

	<b>INFOTEC Centro de Investigación e Innovación en Tecnologías de la Información y Comunicación</b>	<b>Hoja</b>	1 DE 9
		<b>Versión</b>	1.0
		<b>Fecha de elaboración</b>	15/Diciembre/2015
		<b>Fecha de actualización</b>	15/Enero/2016

**Manual Integración Cloudino Connector FIWARE IoT**

## Connecting Cloudino Connector to FIWARE IoT

### 1. What is FIWARE IoT

FIWARE is an open software ecosystem provided by the FIWARE Community (<http://www.fiware.org>).

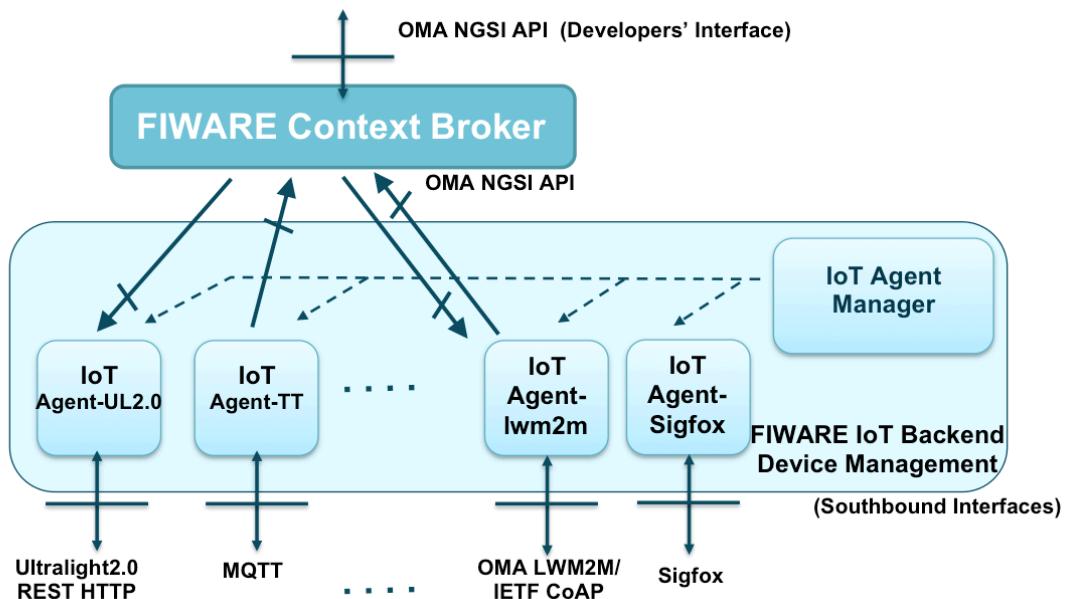
FIWARE exposes to developers Data Context elements or entities (JSON objects) with attributes and metadata with a uniform REST API (NGSI9/10). NGSI is now being adopted by more than 60 smartcities worldwide:

- <http://connectedsmartcities.eu/open-agile-smart-cities>

Orion Context Broker is available at: <http://bit.ly/github-ContextBroker>

FIWARE IoT is an opensource software stack aiming to bring Data-level interoperability to the complex salad of standards and protocols in the world of IoT today.

FIWARE IoT is able to expose -by means of the Orion Context Broker component- all IoT devices information and commands using the Data Context API (NGSI).



