

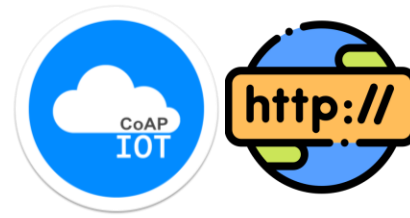
# Laboratorio 5:

## HTTP y CoAP Servers

# Agenda

- Maquina Virtual
- Demo HTTP localhost
- Demo CoAP localhost
- Copper
- Wireshark

# Máquina Virtual



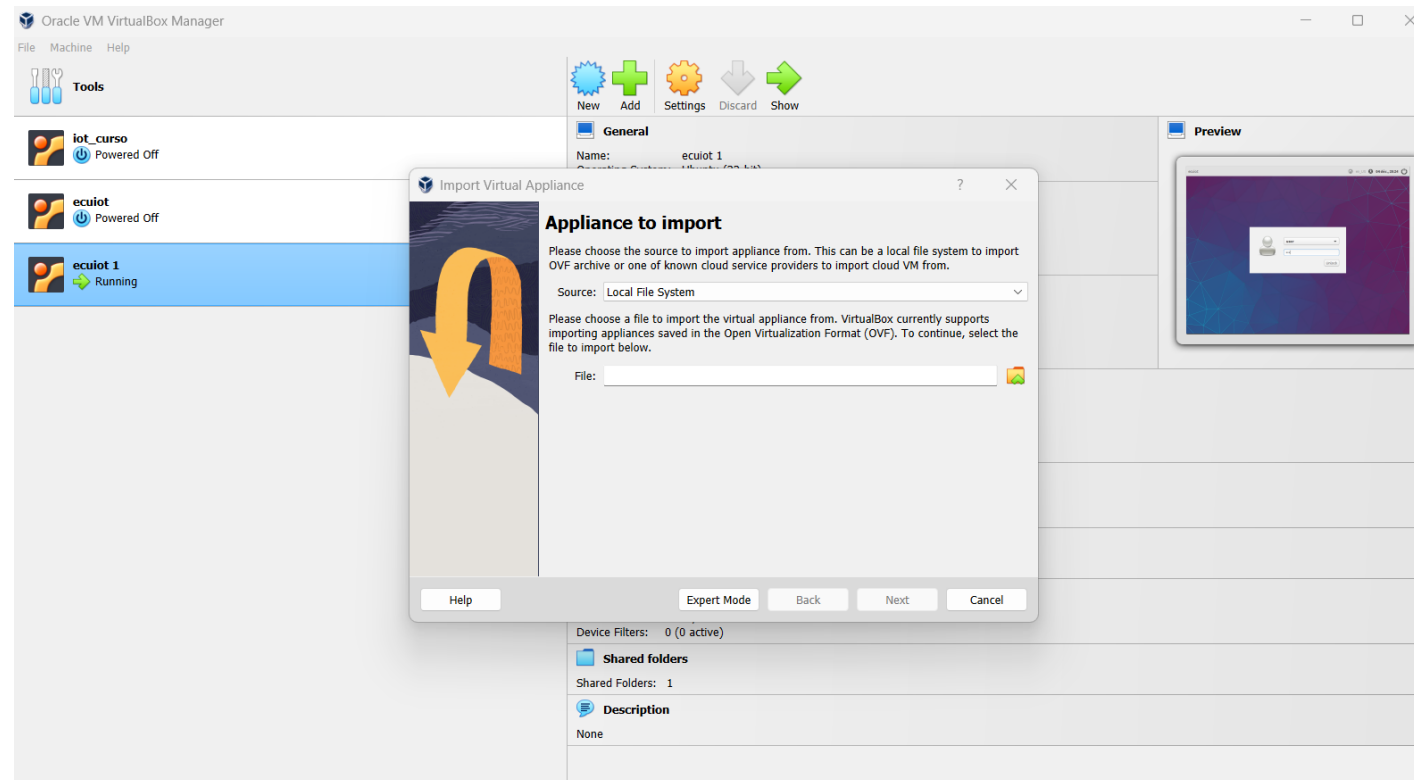
File -> Import Appliance...

