**Ejercicios sesiones 13, 14 y 15**

Partiendo del proyecto final, deberéis utilizar crear un formulario con el que crear tareas. Para hacer esto usaréis:

Formik para el formulario.

Yup para las validaciones.

Dentro de Components

**TaskList.jsx**

/\* eslint-disable operator-linebreak \*/

import React from 'react';

import { Formik, Field, Form } from 'formik';

import \* as Yup from 'yup';

import useList from '../../hooks/useList';

import LEVELS from '../../models/levels.enum';

const Tasklist = () => {

const TaskSchema = Yup.object().shape({

name: Yup.string().min(2, 'Too Short!').max(50, 'Too Long!').required('Required'),

description: Yup.string().min(2, 'Too Short!').max(100, 'Too Long!').required('Required'),

});

const defaultTask = {

name: 'Formik',

description: 'Formulario con Formik',

level: LEVELS.URGENT,

done: false,

};

const tasks = useList([defaultTask]);

return (

<div>

<h1>Task List</h1>

<Formik

initialValues={{

name: '',

description: '',

level: LEVELS.NORMAL,

done: false,

}}

onSubmit={(values, actions) => {

setTimeout(() => {

tasks.push(values);

actions.resetForm({});

actions.setSubmitting(false);

}, 500);

}}

validationSchema={TaskSchema}

>

{({ errors }) => (

<Form>

<Field name="name" placeholder="Task Name" />

{errors && errors.name}

<Field name="description" placeholder="Task Description" />

{errors && errors.description}

<Field as="select" name="level">

<option value={LEVELS.NORMAL}>Normal</option>

<option value={LEVELS.URGENT}>Urgent</option>

<option value={LEVELS.BLOCKING}>Blocking</option>

</Field>

<button type="submit">Submit</button>

</Form>

)}

</Formik>

{tasks.isEmpty() ? (

<p>Task List is Empty</p>

) : (

tasks.value.map((task, index) => (

<li key={index} style={{ display: 'flex', alignItems: 'center' }}>

<input type="checkbox" onClick={() => tasks.remove(index)} checked={false} />

<p style={{ fontWeight: 'bold', marginRight: '5px' }}>{task.name}</p>

<p>{task.description}</p>

</li>

))

)}

</div>

);

};

export default Tasklist;

**useList.js //hook**

import { useState } from 'react';

const useList = (initialValue = []) => {

const [value, setValue] = useState(initialValue);

// Push new value to list

const push = (element) => {

setValue((oldValue) => [...oldValue, element]);

};

// Remove value from list

const remove = (index) => {

setValue((oldValue) => oldValue.filter((\_, i) => i !== index));

};

// List is Empty ? true / false

const isEmpty = () => value.length === 0;

// TODO:Develop more functions for lists

return {

value, setValue, push, remove, isEmpty,

};

};

export default useList;

l**evels.enum.js**

const LEVELS = {

NORMAL: 'normal',

URGENT: 'urgent',

BLOCKING: 'blocking',

};

export default LEVELS;