

# Node y JSON – brevísima introducción

Carlos Lombardi

Universidad Nacional de Quilmes – Argentina

Sesiones sobre microservicios

# Node – en dos palabras (1)

**Qué es – un runtime environment para Javascript**

Ejecuta código JS fuera de un browser.

**Para qué se usa**

Código server-side de una aplicación Web.

Se lleva bien con el desarrollo basado en microservicios.

# Node – en dos palabras (2)

## Por qué surge

- ▶ Desarrollo homogéneo de una aplicación Web.  
Usar el mismo lenguaje para server y cliente
- ▶ Máquinas virtuales JS muy rápidas (V8 de Google).

## Otros comentarios

- ▶ Ciclo de desarrollo muy corto (ya lo vamos a ver).
- ▶ Muchas librerías (npm – manejador de librerías).
- ▶ Muy popular.

# Dónde podemos ejecutar código Javascript

- ▶ Browsers.
- ▶ Sandboxes como repl.it (sólo para experimentar).
- ▶ Node.js .

# Jugamos un poco con Node

...

## Accediendo a un servicio Web

```
const request = require('sync-request')    // library
const uri = 'http://...'
const response = request('GET', uri)
console.log(response.body.toString())
```

Respuesta

Texto estructurado

```
'[{"LocalObservationDateTime":"2017-03-31T19:55:00-03:00",
"Temperature":{"Metric":{"Value":23.9,"Unit":"C","UnitType":17},
"Imperial":{"Value":75.0,"Unit":"F","UnitType":18}} ...]'
```

# JSON

Formato de texto estructurado (como XML o YAML).

Similar a JS object notation:

- ▶ Permite anidamiento.
- ▶ No permite métodos, solamente propiedades.

```
'[{ "LocalObservationDateTime": "2017-03-31T19:55:00-03:00",  
  "Temperature": {  
    "Metric": { "Value": 23.9, "Unit": "C", "UnitType": 17 },  
    "Imperial": { "Value": 75, "Unit": "F", "UnitType": 18 }  
  }  
}]'
```

# JSON y Javascript – buenos amigos

JSON.parse

Transforma String JSON a un objeto Javascript.

La inversa: JSON.stringify

```
let jsonString = '{"name": "Carlos", "age": 48}'  
JSON.parse(jsonString).age    // 48  
  
let obj = { name: 'Juan', age: 32 }  
JSON.stringify(obj)           // '{ "name": "Juan", "age": 32 }'
```