File -> dirección de donde se descargó la imagen virtual.

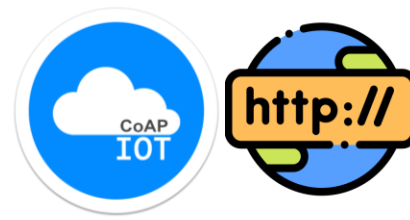
Next ->

Finish ->

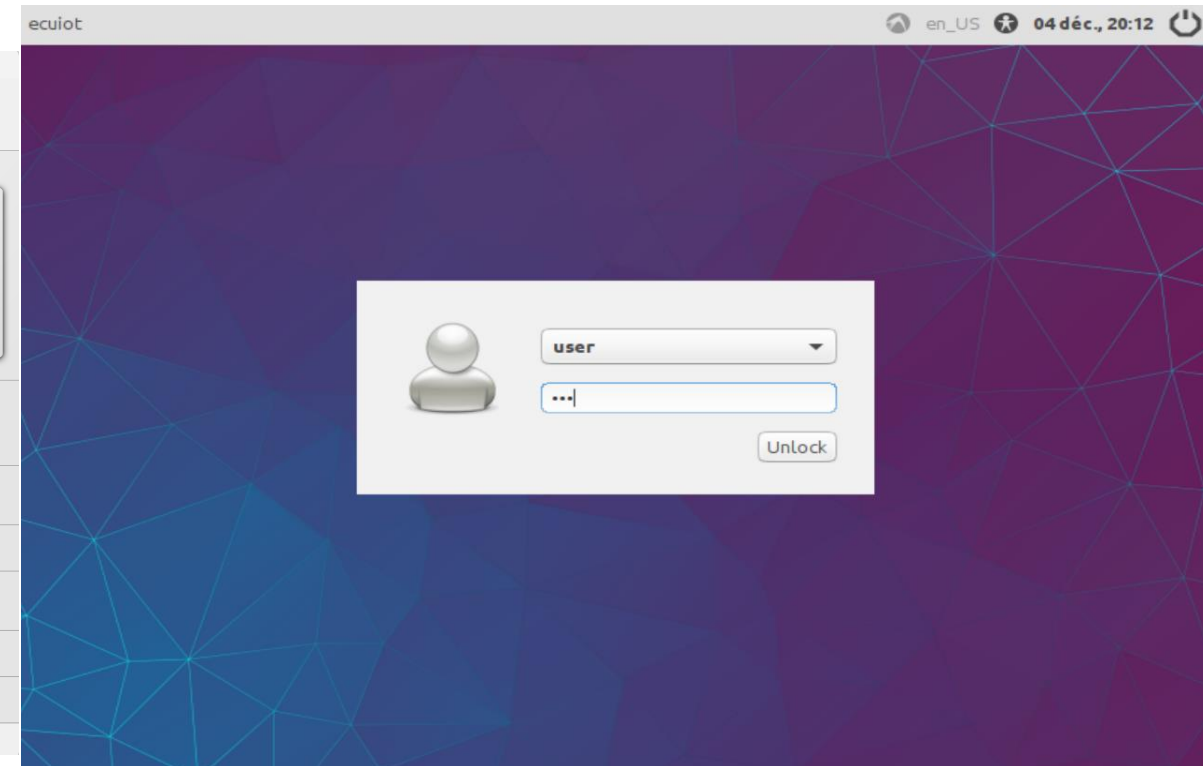
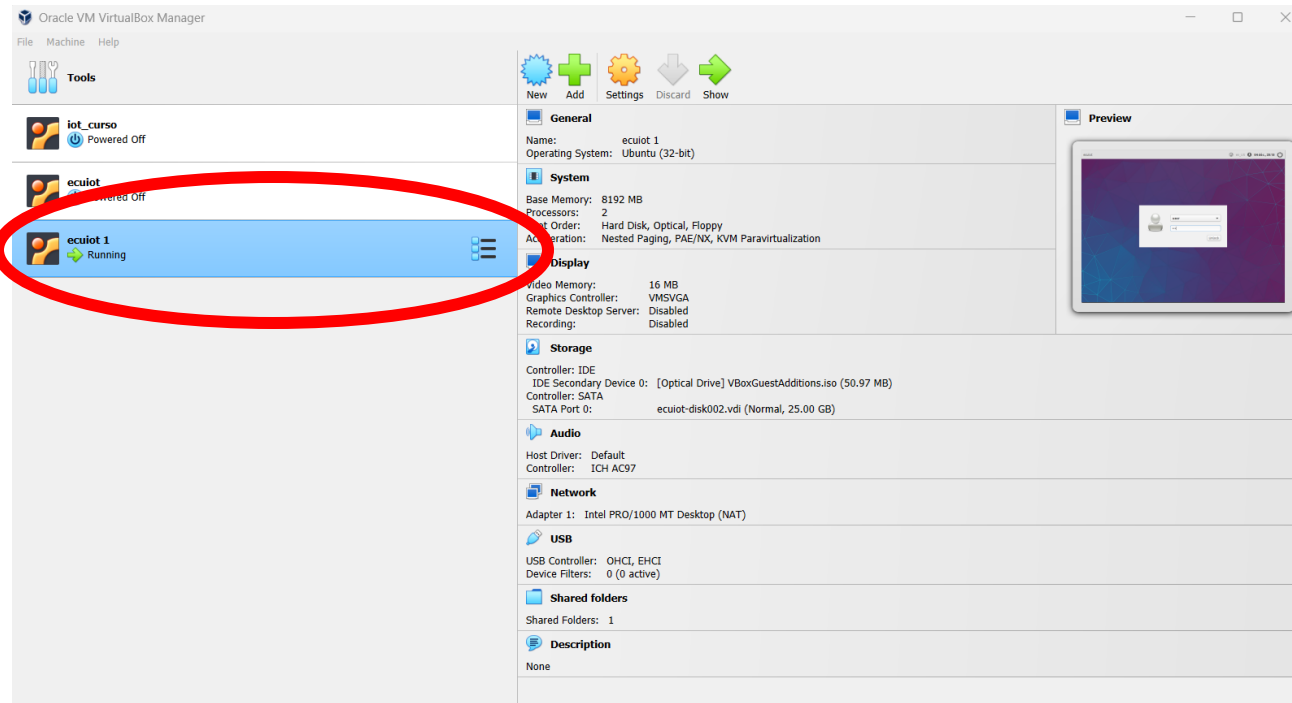
Esperar hasta que la imagen se instale