	<b>INFOTEC Centro de Investigación e Innovación en Tecnologías de la Información y Comunicación</b>	<b>Hoja</b>	2 DE 9
		<b>Versión</b>	1.0
		<b>Fecha de elaboración</b>	15/Diciembre/2015
		<b>Fecha de actualización</b>	15/Enero/2016

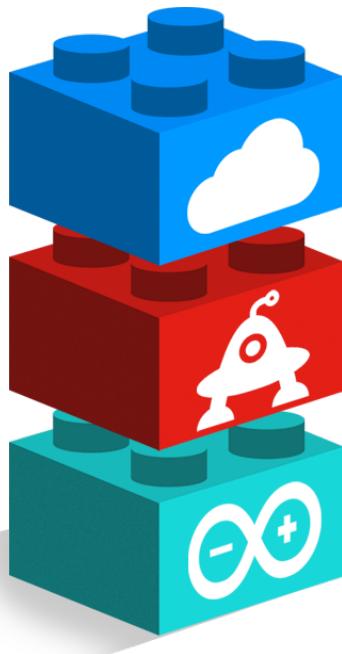
**Manual Integración Cloudino Connector FIWARE IoT**

Any IoT standard or proprietary protocol can be connected to FIWARE via the IoT-Agent components. Currently FIWARE IoT is providing IoT-Agents for:

- Ultralight2.0/HTTP: [http://bit.ly/fiware\\_iot-ul20](http://bit.ly/fiware_iot-ul20)
- MQTT/TCP: [http://bit.ly/fiware\\_iot-ul20](http://bit.ly/fiware_iot-ul20)
- LWM2M/CoAP: [http://bit.ly/fiware\\_iot-lwm2m-coap](http://bit.ly/fiware_iot-lwm2m-coap)
- SIGFox Cloud: <https://github.com/telefonicaid/sigfox-iotagent>

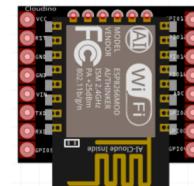
## 2. What is Cloudino Connector

**Cloudino Connector** is a WiFi device that lets you connect to the Internet different microcontrollers (Atmel AVR, PIC, Intel Edison, etc.) in a simple and transparent way.



**Cloud Service**

**Cloudino Connector**



**Microcontroller (Arduino)**

The **Cloudino Connector** is not like an Arduino shield, is other independent processor working in parallel dedicated only to the network layer including the IoT protocols, leaving the Arduino or microcontroller dedicated only to process data from sensors and control actuators, while allows reprogramming the Arduino or microcontroller via WiFi or Cloud. Leaving the configuration of protocols to the Cloudino Web Interface and out off your Arduino code.

	<b>INFOTEC Centro de Investigación e Innovación en Tecnologías de la Información y Comunicación</b>	<b>Hoja</b>	3 DE 9
		<b>Versión</b>	1.0
		<b>Fecha de elaboración</b>	15/Diciembre/2015
		<b>Fecha de actualización</b>	15/Enero/2016

**Manual Integración Cloudino Connector FIWARE IoT**

### 3. How to use Cloudino Connector with FIWARE

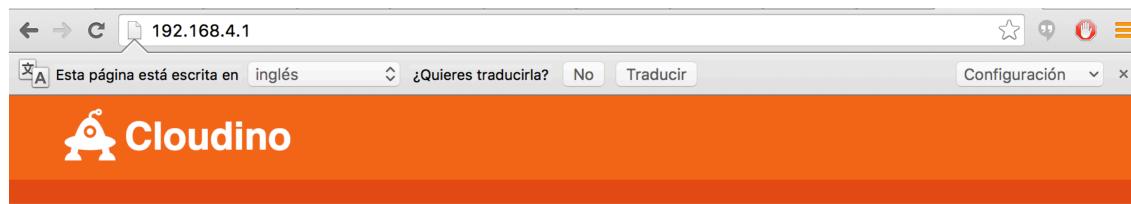
**Cloudino Connector** can be integrated with **FIWARE** above-described **FIWARE IoT** ecosystems using different mechanisms:

- Direct Connection
- Connection via MQTT IoT-Agent
- Connection via Cloudino.io cloud service (work in process)

#### 3.1. Direct Connection to FIWARE Context Broker

**Cloudino Connector** can connect to the FIWARE Context Broker without an IoT-Agent, using the simple Cloudino **Configuration Web Interface**.

