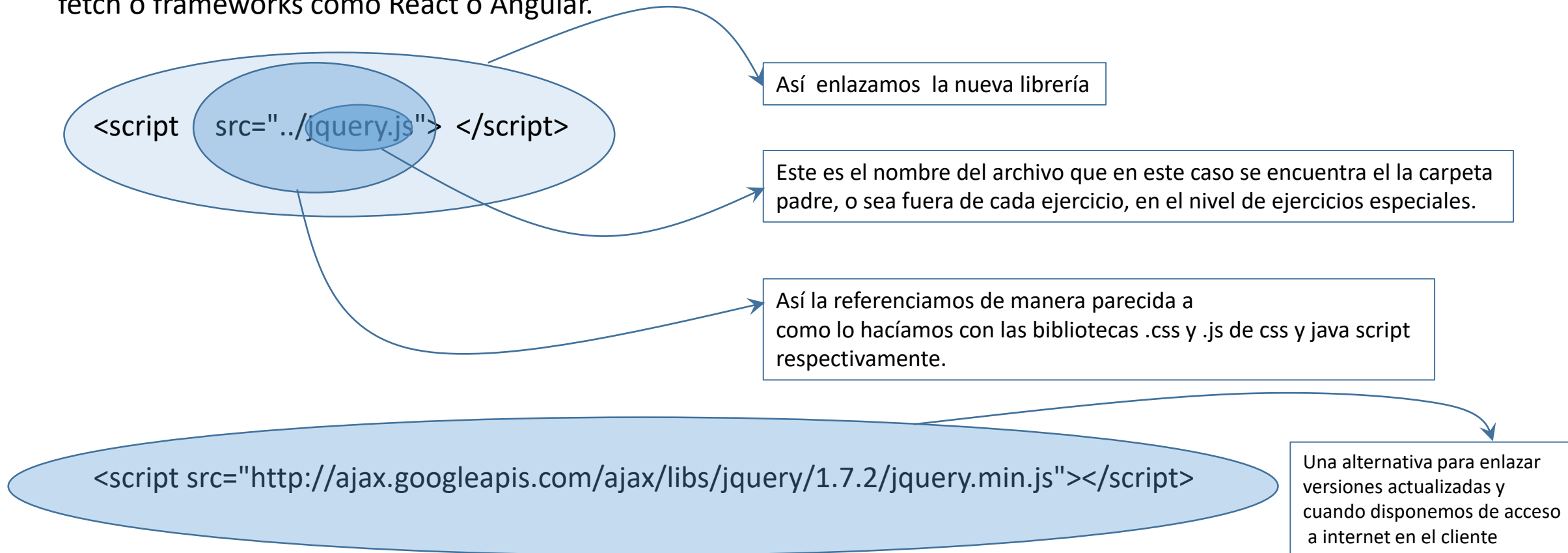


**ACLARACION: Leer bien documento de JSON antes de empezar con este.**

### Librería JQuery:

Con el objeto de obtener un poco más de funcionalidad en el lenguaje, de simplificar el trabajo de programación y de aprender a trabajar con una librería externa trabajaremos haciendo uso de esta herramienta.

Jquery se sigue usando para proyectos heredados, proyectos simples, complementos de Wordpress, tableros de control y demás. Las alternativas modernas es el uso de API Web's nativas como la API fetch o frameworks como React o Angular.



**Todo el código Java script se puede quitar del documento HTML y escribirlo en un archivo aparte ( \*.js), de igual manera que la librería de JQuery.**  
**Podemos también declarar una variable de tipo texto JSON dentro de un archivo .json**

`<script src="./arregloEmpleados.js" type="text/javascript"></script>`

Dentro de este archivo estaría la declaración de la variable de texto JSON. Así quedaría referenciada en el código HTML.

## Algo de JQuery

`$(document)`

Así se referencia el Objeto Documento en el DOM usando JQUERY.  
Más sencillo que el resultado devuelto por el método `document.getElementById("")`.

Método `ready()` aplicado al documento.  
Recibe como argumento una función anónima.

```
$(document).ready(function() {
```

Funcion anónima para definir código a ejecutar en el evento de finalización de carga del documento en el navegador. Esto permitiría escribir código java script en cualquier parte del documento

```
/*Codigo a ser ejecutado cuando termine de cargar el documento*/
```

```
});
```

Cierra función anónima  
Cierra argumentos del `ready()`  
Termina instrucción.

```
$(document).ready(function() {
```

```
$("#btAccionCarga").click(function() {
```

Objeto button del DOM.  
Esto es el equivalente a `document.getElementById("btAccionCarga")` pero aplicable cuando se usa jQuery. Observar que se usan los mismos selectores que en CSS. Numeral por ejemplo hace referencia a un elemento único del documento con su id.

```
/*Codigo a ejecutar en el evento click*/
```

Observar que si referencio al objeto usando jQuery, entonces debo usar un método compatible con jQuery. Este caso se trata de un método `click()` en lugar de una propiedad `onClick` como aplicábamos cuando no usábamos jQuery.  
El método `click()` recibe como argumento la función anónima a ejecutar.

```
});
```

Cierra función anónima  
Cierra argumento del `click()`  
Termina instrucción.

```
});
```

Codigo a ejecutar cuando el documento este totalmente cargado Y cuando se pulse click sobre `btAccionCarga`

## Vaciar un contenedor usando jQuery

`$("#tbDatos").empty();`

Objeto del DOM representando por ejemplo a un elemento html `<tbody id="tbDatos">`

Este método solo aplicable en JQuery permite vaciar al elemento contenedor de todo contenido. Recordar que cuando NO usábamos JQuery, había que barrer el arreglo de objetos contenidos y eliminarlos de a uno usando el método `removeChild()`.

