

Actividades Repaso UD 10

Ejercicio 1 - Lista de tareas con componentes

Vamos a seguir con la aplicación de la lista de cosas que hacer pero dividiéndola en componentes.

La decisión de qué componentes crear es subjetiva pero en principio cuanto más descompongamos más posibilidades tendremos de reutilizar componentes. Nosotros haremos los siguientes componentes:

- **todo-list**: muestra la lista de tareas a realizar. Dentro tendrá:
 - **todo-item**: cada una de las tareas a hacer
- **add-item**: incluye el formulario para añadir una nueva tarea (el input y el botón)
- **del-all**: el botón para borrar toda la lista

Pasos que se deben realizar:

1. Crear el componente que mostrará la lista: *todo-list*.
 - Su *template* es un div que incluye el título (que será una variable para poderlo reutilizar) y la lista con las tareas a mostrar. Cada una de ellas será un subcomponente llamado *todo-item*
 - Como parámetro recibirá el título de la lista como hemos indicado antes
 - Llama al subcomponente *todo-item* para cada tarea (v-for) y le pasa la tarea que debe mostrar
 - Sus datos serán el array de tareas
 - Los métodos los dejamos tal cual aunque ahora no funcionan porque nadie los llama.
2. Crear el componente al que llama el anterior, *todo-item*.
 - Recibirá un objeto con la tarea a mostrar
 - Su *template* será el que tenía en el HTML pero quitando el *v-for* porque él sólo se encarga de mostrar 1 item
 - El método para borrarlo al hacer doble click ya no puede funcionar porque el componente no tiene acceso al array de tareas. De momento sólo ponemos un *alert*

que nos diga que lo queremos borrar

3. Crear el componente *add-item*.

- Su *template* será el `<input>` y el `<button>` que teníamos en el HTML, pero como sólo puede haber un elemento en el template los incluimos dentro de un
- No recibe ningún parámetro pero sí tiene una variable propia, *newTodo*, que quitamos del componente principal para añadirla a este componente
- El método `addTodo` ya no funciona porque no tengo acceso al array de tareas así que de momento muestro un *alert* con lo que querría añadir

4. Creo el componente *del-all*

- Su *template* es el botón
- Ni recibe parámetros ni tiene variables propias
- Con el método pasa lo mismo que en los otros casos así que simplemente muestro un *alert*

Ejercicio 2 - Componentes en SFC

Recordemos que la aplicación que estamos desarrollando tiene los componentes:

- *todo-list*: lista de tareas a hacer. Cada item de la lista es un componente *todo-item*
- *todo-item*: cada elemento de la lista de tareas a hacer
- *todo-add*: formulario para añadir una nueva tarea
- *todo-del-all*: botón para eliminar todas las tareas

Para transformar esto en SFC simplemente crearemos un fichero para cada uno de estos componentes. Nuestro anterior *index.html* será el `<template>` del componente principal **App.vue**, que en una sección `<script>` deberá importar y registrar cada uno de los componentes usados en el *template* (*todo-list*, *todo-add* y *todo-del-all*).