>
  <DataProperty IRI="#hasPhoneNumber"/>
  <Datatype abbreviatedIRI="xsd:string"/>
</DataPropertyRange>
<DataPropertyRange>
  <DataProperty IRI="#hasState"/>
  <Datatype abbreviatedIRI="xsd:int"/>
</DataPropertyRange>
<DataPropertyRange>
  <DataProperty IRI="#hasSymbol"/>
  <Datatype abbreviatedIRI="xsd:string"/>
</DataPropertyRange>
<DataPropertyRange>
  <DataProperty IRI="#hasTempValue"/>
  <Datatype abbreviatedIRI="xsd:float"/>
</DataPropertyRange>
<DataPropertyRange>
  <DataProperty IRI="#idCardRead"/>
  <Datatype abbreviatedIRI="xsd:string"/>
</DataPropertyRange>
<DataPropertyRange>
  <DataProperty IRI="#idNumberRead"/>
  <Datatype abbreviatedIRI="xsd:int"/>
</DataPropertyRange>
<DataPropertyRange>
  <DataProperty IRI="#itIsActive"/>
  <Datatype abbreviatedIRI="xsd:boolean"/>
</DataPropertyRange>
<DataPropertyRange>
  <DataProperty IRI="#readA"/>
  <Datatype abbreviatedIRI="xsd:string"/>
</DataPropertyRange>
<AnnotationAssertion>
  <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
  <IRI#Actuator</IRI>
  <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">Los individuos que
aqu&#237; clasifican son aquellos que pueden llevar a cabo acciones en el mundo fisico, por
ejemplo:

```

relevadores, solenoides, switches, lamparas, aire acondicionado, ventiladores, etc.

Individuals classified here are those who can take action in the physical world, for example: relays, solenoids, switches, lamps, air conditioning, fans, etc.</Literal>

</AnnotationAssertion>

<AnnotationAssertion>

<AnnotationProperty abbreviatedIRI="rdfs:comment"/>

<IRI>#CelsiusDegrees</IRI>

<Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">Grados celsius

Celsius degrees</Literal>

</AnnotationAssertion>

<AnnotationAssertion>

<AnnotationProperty abbreviatedIRI="rdfs:comment"/>

<IRI>#Centimeters</IRI>

<Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">unidad de medida centimetros

measure unit centimeters</Literal>

</AnnotationAssertion>

<AnnotationAssertion>

<AnnotationProperty abbreviatedIRI="rdfs:comment"/>

<IRI>#DHT11A</IRI>

<Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">Sensor combinado que mide temperatura y humedad

combined sensor that measures temperature and humidity</Literal>

</AnnotationAssertion>

<AnnotationAssertion>

<AnnotationProperty abbreviatedIRI="rdfs:comment"/>

<IRI>#Device</IRI>

<Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">Esta clase contiene a las medidas o magnitudes físicas las cuales pueden ser medida a través de sensores, las medidas son:

distancia, humedad, intensidad luminosa y temperatura.

This class contains measures or physical quantities which can be measured through sensors, the measures are:

distance, humidity, light intensity and temperature</Literal>

</AnnotationAssertion>

<AnnotationAssertion>

<AnnotationProperty abbreviatedIRI="rdfs:comment"/>

<IRI>#Device</IRI>

<Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">Esta clase contiene a los dispositivos con los que se puede llevar a cabo una interacción, por lo tanto contempla a:

Sensores los cuales proveeran información,

Actuadores los cuales llevaran a cabo las tareas de encender o apagar a algún dispositivo eléctrico.

SmartPhone el cual tiene como tarea el interactuar con el usuario este pertenece a un usuario.

Tarjeta RFID esta se le otorga a un usuario ya sea visitante, Alumno o trabajador y contiene un ID unico y un numero asignado para el usuario por ejemplo: Matricula o numero economico

This class contains devices that can perform an interaction,  
therefore contemplates:

Sensors which provide information,

Actuators which carry out tasks to turn on or off any electrical device.

SmartPhone which has the task to interact with the user that belongs to a user.

RFID card that is given to a user either visitor, student or worker and contains a unique ID and a number assigned to the user for example: Serial or economic number</Literal>

</AnnotationAssertion>