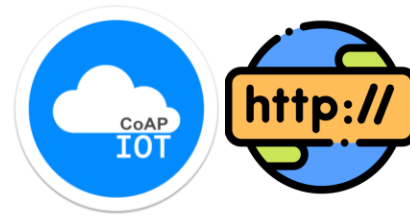
# Máquina Virtual



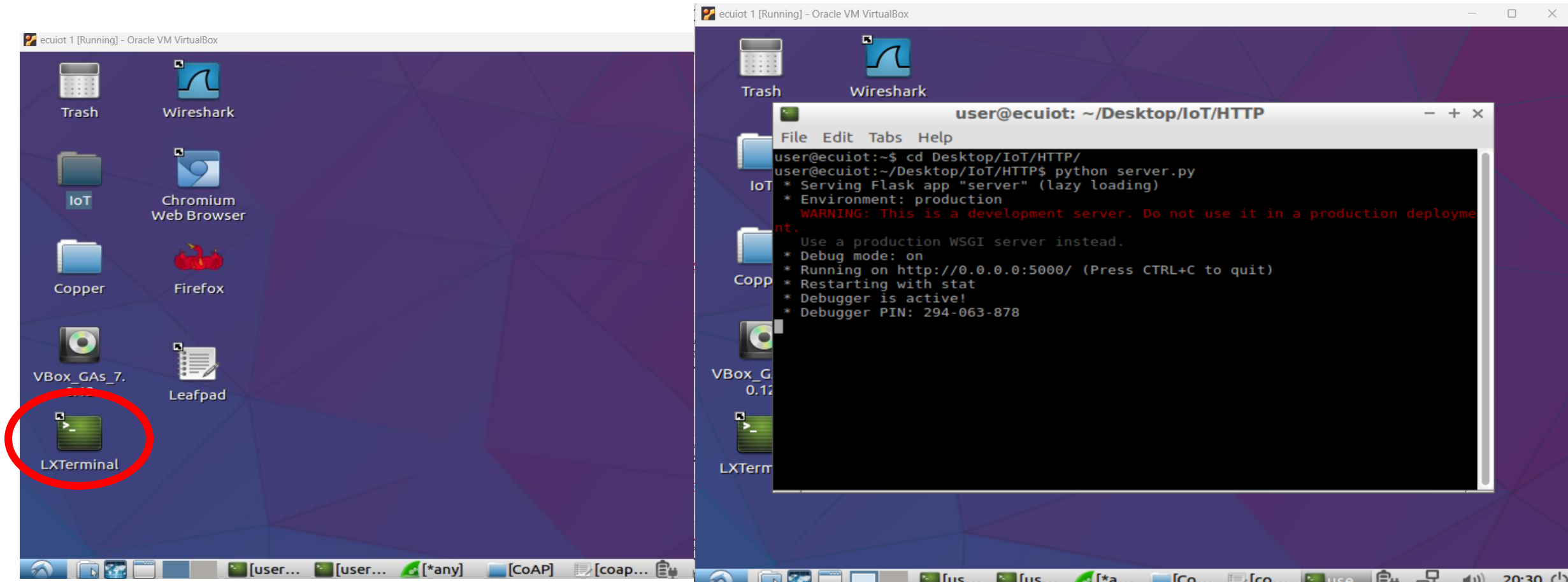
Iniciar Máquina Virtual llamada "ecuiot" desde Oracle VM.  
Clave de acceso: 123



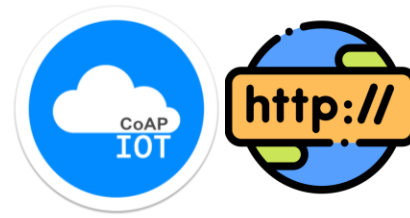
# HTTP Servicio Ejemplo



En la terminal acceder carpeta con script de HTTP servicio.  
Iniciar el servicio: `python service.py`



# HTTP Servicio Ejemplo



En una terminal nueva se puede interactuar con el servicio creado:  
`curl -X get http://0.0.0.0:5000/living_room/temperature`

A screenshot of a terminal window titled 'user@ecuiot: ~'. The terminal shows the following commands and output:

```
user@ecuiot:~$ curl X- GET http://0.0.0.0:5000/living_room/temperature
curl: (6) Could not resolve host: X-
curl: (6) Could not resolve host: GET
24.58
user@ecuiot:~$ curl -X GET http://0.0.0.0:5000/living_room/temperature
17.51
user@ecuiot:~$
```

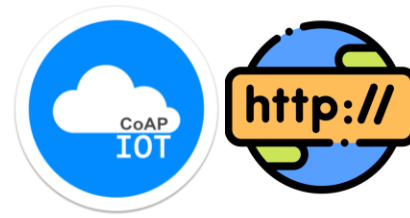
Below the terminal window, a log file snippet is visible, showing a successful GET request:

```
127.0.0.1 - - [04/Dec/2023 20:33:32] "GET /living_room/temperature HTTP/1.1" 200
```

