

## Práctica Integradora

1. Tomando como punto de partida el archivo **app.html** vamos a generar un archivo **js/app.js** y vincularemos los dos archivos.
2. El objetivo de esta práctica es generar un *To Do List* que se deberá ver así:

### To do list

Ingrese una tarea:

3. Para generar los elementos `<li>` dentro de la lista vamos a utilizar el siguiente código html:
  - a. Para los elementos que no han sido clicados:

```
<li class="list-group-item list-group-item-info"></li>
```
  - b. Para los elementos clicados:

```
<li class="list-group-item list-group-item-danger"></li>
```

4. Una vez con 1 o más tareas agregadas a la lista, si hacemos clic en el botón **¿Guardar tareas?** deberá agregarse otro botón que diga **¿Olvidar tareas?**. Y adicionalmente se deberá informar al visitante la cantidad de tareas guardadas durante un periodo de 3 segundos.

## To do list

Ingrese una tarea:

  

Agregar Tarea

Una tarea

×

Otra tarea más

×

Otra más

×

¿Guardar tareas?

5. Otro de los objetivos del botón **¿Guardar tareas?** será agregar el contenido de la lista al almacenamiento interno del navegador. Por lo tanto: si la 1er vez que se ingresa al documento, existe un listado de tareas almacenado, el mismo se cargará de manera automática en la lista.

6. Por otro lado, si se clicla el botón **¿Olvidar tareas?** se deberá resetear el almacenamiento interno del navegador para olvidar dichos datos. Y a su vez se deberá informar al visitante que se olvidaron dichas tareas:

## To do list

Ingrese una tarea:

Agregar Tarea

Una tarea	×
Otra tarea más	×
Otra más	×

¿Guardar tareas?

¿Olvidar tareas?

7. Así mismo, en el archivo **my-to-do-list.html** se deberá visualizar el listado de tareas almacenado así:

## Tienes un total de 3 tareas:

Una tarea	×
Otra tarea más	×
Otra más	×

8. Si por el contrario en el almacenamiento interno no hay datos, en éste último archivo se deberá mostrar un texto **"Excelente, no tienes tareas pendientes"**.

9. Todo lo anterior se deberá generar con las notaciones de ES6: `let`, `const`, `arrow functions`, `template literals`, etc.