## Asignar un estilo a un elemento HTML

`$('#contenedor').attr("class","contenedorActivo");`

El método `attr()` permite cambiar un atributo de un elemento. Los argumentos pasados se corresponden con el nombre del atributo y el valor del mismo.  
`className` se usa con javascript, `class` con jquery

Observar que en JQuery todos los métodos aplicados reemplazan a los atributos.  
NO aplicar métodos de JQuery a objetos referenciados como `document.getElementById("")`.

## Creación dinámica de opciones para un select

Dada la siguiente lista desplegable en html (VACIO DE OPCIONES `<option></option>`)  
Llenarlo de ellas usando Java script y tomando como fuente una variable JSON.

```
<select id="selectArticulosUm" name="formArticulosUm" required>
```

## Html <SELECT> vacío de opciones

```
var textoUnidadesDeMedida = '{\"unidadesDeMedida\": [' +
'{\"codUm\":\"KG\",\"descripcionUm\":\"Kilogramos\"},'+
'{\"codUm\":\"BO05LT\",\"descripcionUm\":\"Bolsa 0.5 litros\"},'+
'{\"codUm\":\"LAT\",\"descripcionUm\":\"Lata\"}'+
']}';
```

JSON para definir unidades de medida.

Las comillas simples interiores y el concatenador+ es solo para poder visualizar mejor la cadena JSON.

# Crear un objeto objJson, asignarlo a una variable de java script y barrerlo para crear dinámicamente el select

```
const objJson=JSON.parse(textoUnidadesDeMedida);
```

Utilizando este método estatico, se obtiene una variable de Java script de tipo objeto a partir de una variable de texto json. En este caso resulta un objeto con un solo atributo: "nombre"="UnidadesDeMedida" Este atributo es de tipo array y sus elementos son objetos . Leer la sintaxis del texto JSON de la pagina anterior:

```
function creaOpciones() {
```

```
objJson.unidadesDeMedida.forEach(function(argValor,argIndice) {
```

Esto es el atributo unidadesDeMedida del objeto objJson  
Es un array

```
var objOpcion= document.createElement("option");
```

El método forEach() aplicado al array ejecutará el código de una función anónima a cada uno de los elementos. Como argumentos a usar en cada ciclo aparecen argValor que corresponde a los valores de cada elemento del array en sí mismo y el índice numérico asociado [0], [1], .....

Se crea dinámicamente en memoria un objeto nuevo del DOM (en este caso un option)

```
objOpcion.setAttribute("className","elementoOptionSelect");
```

Al nuevo objeto se les asigna las propiedades className para asociar estilo y value para darle valor. Recordemos que class es solo con jquery

```
objOpcion.setAttribute("value",argValor.codUm);
```

```
objOpcion.innerHTML = argValor.descripcionUm;
```

Al nuevo objeto se le asigna un nodo de texto con la descripción del elemeto (es lo que se verá en la lista desplegable

```
document.getElementById("selectArticulosUm").appendChild(objOpcion);
```

El nuevo objeto objOption es adicionado al objeto Select del DOM

Cierra funcion anonima, cierra arg de forEach() y termina instrucción.

Cierra function creaOpciones()

```
}
```

```
});
```

## Ejemplo de como crear dinámicamente filas en el tbody de una tabla:

**Primero:** Parsear el texto JSON para obtener una variable java script tipo objeto.

**Segundo:** Barrer con un forEach() cada elemento del array obtenido como primer atributo del objeto.

**Tercero:** Crear dinámicamente filas y columnas de la tabla de igual manera que cuando creábamos opciones de un select.

```
var objTr= document.createElement("tr");
```

Crear en memoria un elemento html fila

```
var objTd=document.createElement("td");
```

Crear en memoria un elemento html celda.

```
objTd.setAttribute("campo-dato","articulos_codArt");
```

Asignar un par nombre – valor como atributo de la celda creada

```
objTd.innerHTML=argValor.codArt;
```

Asignar un contenido html como nodo de texto de la celda creada

```
objTr.appendChild(objTd);
```

Adicionar celda creada al objeto fila creado

Repetir estas asignaciones por cada atributo del artículo.

```
objTbDatos.appendChild(objTr);
```

Adicionar la fila creada en memoria al elemento tbody de la tabla

## Haciendo visible un contenedor modal

```
$(document).ready(function() {  
    document.getElementById('contenedor').className="contenedorActivo";  
    document.getElementById('ventanaModal').className="ventanaModalApagado";  
});
```

Nombre de clase del estilo definido para un contenedor principal activo. En el se activarán todos los controles.

Nombre de clase del estilo definido para un contenedor modal apagado. (Contenedor donde por ejemplo pueda mostrarse un formulario).

En la versión jQuery sería:  
\$('#contenedor').attr("class","contenedorActivo");

CSS

```
div.ventanaModalApagado {  
    visibility:hidden;  
}  
  
div.ventanaModalPrendido {  
    visibility:visible;  
}  
  
div.contenedorActivo {  
    opacity:1;  
    pointer-events: auto;  
}
```

Activacion/Desactivación de contenedores  
Principal/Modal

Permite quitar toda transparencia el contenedor principal

Permite activar todos los elementos sensibles a evento.  
Con valor none se desactivan.



```
div.contenedorPasivo {  
  opacity:0.3;  
  pointer-events: none;  
}
```

Otorgar transparencia al contenedor

Permite desactivar todos los elementos sensibles a evento.