The **Cloudino Connector** starts an access point that lets you connect to the configuration web interface at: <http://192.168.4.1>



#### Welcome to Cloudino Platform

##### An easy way to connect your Arduino to the Cloud

Cloudino is an Internet of Things Platform (Cloud Connector) that allows you to fully connect to the cloud, different MCUs platforms (Atmel AVR, Microchip PIC, etc.) in a very easy and transparent way.

Cloudino is developed and maintained by INFOTEC (Public Research Center of CONACYT) as an Open Source and Open Hardware Platform

##### Configuration

- Cloudino Status
- Access Point Configuration
- Wifi Configuration
- mDNS Configuration
- JavaScript Configuration
- Server Configuration

[cloudino.io](http://cloudino.io)    [infotec.mx](http://infotec.mx)

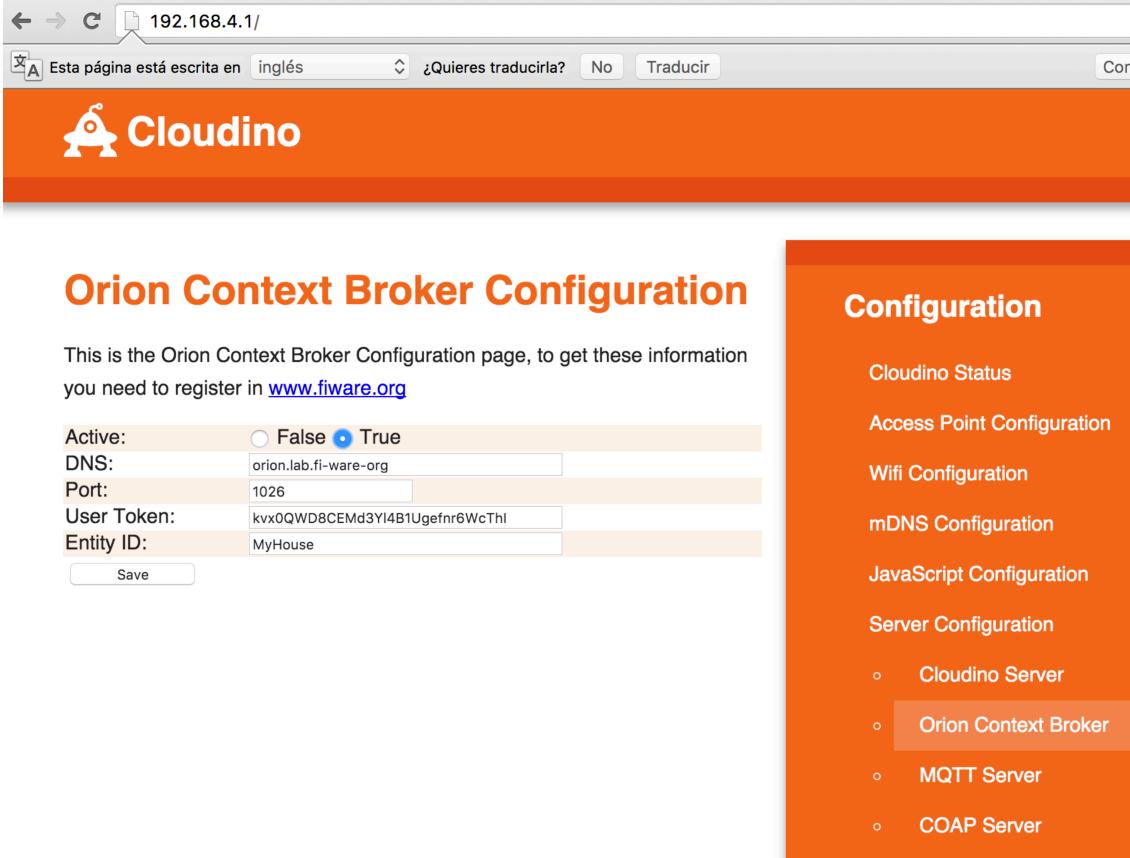
Cloudino © 2015 INFOTEC

To use a direct connection to FIWARE Context Broker you have to select the FIWARE Orion Context Broker in Server Configuration and setting the next fields:



