POO - To-Do v3.0

1.- Inicio carpetas → Crear toda la arquitectura de archivos y dependencias

- Instalar bootstrap, sass y bootstrap-icons
- Crear el archivo Sass y el script en el `package.json` para compilar sass
- Importar en nuestro `style.scss` bootstrap , bootstrap-icons y una fuente de google.

2.- Inicio de HTML → Crear el markup de toda la app en HTML

- Un input (con botón o sin botón)
- Una caja que recibirá todas las tareas.

3.- Inicio de JavaScript → Creamos toda la base de nuestra App

- Una class genérica que tenga toda nuestra To-DoApp
  - Las propiedades que necesite para empezar
  - Instanciar por primera vez nuestra To-DoApp .
- Una class para cada To-do `class To-do`
  - Con: id , task
    - El id lo podemos traer del paquete `nanoid` de npm.
- Exportamos cada clase y las importamos donde haga falta.

4.- Cazar lo que el usuario ponga y guardar su valor

- Cazar el form/input/button/loquesea.
- Obtener su valor.
- Evaluar si los valores introducidos son correctos
- Crear un nuevo ToDo con los datos.
- Añadirlo a la "base de datos".

5.- Imprimir todos los ToDos en su sitio

- Vaciar la caja de ToDos actual del DOM
- Recorrer nuestra "base de datos"
- Crear el "markup" de cada ToDo
- Añadir todos los nuevos ToDos al DOM
- Resetear el input a "vacío" y con su "focus" activado.

6.- Refactorizar la función hasta ahora

- Que handle... solo se encargue de distribuir el código, no debería hacer la instancia del nuevo `ToDo` y meterlo en la "BD". Eso debería estar en otra función.
- Crear el "markup" del nuevo ToDo debería estar en otra función.
- El reset del formulario debería estar en otra función.
- Haz la refactorización y revisa que todo siga exactamente igual.

Ya tenemos la `C` y la `R` de nuestro CRUD. Podemos crear elementos y leerlos en pantalla!!!!😃

7.- Vamos a eliminar elementos

- Escuchar el click del botón X
- Si hemos creado cada ToDo con `createElement` podemos añadirle el addEventListener a cualquiera de sus botones en el momento de ser creados, con un `querySelector` y además PODEMOS PASARLE EL ID DIRECTAMENTE
- Si, en cambio, hemos creado el ToDo con innerHTML tendremos que hacer un `Event Delegation` y escuchar el click al botón que tenga la clase de borrar.
- Borramos con la función correspondiente a ese ToDo en concreto de la "BD". → `filter`
- **¡EXTRA!** Podéis añadir una animación para que se vaya suave

8.- Cambiar el estado de un ToDo a `completado`

- Crear la propiedad `isCompleted` en la class `ToDo` inicializada en `false`
- Escuchar el click en el botón de completar y lanzar su función (*lo mismo que con el delete*)
- Cambiar el estado en la base de datos:
  - Que se encargue la clase principal de hacer eso con un `map()`
  - Lo más correcto sería que se encargara el propio ToDo de cambiar su propio estado. Para ello creamos un método nuevo en la class `ToDo` y buscamos el objeto desde la clase principal con el `find()` y lo ejecutamos.
- Cambiar el icono a "completado"
- **¡EXTRA!** Añadir una animación al cambiar el estado.

YA TENEMOS TODO NUESTRO CRUD (en este caso sin hacer update)! 😎

9.- Añadir el markup de la botonera de filtros

- Puro HTML. Simplemente dejar preparados los botones con sus clases para llamarlos desde javascript.
- En nuestro caso, además, dejamos preparada la caja donde saldrá el número de tareas pendientes.

10.- Añadir con JavaScript el número de tareas pendientes

- Buscar los 3 momentos donde cambia el número de tareas activas y lanzar la función que imprima el número de tareas que queden.

11.- Filtrar resultados por "Todas" / "Activas" / "Completadas"

- Conseguir los 3 botones y para cada uno filtrar la "BD" según convenga.
- Imprimir los ToDos filtrados:
  - Cambiar la función actual que imprime todos los ToDos para que los imprima desde un array que le pasemos, no desde la "BD".
- Cambiar los estilos de los botones para que se quede marcado el que has hecho click.

12.- Limpiar todos los completados

- Escuchar el botón que limpia y lanzar su función.
- Borrar todos los ToDos completados de la "BD".
- Imprimir el resto.

13.- LocalStorage

- En cada punto donde cambiamos nuestra "BD" añadimos el array a localStorage (con una función);
  - Cuando añadimos.
  - Cuando eliminamos.
  - Cuando cambiamos completado o no completado.
  - Cuando limpiamos todos los completados
- Nada más empezar la app, conseguir datos del localStorage e imprimirlos.
  - Mirar si hay elementos en el LS
    - Si hay, sobreescribir nuestra "BD"
    - Si no hay, poner un array vacio en la "BD"
  - Imprimirlos

¡TENEMOS NUESTRA APP DE ToDos ACABADA!