

Aplicación de gestión de pedidos React

Enunciado

La idea es manejar un modelo con una estructura con algo de complejidad que al presentarlo en el navegador se tenga que decidir y plantear las mejores alternativas para dividir en componentes y la comunicación entre los mismos.

El modelo planteado es el de un Pedido (usuarios avanzados pueden también gestionar la lista de pedidos).

Las reglas son sencillas:

- El pedido tiene una cabecera con información general sobre el pedido como el número, la fecha y el cliente
- Cada pedido tiene una serie de líneas de pedidos con el artículo que se va a solicitar y el importe del mismo
- Al modificar el importe de alguna línea de pedido el importe total del pedido se actualiza
- Cada línea de pedido podrá ser validada o invalidada (volver al estado pendiente)
- El pedido solo podrá ser enviado si todas las líneas del pedido están validadas
- El estado del pedido indica el porcentaje de líneas que están validadas, por ejemplo un pedido que tiene la mitad de sus líneas validadas estará al 50%

Resumiendo:

- Importe total es un campo calculado.
- Estado depende de si todas las líneas han sido validadas.
- En la tabla tengo un checkbox de selección multiple para validar / invalidar las líneas seleccionadas.
- El campo importe es de edición.
- El botón de enviar solo se habilita si el estado es 100% (todas las líneas validadas).

Un mock de como sería la ventana:

Pedido a proveedor

Número: Proveedor: Fecha:

Importe Total: Estado:

	Estado	Descripción	Importe
<input type="checkbox"/>	Valido	Reactivos maquinaria	2.345€
<input checked="" type="checkbox"/>	Pendiente	Recambios impresión	135€
<input type="checkbox"/>	Pendiente	Soportes plataforma	540€

Pistas

Planteate arrancar por:

- Un contenedor que tenga la información del pedido completo.
- Un componente cabecera:
 - Le pasamos por propiedad:
 - Los datos de cabecera (una entidad número, proveedor, fecha).
 - El importe total y el estado.
 - Un callback para enviar.
 - El calculo si enviar esta deshabilitado o no, podemos decidir si dejarlo en el componente o subirlo al contenedor.
- Un componete detalle del pedido.
 - Le pasamos por propiedad:
 - El listado de detalle del pedido.
 - Callback para actualizar un array de entradas (validar / invalidar)
 - Callback para actualizar una cantidad monetaria (id / valor)
 - Aquí podemos evaluar si el listado de items seleccionados lo tenemos en el container, o si lo encapsulamos en el detalle (si lo subimos al container, lo que es el callback validar / invalidar no le harían falta parametros para informar, el padre ya tiene la información).
 - Para hacer la edición empieza fácil, un map y tira cada fila, también puedes usar una tabla de material ui.

Con que podemos jugar adicionalmente:

- Podrías tener un contexto que wrapeara a esta ventana y no tener que pasar cabecera y detalle para abajo directamente tirar del contexto.

- Podrías romper el componente de cabecera en dos hijos:
 - Una con la info pura de cabecera (a esta le puedes aplicar `React.Memo` y evitar que se repinte, no es necesario para este caso pero es útil para practicar).
 - Otro con el botón de enviar del pedido y el total sumariado.
- También puedes probar a resolver este ejercicio utilizando `use reducer`:
 - En el contenedor tendrías el `reduced`, pasarías el estado para abajo, también el `dispatch`.
 - Vas actualizando toda la info que haga falta en el `reducer`.