	<b>INFOTEC Centro de Investigación e Innovación en Tecnologías de la Información y Comunicación</b>	<b>Hoja</b>	4 DE 9
		<b>Versión</b>	1.0
		<b>Fecha de elaboración</b>	15/Diciembre/2015
		<b>Fecha de actualización</b>	15/Enero/2016

**Manual Integración Cloudino Connector FIWARE IoT**



The screenshot shows a web browser window with the URL `192.168.4.1/`. The page title is "Cloudino". On the left, there's a configuration form for "Orion Context Broker Configuration" with fields: Active (radio button selected for True), DNS (orion.lab.fi-ware.org), Port (1026), User Token (kvx0QWD8CEMd3Yl4B1Ugefnr6WcThI), and Entity ID (MyHouse). A "Save" button is at the bottom. On the right, a sidebar titled "Configuration" lists various services: Cloudino Status, Access Point Configuration, Wifi Configuration, mDNS Configuration, JavaScript Configuration, Server Configuration (with Orion Context Broker selected), MQTT Server, and COAP Server.

**Active:** *True*

**DNS:** *Orion.lab.fi-ware.org*

**Port:** *1026*

**User Token:** (*generate token from FIWARE Server, for example kvx0QWD8CEMd3Yl4B1Ugefnr6WcThI. For more information go to <https://forge.fiware.org/plugins/mediawiki/wiki/fiware/index.php/Publish/Subs>cribe\_Broker\_-\_Orion\_Context\_Broker\_-\_Quick\_Start\_for\_Programmers*

**Entity ID:** (*Instance name, for example "MyHouse"*)

	<b>INFOTEC Centro de Investigación e Innovación en Tecnologías de la Información y Comunicación</b>	<b>Hoja</b>	5 DE 9
		<b>Versión</b>	1.0
		<b>Fecha de elaboración</b>	15/Diciembre/2015
		<b>Fecha de actualización</b>	15/Enero/2016

**Manual Integración Cloudino Connector FIWARE IoT**

## Example of Arduino Code to Post Temperature and Humidity

```
#include <Cloudino.h>
#include <dht11.h>

#define DHT11PIN 8

Cloudino cdino;           //Cloudino Library
dht11 DHT11;              //DHT11 Library

void getSensor()
{
    int chk = DHT11.read(DHT11PIN);
    cdino.post("temperature",String((float)DHT11.temperature,2));
    cdino.post("humidity",String((float)DHT11.humidity,2));
    cdino.print("Timer done!"); //Send to console
}

void setup()
{
    cdino.setInterval(10000,getSensor); //Timer every 10 seconds
    cdino.begin();
}

void loop()
{
    cdino.loop();
}
```

## Example of request to FIWARE Context Broker

```
curl https://orion.lab.fi-ware.org:1026/ngsi10/contextEntities/MyHouse -X GET
-s -S --header 'Content-Type: application/json' --header 'Accept:
application/json' --header "X-Auth-Token: kvx0QWD8CEMd3Yl4B1Ugefnr6WcThI" |
python -mjson.tool
```