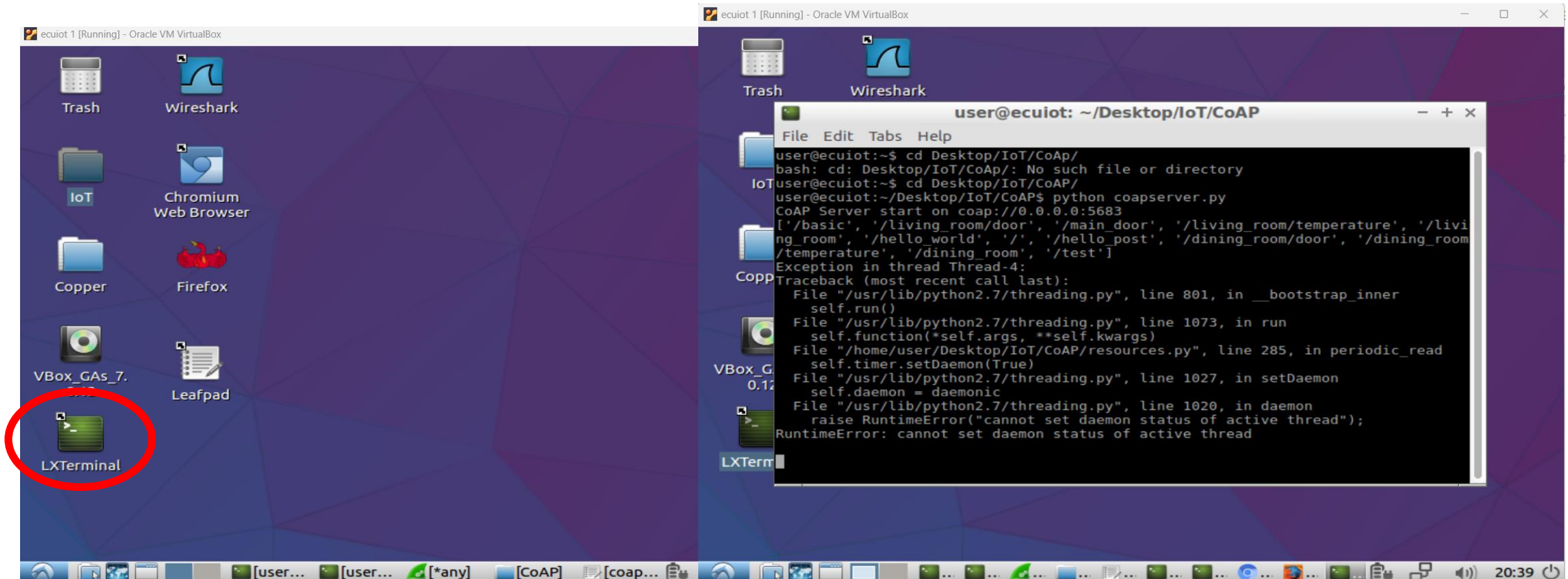
Métodos:

- GET: Lee Recursos
- POST: Crea Nuevos Recursos
- PUT: Actualiza Recursos
- DELETE: Elimina Recursos.

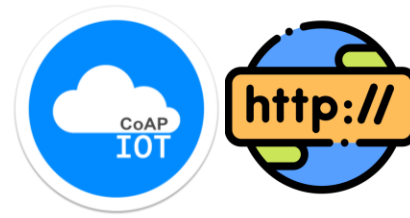
# CoAP Servicio Ejemplo



En la terminal acceder carpeta con script de CoAP servicio.  
Iniciar el servicio: `python coapserver.py`



# CoAP Servicio Ejemplo

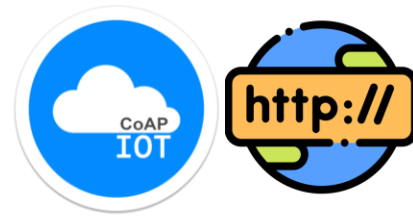


En una terminal nueva se puede interactuar con el servicio creado:  
`coap get coap://0.0.0.0:5683/living_room/temperature`

A screenshot of a terminal window titled 'ecuiot 1 [Running] - Oracle VM VirtualBox'. The terminal shows a user@ecuiot:~ prompt. The user enters 'coap://0.0.0.0:5683/living\_room/temperature' and the prompt changes to 'bash: coap://0.0.0.0:5683/living\_room/temperature: No such file or directory'. Then the user enters 'coap get coap://0.0.0.0:5683/living\_room/temperature' and the terminal displays '(2.05) 18.4'. Below the terminal window, a log window shows a debug message: '2023-12-04 20:41:06,401 - Thread-34 - coapthon.server.coap - DEBUG - send\_datagram - From None, To ('127.0.0.1', 55317), ACK-51056, CONTENT-Údö [1] 18.4...4 bytes'.



