

# Trabajo Práctico N°3

Nahuel Defossé

2018

## Taller de Nuevas Tecnologías

### Notas

Los trabajos pueden realizarse en grupos de hasta *dos personas*. La fecha de entrega será pautaada, en dos semanas a partir de la fecha en la cual esté disponible. El formato de entrega puede ser PDF o HTML (o enlace a repositorio dónde debe haber un documento README.md, README.rtf y/o un Cuaderno Jupyter). Se deberá incluir evidencia de el trabajo realizado. Los trabajos no aprobados tendrán un período de recuperación de una semana posterior a la fecha de recepción de las devoluciones.

### Creación de Nodos y uso de File API de HTML5 para archivos SVG

1. Integre NodeRED en una aplicación Express <sup>1</sup>. Utilice el proyecto heredado del práctico anterior.
2. Agregue una vista en la aplicación que contiene a NodeRED que sirva a través de HTTP un formulario que utilice la **File API** de HTML5. <sup>2</sup> Inyecte los contenidos en un formulario mediante manipulación de DOM como entradas ocultas.
3. Basándose en la guía oficial de NodeRED <sup>3</sup> cree un Nodo llamado **SVGStorage** que almacene dentro de la definición archivos SVG.
4. Agregue a **SVGStorage** la capacidad de responder a mensajes desde nodos HTTP. No debe crear nuevos objetos como respuestas, sino aumentar las entradas. Ante un mensaje

---

<sup>1</sup><https://nodered.org/docs/embedding>

<sup>2</sup><https://www.w3.org/TR/FileAPI/>

<sup>3</sup><https://nodered.org/docs/creating-nodes/>

```
{
  topic: "svg-list"
}
```

debe ser capaz de responder con un diccionario de objetos SVG (archivos) almacenados (a través del pto 3.). Ej:

```
{
  topic: "svg-list"
  payload: [
    {
      id: 'file-1',
      name: 'Diagrama 1',
    },
    {
      id: 'file-1',
      name: 'Diagrama 2',
    }
  ]
}
```

5. Agregue a `SVGStorage` la capacidad de responder mensajes donde el tópico sea `svg-get-<id>`. El `payload` debe ser el contenido del archivo.

```
{
  topic: "svg-get-ID"
}
```

6. Agregue el código necesario para poder obtener los archivos si los mensajes son ingresados a través de un nodo de entrada HTTP. Cree un flujo con entrada HTTP, su nodo y un nodo de salida.