## 3.2. Configuring Cloudino Connector with MQTT IoT-Agent

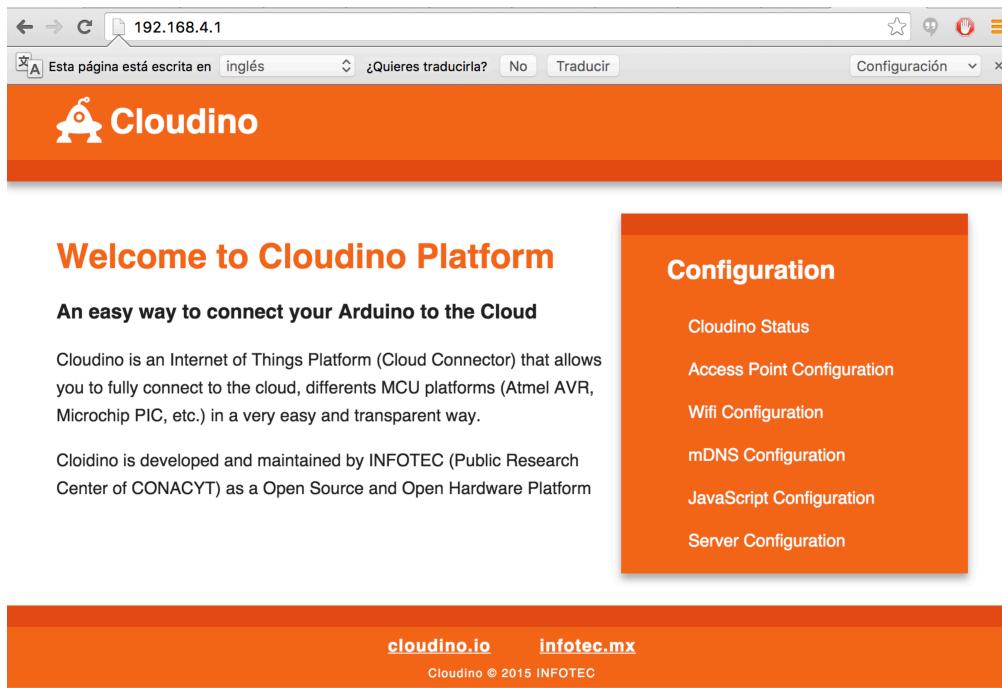
**Cloudino Connector** can connect to the FIWARE using MQTT IoT-Agent, using the simple Cloudino **Configuration Web Interface**.

The **Cloudino Connector** starts an access point that lets you connect to the configuration web interface at: <http://192.168.4.1>



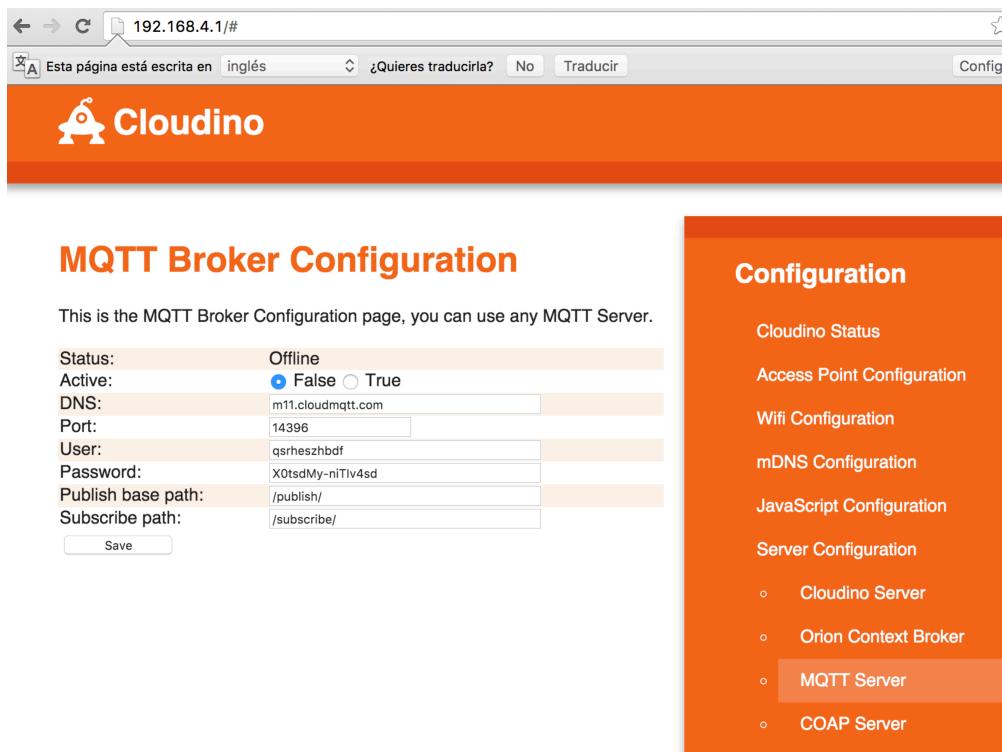
	<b>INFOTEC Centro de Investigación e Innovación en Tecnologías de la Información y Comunicación</b>	<b>Hoja</b>	6 DE 9
		<b>Versión</b>	1.0
		<b>Fecha de elaboración</b>	15/Diciembre/2015
		<b>Fecha de actualización</b>	15/Enero/2016