<AnnotationAssertion>

<AnnotationProperty abbreviatedIRI="rdfs:comment"/>

<IRI>#Distance</IRI>

<Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">Los individuos que aqu&#237; clasifican son los que permiten medir distancias por ejemplo: centimetros y pulgadas.

Individuals who are classified here to measure distances eg centimeters and inches</Literal>

</AnnotationAssertion>

<AnnotationAssertion>

<AnnotationProperty abbreviatedIRI="rdfs:comment"/>

<IRI>#FahrenheitDegrees</IRI>

<Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">Grados Farenheit

Farenheit degrees</Literal>

</AnnotationAssertion>

<AnnotationAssertion>

<AnnotationProperty abbreviatedIRI="rdfs:comment"/>

<IRI>#Humidity</IRI>

<Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">Los individuos que aqu&#237; clasifican son los que permiten medir la humedad del ambiente en el que se encuentran.

Individuals who are classified here to measure the humidity in found.</Literal>

</AnnotationAssertion>

<AnnotationAssertion>

<AnnotationProperty abbreviatedIRI="rdfs:comment"/>

<IRI>#Inches</IRI>

<Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">unidad de medida pulgadas

measure unit inches</Literal>

</AnnotationAssertion>

<AnnotationAssertion>

<AnnotationProperty abbreviatedIRI="rdfs:comment"/>

<IRI>#LightIntensity</IRI>

<Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">Los individuos que aqu&#237; clasifican son los que permiten medir la intensidad luminosa del ambiente.

Individuals who are classified here to measure ambient light intensity.</Literal>  
</AnnotationAssertion>  
<AnnotationAssertion>  
 <AnnotationProperty abbreviatedIRI="rdfs:comment"/>  
 <IRI>#Lum1</IRI>  
 <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">Lampara o luminaria de  
techo

Lamp fixed in a roof</Literal>  
</AnnotationAssertion>  
<AnnotationAssertion>  
 <AnnotationProperty abbreviatedIRI="rdfs:comment"/>  
 <IRI>#PhysicalMeasurement</IRI>  
 <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">This class contains  
measures or physical quantities which can be measured through sensors, the measures are:  
  
distance, humidity, light intensity and temperature.

Esta clase contiene a las medidas o magnitudes físicas las cuales pueden ser medida a través de sensores, las medidas son:

distancia, humedad, intensidad luminosa y temperatura</Literal>  
</AnnotationAssertion>  
<AnnotationAssertion>  
 <AnnotationProperty abbreviatedIRI="rdfs:comment"/>  
 <IRI>#RFIDCard</IRI>  
 <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">Los individuos que  
aquí clasifican son las tarjetas de RFID del tipo Mifare Classic las cuales pueden contener  
datos del usuario al que son asignadas, además poseen un número de identificación único.

Individuals are classified here cards Mifare Classic RFID type which may contain user data to  
which they are assigned, also have a unique number of the identification</Literal>  
</AnnotationAssertion>  
<AnnotationAssertion>  
 <AnnotationProperty abbreviatedIRI="rdfs:comment"/>  
 <IRI>#Sensor</IRI>  
 <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">Los individuos que  
aquí clasifican son los sensores los cuales nos permiten obtener información del medio  
ambiente en el que se encuentran.

Individuals classified here are sensors which allow us to obtain information from the environment  
in which they are</Literal>  
</AnnotationAssertion>  
<AnnotationAssertion>  
 <AnnotationProperty abbreviatedIRI="rdfs:comment"/>  
 <IRI>#SmartPhone</IRI>  
 <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">Los individuos que  
aquí clasifican son los teléfonos inteligentes los cuales son cada vez más aptos para  
compartir información con otros sistemas, además pertenecen solo a un usuario.

Individuals classified here are smart phones which are increasingly able to share information with  
other systems, also belong only to a user.</Literal>  
</AnnotationAssertion>

