**VERSION 01**

{*/\* {listado.map(({ tarea, index }) => (*

*<Box*

*key={index}*

*sx={{*

*display: "flex",*

*padding: "10px",*

*margin: "9px 0",*

*alignItems: "center",*

*border: "1px solid darkgreenn",*

*borderRadius: "5px",*

*backgroundColor: "mediumseagreen",*

*}}*

*>*

*{tarea.name} ( {tarea.hour} {tarea.hour <= 1 ? "hour" : "hours"} )*

*<Box*

*sx={{*

*display: "flex",*

*flexDirection: "row-reverse",*

*flex: "auto",*

*}}*

*>*

*<Button*

*variant="contained"*

*color="error"*

*startIcon={<DeleteIcon></DeleteIcon>}*

*onClick={() => botonDelete(index)}*

*sx={{*

*margin: 3,*

*marginLeft: 0,*

*alignItems: "center",*

*justifyContent: "end",*

*}}*

*></Button>*

*<Button*

*variant="contained"*

*color="primary"*

*startIcon={<EditIcon></EditIcon>}*

*onClick={() => botonEdit(index)}*

*sx={{*

*margin: 3,*

*alignItems: "center",*

*justifyContent: "end",*

*}}*

*></Button>*

*</Box>*

*</Box>*

*))} \*/*}

**VERSION 02**

*/\**

*const [name, setName] = useState("");*

*const [hour, setHour] = useState("");*

*const [activarGuardar, setActivarGuardar] = useState(false);*

*const [tareaSeleccionada, setTareaSeleccionada] = useState({});*

*const [listado, setListado] = useState([]);*

*const [deshabilitarBotonDetele, setDeshabilitarBotonDetele] = useState(false);*

*const inputChangeName = (event) => {*

*setName(event.target.value);*

*};*

*const inputChangeHour = (event) => {*

*setHour(event.target.value);*

*};*

*const botonAdd = () => {*

*if (name.trim() === "" || hour.trim() === "") {*

*props.openYMessage("Debe ingresar: Actividad y Horas Completo");*

*} else if (*

*listado.find(*

*(tarea) => tarea.name === name.trim() && tarea.hour === hour.trim()*

*)*

*) {*

*props.openYMessage("Exite otra actividad y hora idéntica");*

*} else {*

*setListado([...listado, { name: name.trim(), hour: hour }]);*

*setName("");*

*setHour("");*

*}*

*};*

*const botonDelete = (index) => {*

*const newArreglo = [...listado];*

*newArreglo.splice(index, 1);*

*setListado(newArreglo);*

*};*

*const botonEdit = (index) => {*

*setName(listado[index].name);*

*setHour(listado[index].hour);*

*setTareaSeleccionada({*

*name: listado[index].name,*

*hour: listado[index].hour,*

*});*

*setDeshabilitarBotonDetele(true);*

*setActivarGuardar(true);*

*};*

*const botonSave = () => {*

*const indexTareaSeleccionada = listado.findIndex(*

*(tarea) =>*

*tarea.name === tareaSeleccionada.name &&*

*tarea.hour === tareaSeleccionada.hour*

*);*

*if (name.trim() === "" || hour.trim() === "") {*

*props.openYMessage("Debe ingresar: Actividad y Horas Completo");*

*setActivarGuardar(true);*

*} else {*

*const newArreglo = [...listado];*

*newArreglo.splice(indexTareaSeleccionada, 1, {*

*name: name.trim(),*

*hour: hour,*

*});*

*setListado(newArreglo);*

*setName("");*

*setHour("");*

*setActivarGuardar(false);*

*setDeshabilitarBotonDetele(false);*

*}*

*};*

*return (*

*<Box*

*width={"50%"}*

*sx={{*

*padding: "20px",*

*margin: "auto",*

*borderRadius: "10px",*

*backgroundColor: "ghostwhite",*

*}}*

*>*

*<TextField*

*label="Activity"*

*color="primary"*

*focused*

*fullWidth*

*required*

*autoComplete="off"*

*value={name}*

*onChange={inputChangeName}*

*margin="normal"*

*/>*

*<TextField*

*type="number"*

*label="Hours"*

*color="primary"*

*focused*

*fullWidth*

*required*

*autoComplete="off"*

*value={hour}*

*onChange={inputChangeHour}*

*margin="normal"*

*/>*

*{activarGuardar ? (*

*<Button*

*variant="contained"*

*color="success"*

*onClick={() => botonSave()}*

*sx={{*

*margin: "7px",*

*}}*

*>*

*SAVE*

*</Button>*

*) : (*

*<Button*

*variant="contained"*

*onClick={botonAdd}*

*sx={{*

*margin: "7px",*

*}}*

*>*

*ADD*

*</Button>*

*)}*

*{listado.map(({ name, hour }, index) => (*

*<Tarea*

*key={index}*

*name={name}*

*hour={hour}*

*botonDelete={() => botonDelete(index)}*

*botonEdit={() => botonEdit(index)}*

*deshabilitarBotonDetele={*

*tareaSeleccionada.name === name && tareaSeleccionada.hour === hour*

*? deshabilitarBotonDetele*

*: false*

*}*

*></Tarea>*

*))}*

*</Box>*

*);*

*\*/*