**Manual Integración Cloudino Connector FIWARE IoT**



The screenshot shows the Cloudino Platform homepage at 192.168.4.1. The main content area features a heading "Welcome to Cloudino Platform" and a sub-heading "An easy way to connect your Arduino to the Cloud". It includes a brief description of Cloudino as an IoT platform and mentions its development by INFOTEC and CONACYT. To the right, a sidebar titled "Configuration" lists various settings: Cloudino Status, Access Point Configuration, Wifi Configuration, mDNS Configuration, JavaScript Configuration, and Server Configuration.

To configure the MQTT Protocol to connect to FIWARE Context Broker you have to select the MQTT Server in Server Configuration and setting the next fields:



The screenshot shows the MQTT Broker Configuration page at 192.168.4.1/#. The main content area displays form fields for configuring an MQTT server, including fields for Status (Offline), Active (False selected), DNS (m11.cloudmqtt.com), Port (14396), User (qsrheshbdf), Password (X0tsdMy-niTlv4sd), Publish base path (/publish/), and Subscribe path (/subscribe/). A "Save" button is located below the form. To the right, a sidebar titled "Configuration" lists the same options as the previous screenshot, with "MQTT Server" highlighted under the "Server Configuration" section.

	<b>INFOTEC Centro de Investigación e Innovación en Tecnologías de la Información y Comunicación</b>	<b>Hoja</b>	7 DE 9
		<b>Versión</b>	1.0
		<b>Fecha de elaboración</b>	15/Diciembre/2015
		<b>Fecha de actualización</b>	15/Enero/2016

**Manual Integración Cloudino Connector FIWARE IoT**

**Active:** true

**DNS:** [dns or ip to the MTQQ Server]

**Port:** [port]

**User:** [user]

**Password:** [password]

**Publish base path:** [publish path]

**Subscribe path:** [subscribe path]

## Example of Arduino Code to Post Temperature and Humidity

```
#include <Cloudino.h>
#include <dht11.h>

#define DHT11PIN 8

Cloudino cdino;           //Cloudino Library
dht11 DHT11;              //DHT11 Library

void getSensor()
{
    int chk = DHT11.read(DHT11PIN);
    cdino.post("temperature",String((float)DHT11.temperature,2));
    cdino.post("humidity",String((float)DHT11.humidity,2));
    cdino.print("Timer done!"); //Send to console
}

void setup()
{
    cdino.setInterval(10000,getSensor); //Timer every 10 seconds
    cdino.begin();
}

void loop()
{
    cdino.loop();
}
```

## Example of request to FIWARE Context Broker

```
curl https://orion.lab.fi-ware.org:1026/ngsi10/contextEntities/MyHouse -X GET
-s -S --header 'Content-Type: application/json' --header 'Accept:
application/json' --header "X-Auth-Token: kvx0QWD8CEMd3Yl4B1Ugefnr6WcThI" |
```



	<b>INFOTEC Centro de Investigación e Innovación en Tecnologías de la Información y Comunicación</b>	<b>Hoja</b>	8 DE 9
		<b>Versión</b>	1.0
		<b>Fecha de elaboración</b>	15/Diciembre/2015
		<b>Fecha de actualización</b>	15/Enero/2016