```
<AnnotationAssertion>
  <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
  <IRI>#SonyX1</IRI>
  <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">SmartPhone
```

telefono inteligente</Literal>

```
</AnnotationAssertion>
```

```
<AnnotationAssertion>
```

```
  <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
```

```
  <IRI>#Tarjeta00</IRI>
```

```
  <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">RFID card
```

tarjeta RFID</Literal>

```
</AnnotationAssertion>
```

```
<AnnotationAssertion>
```

```
  <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
```

```
  <IRI>#Temperature</IRI>
```

```
  <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">Los individuos que
aqu&#237; clasifican son los que permiten medir la temperatura del ambiente en el que se
encuentran.
```

Individuals who are classified here to measure the temperature of the environment in which they find themselves.</Literal>

```
</AnnotationAssertion>
```

```
<AnnotationAssertion>
```

```
  <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
```

```
  <IRI>#avgHum</IRI>
```

```
  <Literal datatypeIRI="http://www.w3.org/1999/02/22-rdf-syntax-ns#PlainLiteral">It refers
to the humidity average that has been calculated from the read a group of sensors installed in a
physical space.
```

Se refiere a la humedad promedio que ha sido calculada a partir de la lectura de un grupo de sensores instalados en un espacio f&#237;sico.</Literal>

```
</AnnotationAssertion>
```

```
<AnnotationAssertion>
```

```
  <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
```

```
  <IRI>#avgLight</IRI>
```

```
  <Literal datatypeIRI="http://www.w3.org/1999/02/22-rdf-syntax-ns#PlainLiteral">It refers
to the light intensity average that has been calculated from the read a group of sensors
installed in a physical space.
```

Se refiere al promedio de la intensidad luminosa que ha sido calculada a partir de la lectura de un grupo de sensores instalados en un espacio f&#237;sico.</Literal>

```
</AnnotationAssertion>
```

```
<AnnotationAssertion>
```

```
  <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
```

```
  <IRI>#avgTemp</IRI>
```

```
  <Literal datatypeIRI="http://www.w3.org/1999/02/22-rdf-syntax-ns#PlainLiteral">It refers
to the average temperature that has been calculated from the read a group of sensors installed in
a physical space.
```

Se refiere a la temperatura promedio que ha sido calculada a partir de la lectura de un grupo de sensores instalados en un espacio f&#237;sico.</Literal>

```
</AnnotationAssertion>
```

```
<AnnotationAssertion>
  <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
  <IRI##hasBrand</IRI>
  <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">muestra la marca de fabrica
del dipostivo
```

show the brand of a device</Literal>

```
</AnnotationAssertion>
<AnnotationAssertion>
  <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
  <IRI##hasHumidityValue</IRI>
  <Literal datatypeIRI="http://www.w3.org/1999/02/22-rdf-syntax-ns##PlainLiteral">It refers
to the humidity value that has been read from a sensor installed in a physical space.
```

Se refiere a el valor de humedad que se ha leído a partir de un sensor instalado en un espacio físico.</Literal>

```
</AnnotationAssertion>
<AnnotationAssertion>
  <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
  <IRI##hasIDCard</IRI>
  <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">es el ID unico de una
tarjeta de RFID
```

It's the ID unique of a RFID card</Literal>

```
</AnnotationAssertion>
<AnnotationAssertion>
  <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
  <IRI##hasIMEI</IRI>
  <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">Es el numero de IMEI de 15
digitos de un telefono movil
```

It's the number IMEI of 15 digits of a mobile phone</Literal>

```
</AnnotationAssertion>
<AnnotationAssertion>
  <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
  <IRI##hasIPAddress</IRI>
  <Literal datatypeIRI="http://www.w3.org/1999/02/22-rdf-syntax-ns##PlainLiteral">It refers
to the IP address that a actuator has in the wireless network.
```

Se refiere a la dirección IP que un actuador tiene en la red inalámbrica.</Literal>

```
</AnnotationAssertion>
<AnnotationAssertion>
  <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
  <IRI##hasKind</IRI>
  <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">tipo de dispositivo,
inalambrico, alambrico, fijo, etc.
```

It's the kind of a device, wireless, wired, fixed, etc.</Literal>

```
</AnnotationAssertion>
<AnnotationAssertion>
  <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
  <IRI##hasLightValue</IRI>
  <Literal datatypeIRI="http://www.w3.org/1999/02/22-rdf-syntax-ns##PlainLiteral">It refers
to the light intensity value that has been read from a sensor installed in a physical space.
```



Se refiere al valor de intensidad de luz que ha sido leido desde un sensor instalado en un espacio fisico.

```
</Literal>
</AnnotationAssertion>
<AnnotationAssertion>
  <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
  <IRI##hasMacAddress</IRI>
  <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">indica la direccion MAC de un dispositivo de red
```

show the MAC address of a device in the net

```
</Literal>
</AnnotationAssertion>
<AnnotationAssertion>
  <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
  <IRI##hasModel</IRI>
  <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">modelo del dispositivo
```

the model of a device

```
</Literal>
</AnnotationAssertion>
<AnnotationAssertion>
  <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
  <IRI##hasNameDevice</IRI>
  <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">nombre que se le ha asignado a un dispositivo
```

the name that has been assigned to a device

```
</Literal>
</AnnotationAssertion>
<AnnotationAssertion>
  <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
  <IRI##hasNumber</IRI>
  <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">numero de dispositivo por ejemplo
si tengo 2 ventiladores uno seria el 1y el otro el 2
```

the number of a device, e.g, if i have 2 fans, one must be the 1 and the other must be the 2