# CoAP Servicio Ejemplo con Cupper



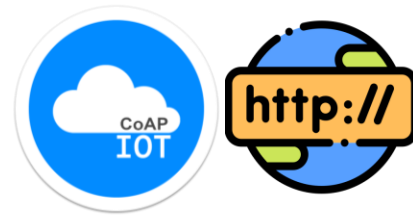
Métodos

Recursos

The screenshot shows the Cupper CoAP client interface in Mozilla Firefox. The address bar displays `coap://0.0.0.0:5683/living room`. The toolbar includes methods: Ping, Discover, GET, FETCH, POST, PATCH, PUT, iPATCH, DELETE, and Observe. The 'Discovering' tree on the left shows the resource hierarchy: `0.0.0.0:5683` (expanded) → `.well-known` (expanded) → `core` (expanded) → `basic` (expanded) → `dining_room` (expanded) → `door` (expanded) → `temperature` (expanded) → `hello_post` (expanded) → `hello_world` (expanded) → `living_room` (selected) → `door` (expanded) → `temperature` (expanded) → `main_door` (expanded) → `test` (expanded). The central area is labeled 'Petición/Respuesta'. The bottom status bar shows the CoAP message log with the following entries:

T...	CoAP Message	...	Token	Payload
21:10:54	CON-GET	15693 (0)	empty	
21:10:54	ACK-2.05 Content	15693	empty	<code>&lt;/basic&gt;;rt="rt1";if="if1",&lt;/living_room/door&gt;;&lt;/main_door&gt;;&lt;/living_room/temperature&gt;;obs,&lt;/living_room...</code>

# CoAP Servicio Ejemplo con Cupper



DISCOVER: Muestra los recursos disponibles.

GET: Lee el valor de los recursos.

POST/PULL: Cambia el valor de un recurso.

POST: Crea un nuevo recurso.

DELETE: elimina recursos.

OBSERVE: Obtiene valores de recursos cada vez que cambia.

# CoAP con Wireshark inspección de paquetes



- **coap.code == X**

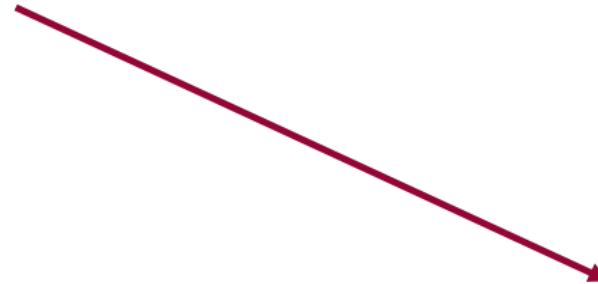
X puede ser el método requerido



Code	Name	Reference
1	GET	[RFCXXXX]
2	POST	[RFCXXXX]
3	PUT	[RFCXXXX]
4	DELETE	[RFCXXXX]

CoAP Method Codes

X puede ser el código de respuesta



**coap.type == Y**

Y puede ser el tipo de mensaje

0 = CON

1 = NON

2 = ACK

3 = RST

Code	Description	Reference
65	2.01 Created	[RFCXXXX]
66	2.02 Deleted	[RFCXXXX]
67	2.03 Valid	[RFCXXXX]
68	2.04 Changed	[RFCXXXX]
69	2.05 Content	[RFCXXXX]
128	4.00 Bad Request	[RFCXXXX]
129	4.01 Unauthorized	[RFCXXXX]
130	4.02 Bad Option	[RFCXXXX]
131	4.03 Forbidden	[RFCXXXX]
132	4.04 Not Found	[RFCXXXX]
133	4.05 Method Not Allowed	[RFCXXXX]
134	4.06 Not Acceptable	[RFCXXXX]
140	4.12 Precondition Failed	[RFCXXXX]
141	4.13 Request Entity Too Large	[RFCXXXX]
143	4.15 Unsupported Media Type	[RFCXXXX]
160	5.00 Internal Server Error	[RFCXXXX]
161	5.01 Not Implemented	[RFCXXXX]
162	5.02 Bad Gateway	[RFCXXXX]
163	5.03 Service Unavailable	[RFCXXXX]
164	5.04 Gateway Timeout	[RFCXXXX]
165	5.05 Proxying Not Supported	[RFCXXXX]

CoAP Response Codes