**Manual Integración Cloudino Connector FIWARE IoT**

```
python -mjson.tool
```

## Example of Arduino Code on Receiving Messages

```
#include <Cloudino.h>

Cloudino cdino;

void alarm(String msg)
{
    if(msg=="true")
    {
        digitalWrite(13, HIGH);      //Turn the Alarm ON (HIGH is the voltage level)
    }else
    {
        digitalWrite(13, LOW);      //Turn the Alarm off by making the voltage LOW
    }
}

void setup()
{
    pinMode(13, OUTPUT);
    cdino.on("alarm",alarm);      //receive "alarm" message
    cdino.begin();
}

void loop()
{
    cdino.loop();
}
```

### 3.3. Configuring Cloudino Connector with Cloudino.io Cloud Service (work in process)

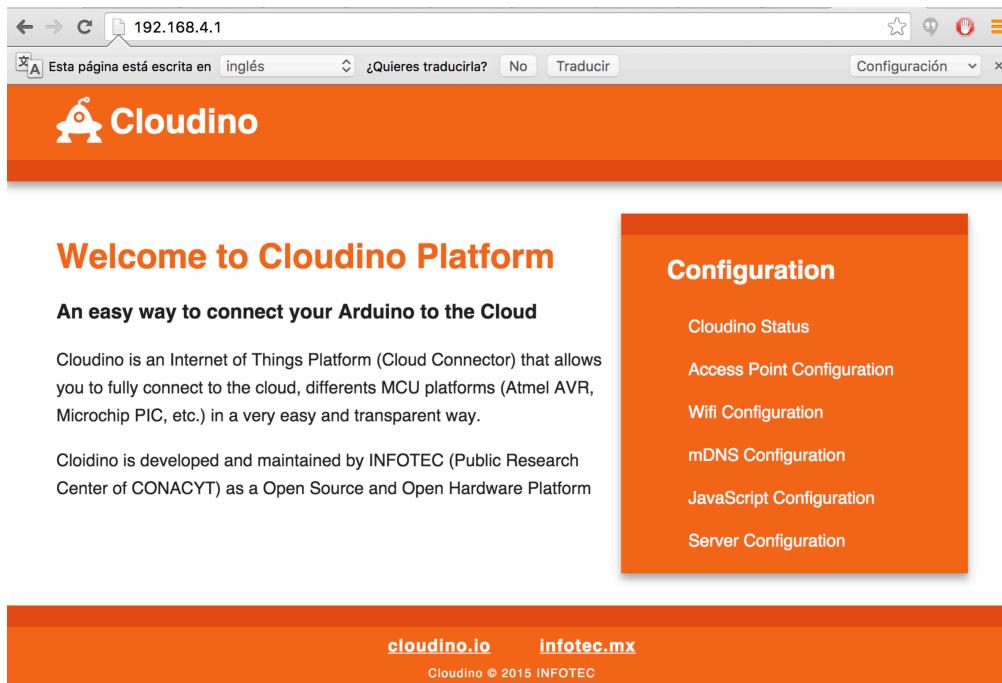
The **Cloudino Connector** can connect to the FIWARE using MQTT IoT-Agent, using the simple Cloudino **Configuration Web Interface**.