```
</Literal>
</AnnotationAssertion>
<AnnotationAssertion>
  <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
  <IRI##hasNumberOfIdentification</IRI>
  <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">numero de identificacion que se asigna a una persona
este numero puede ser Matricula en caso de los alumnos
numero economico en caso de los trabajadores
y un numero de 3 digitos en caso de visitantes
```

the number of identification that has been assigned to a person, this number could be a enrollment in the case of a student, economic number in the case of a worker and a number of 3 digits in the case of a visitor

```
</Literal>
</AnnotationAssertion>
<AnnotationAssertion>
  <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
  <IRI##hasOwner</IRI>
```

<Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">indica quien es el propietario de un dispositivo por ejemplo un telefono celular tiene un solo propietario

show who is the proprietary of a device, e.g, a cell phone has only one proprietary</Literal>

</AnnotationAssertion>

<AnnotationAssertion>

<AnnotationProperty abbreviatedIRI="rdfs:comment"/>

<IRI>#hasPhoneNumber</IRI>

<Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">numero de telefono del dispositivo movil

number of telephone of the mobile device</Literal>

</AnnotationAssertion>

<AnnotationAssertion>

<AnnotationProperty abbreviatedIRI="rdfs:comment"/>

<IRI>#hasState</IRI>

<Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">indica el estado de un dispositivo por ejemplo

0 = apagado

1 = encendido

show the state of a device, e.g,

0 = off

1 = on</Literal>

</AnnotationAssertion>

<AnnotationAssertion>

<AnnotationProperty abbreviatedIRI="rdfs:comment"/>

<IRI>#hasSymbol</IRI>

<Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">simbolo con el que se identifica a una medida o magnitud fisica

symbol that identifies a measure or physical quantity</Literal>

</AnnotationAssertion>

<AnnotationAssertion>

<AnnotationProperty abbreviatedIRI="rdfs:comment"/>

<IRI>#hasTempValue</IRI>

<Literal datatypeIRI="http://www.w3.org/1999/02/22-rdf-syntax-ns#PlainLiteral">It refers to the temperature value that has been read from a sensor installed in a physical space.

Se refiere al valor de la temperatura que se ha leído desde un sensor instalado en un espacio físico.</Literal>

</AnnotationAssertion>

<AnnotationAssertion>

<AnnotationProperty abbreviatedIRI="rdfs:comment"/>

<IRI>#humidity</IRI>

<Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">humidity

humedad</Literal>

</AnnotationAssertion>

<AnnotationAssertion>

<AnnotationProperty abbreviatedIRI="rdfs:comment"/>

<IRI>#idCardRead</IRI>

<Literal datatypeIRI="http://www.w3.org/1999/02/22-rdf-syntax-ns#PlainLiteral">Refers to the number that has been read from the RFID card reader.

Se refiere al número que se ha leído desde el lector de tarjetas RFID.</Literal>

</AnnotationAssertion>

<AnnotationAssertion>

<AnnotationProperty abbreviatedIRI="rdfs:comment"/>

<IRI>#idNumberRead</IRI>

<Literal datatypeIRI="http://www.w3.org/1999/02/22-rdf-syntax-ns#PlainLiteral">Refers to the id number that has been read from the RFID card reader.

Se refiere al número de identificación que se ha leído desde el lector de tarjetas RFID.</Literal>

</AnnotationAssertion>

<AnnotationAssertion>

<AnnotationProperty abbreviatedIRI="rdfs:comment"/>

<IRI>#itIsActive</IRI>

<Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">Indica si un dispositivo esta activo o no

show if a device it's active or not</Literal>

</AnnotationAssertion>

<AnnotationAssertion>

<AnnotationProperty abbreviatedIRI="rdfs:comment"/>

<IRI>#lum</IRI>

<Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">light unit

unidad de luminosidad</Literal>

</AnnotationAssertion>

<AnnotationAssertion>

<AnnotationProperty abbreviatedIRI="rdfs:comment"/>

<IRI>#read</IRI>

<Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">Hace referencia a la lectura de una medida fisica

Make a reference to a read of a physical measure</Literal>

</AnnotationAssertion>

</Ontology>

<!-- Generated by the OWL API (version 3.0.0.1413) http://owlapi.sourceforge.net -->

## 11. REFERENCIAS BIBLIOGRÁFICAS

- [1] M. Henríquez and P. Palma, "Control Automático de Condiciones Ambientales en Domótica usando Redes Neuronales Artificiales", *Información tecnológica*, vol. 22, no. 3, pp. 125-139, 2011.
- [2] F. Jordán Teruel, "Estudio de la plataforma Android para dispositivos móviles y desarrollo de aplicación para la administración de redes de sensores inalámbricos", *Hdl.handle.net*, 2010. [Online]. Available: <http://hdl.handle.net/10016/10708>. [Accessed: 19- Feb- 2016].
- [3] J. Castro Heredia, "Uso del protocolo CoAP para la implementación de una aplicación Domótica con redes de sensores inalámbricas", *Hdl.handle.net*, 2014. [Online]. Available: <http://hdl.handle.net/10317/4163>. [Accessed: 19- Feb- 2016].
- [4] A. Hernández Hernández, "Red de monitoreo remoto inalámbrico de sensores con dispositivos ZIGBEE", Proyecto Tecnológico, División de Ciencias Básicas e Ingeniería, Universidad Autónoma Metropolitana Azcapotzalco, México, 2014.
- [5] A. A. Nieto Nava, "Monitoreo Remoto de Temperatura Usando Sensores 1-wire", Proyecto Tecnológico, División de Ciencias Básicas e Ingeniería, Universidad Autónoma Metropolitana Azcapotzalco, México, 2014.
- [6] H. Juarez Alvarez, "Monitoreo Remoto de Temperatura y Humedad Utilizando Tecnología Zigbee", Proyecto Tecnológico, División de Ciencias Básicas e Ingeniería, Universidad Autónoma Azcapotzalco, México, 2014.
- [7] O. Torrente Artero, *Arduino*, 1st ed. San Fernando de Henares, Madrid: RC Libros, 2013.
- [8] J. Bravo and X. Alamán Roldán, *Actas del I Simposio sobre Computación Ubicua e Inteligencia Ambiental*, 1st ed. Madrid: Thomson-Paraninfo, 2005, pp. 75-86.
- [9] "Arduino - Home", *Arduino.cc*, 2016. [Online]. Available: <http://arduino.cc>. [Accessed: 12- Dec- 2016].
- [10] "Processing.org", *Processing.org*, 2016. [Online]. Available: <https://processing.org/>. [Accessed: 12- Dec- 2016].
- [11] C. Date, *An introduction to database systems*, 1st ed. Reading, Mass.: Addison-Wesley Pub. Co., 1981.
- [12] J.A. Reyes Ortiz., *2Creación Automática de Ontologías a partir de textos*, tesis de doctorado, Centro Nacional de Investigación y Desarrollo Tecnológico, Morelos, 2013.
- [13] T. Gruber. *Toward principles for the design of ontologies used for knowledge sharing*. In *Forna Analysis in Conceptual Analysis and Knowledge Representation*. Kluwer, 1993.

- [14] H. Weigand Multilingual ontology-based lexicón for news\_Itoring. In K. Mahesh, editor, The TREVE Project, pages 138-159,1997
- [15] Perramon Tornil, *Programacion de sockets*, 1st ed. Barcelona: Universitat Oberta de Catalunya, 2000.
- [16] J. Hedley, "jsoup Java HTML Parser, with best of DOM, CSS, and jquery", *Jsoup.org*, 2016. [Online]. Available: <http://jsoup.org>. [Accessed: 12- Dec- 2016].
- [17] "OWL API", *Owlapi.sourceforge.net*, 2016. [Online]. Available: <http://owlapi.sourceforge.net/>. [Accessed: 12- Dec- 2016].
- [18] Arduino Playground - DHT11Lib", *Playground.arduino.cc*, 2016. [Online]. Available: <http://playground.arduino.cc/Main/DHT11Lib>. [Accessed: 13- Dec- 2016].
- [19] Light intensity sensor, IM120710017 Datasheet, available online: [ftp://imall.iteadstudio.com/Electronic\\_Brick/IM120710017/DS\\_IM120710017.pdf](ftp://imall.iteadstudio.com/Electronic_Brick/IM120710017/DS_IM120710017.pdf)
- [20] "Arduino - ArduinoBoardUno", *Arduino.cc*, 2016. [Online]. Available: <https://www.arduino.cc/en/Main/ArduinoBoardUno>. [Accessed: 13- Dec- 2016].
- [21] "Arduino - ArduinoWiFiShield101", *Arduino.cc*, 2016. [Online]. Available: <https://www.arduino.cc/en/Main/ArduinoWiFiShield101>. [Accessed: 13- Dec- 2016].