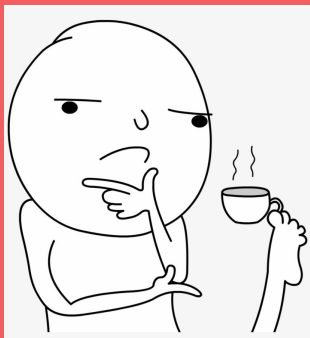




¿Qué es single-spa?

Es un framework que nos permitirá implementar de forma sencilla una aplicación web mediante microfrontends.

“a javascript router for frontend microservices”



¿Cómo?

1. Desarrollar cada microfront (registered app).
2. Configuración rutas internas.
3. Registrar cada microfront en la root app
4. Habilitar en el index.html para montar las app registradas.
5. Iniciar la root app con el enrutador principal.

Estrategias

Monorepo

- 1 solo script webpack o 1 por cada app registrada
- Demasiado complicado si hay varios frameworks

Paquetes NPM

- 1 repositorio por app
 - Se importa cada microfront como un paquete de NPM en lugar del js compilado.
 - Compilado de la app root muy pesado
-

Estrategias

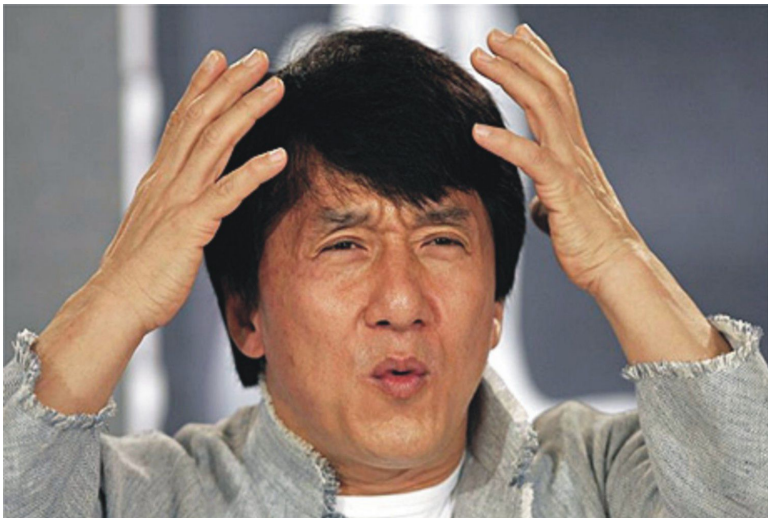
Carga dinámica de módulos

- 1 repositorio por app
- las dependencias se resuelven en runtime
- SystemJS para carga dinámica.

Comparación estrategias

	Monorepo	NPM modules	Module loading
Difficulty to set up	Easy	Medium	Hard
Separate code repositories	No	No	✓
Separate builds	No	✓	✓
Separate deployments	No	✓	✓
Examples	<ul style="list-style-type: none">• simple-webpack-example• single-spa-examples	Contributions accepted!	<ul style="list-style-type: none">• SystemJS example





Comandos

Instalación:

```
npm install --global create-single-spa
```

App root:

```
npx create-single-spa --moduleType  
root-config
```

Micro app:

```
npx create-single-spa --moduleType  
app-parcel
```

Registrar una aplicación

```
registerApplication({  
  name: "@conceptoSura/header-mf",  
  app: () => System.import<LifeCycles>("@conceptoSura/header-mf"),  
  activeWhen: ["/"]  
});
```

```
<% if (isLocal) { %>  
<script type="systemjs-importmap">  
  {  
    "imports": {  
      "@conceptoSura/root-config": " //localhost:9000/conceptoSura-root-config.js",  
      "@conceptoSura/header-mf": " //localhost:4200/main.js"  
    }  
  }  
</script>  
<% } %>
```