The **Cloudino Connector** starts an access point that lets you connect to the configuration web interface at: <http://192.168.4.1>



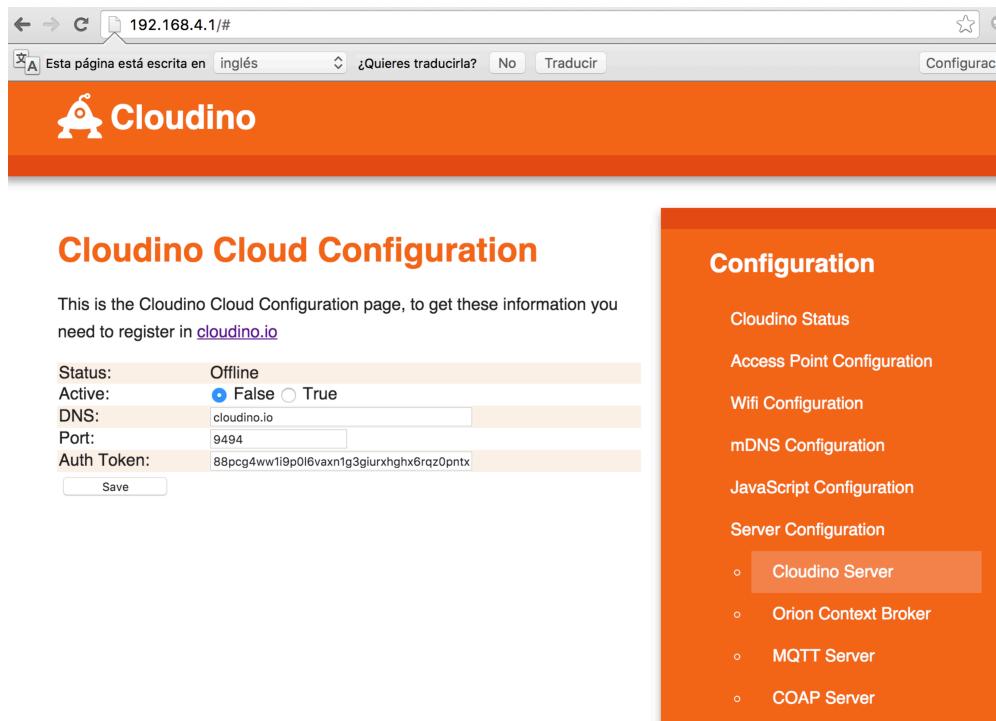
	<b>INFOTEC Centro de Investigación e Innovación en Tecnologías de la Información y Comunicación</b>	<b>Hoja</b>	9 DE 9
		<b>Versión</b>	1.0
		<b>Fecha de elaboración</b>	15/Diciembre/2015
		<b>Fecha de actualización</b>	15/Enero/2016

**Manual Integración Cloudino Connector FIWARE IoT**



The screenshot shows the Cloudino Platform landing page. At the top, there's a header bar with a back button, forward button, refresh button, address bar (192.168.4.1), a search bar, and a language selection dropdown. Below the header is a navigation bar with the Cloudino logo and the text "Welcome to Cloudino Platform". A sub-header "An easy way to connect your Arduino to the Cloud" follows. The main content area contains two columns: one for "Cloudino" (Status: Offline, Active: False, DNS: cloudino.io, Port: 9494, Auth Token: 88pcg4ww19p0l6vaxn1g3giurxghx6rqz0pntx) and another for "Configuration" (Cloudino Status, Access Point Configuration, Wifi Configuration, mDNS Configuration, JavaScript Configuration, Server Configuration). At the bottom, there are links to "cloudino.io" and "infotec.mx" and a copyright notice "Cloudino © 2015 INFOTEC".

To configure the Cloudino Connector to connect to Cloudino.io Cloud Service you have to select the Cloudino Server in Server Configuration and setting the next fields:



The screenshot shows the "Cloudino Cloud Configuration" page. It features a left sidebar with "Cloudino Cloud Configuration" and a right sidebar with "Configuration" (Cloudino Status, Access Point Configuration, Wifi Configuration, mDNS Configuration, JavaScript Configuration, Server Configuration). Under "Server Configuration", "Cloudino Server" is selected. The main form contains fields for Status (Offline), Active (False), DNS (cloudino.io), Port (9494), and Auth Token (88pcg4ww19p0l6vaxn1g3giurxghx6rqz0pntx). A "Save" button is at the bottom of the form.