

```
import {Formik, Form, Field } from 'formik'
import * as Yup from 'yup'
import Alerta from './Alerta'
import { useNavigate } from 'react-router-dom'

const Formulario = ({cliente}) => {

  const nuevoClienteSchema= Yup.object().shape({
    nombre: Yup.string().min(3,'El nombre es muy corto')
      .required('El nombre del cliente es requerido'),
    empresa: Yup.string().required('El nombre de la empresa es obligatorio'),
    email: Yup.string().email('Email no válido').required('El email es
obligatorio'),
    telefono: Yup.number().integer('Número no válido').positive('Número no
válido').typeError('No es válido')
  })

  const navigate= useNavigate()

  const handleSubmit=async(valores)=>{
    try {
      if(cliente.id){
        const url=`http://localhost:4000/clientes/${cliente.id}`

        const respuesta = await fetch(url, {
          method: 'PUT',
          body: JSON.stringify(valores),
          headers:{
            'Content-Type':'application/json'
          }
        })
        const resultado = await respuesta.json()
        navigate('/clientes')

      }else{
        const url="http://localhost:4000/clientes"

        const respuesta = await fetch(url, {
          method: 'POST',
          body: JSON.stringify(valores),
          headers:{
            'Content-Type':'application/json'
          }
        })
        const resultado = await respuesta.json()
        navigate('/clientes')

      }
    } catch (error) {
      console.log(error)
    }
  }
}
```

```

return (
  <div className="bg-white px-5 py-10 mt-10 rounded-md shadow-md md:w-3/4 mx-
auto">

    <h1 className="text-gray-600 font-bold text-3xl uppercase text-center">
      {cliente?.nombre ? "Editar Cliente": "Agregar Cliente"}
    </h1>

    <Formik
      initialValues={{
        nombre: cliente?.nombre ?? "",
        empresa: cliente?.empresa ?? "",
        email: cliente?.email ?? "",
        telefono: cliente?.telefono ?? "",
        notas: cliente?.notas ?? ""
      }}
      enableReinitialize={true}
      onSubmit={async(values, {resetForm})=>{
        await handleSubmit(values)

        resetForm();
      }}
      validationSchema={nuevoClienteSchema}
    >
      ({({errors,touched})=>{
        return (
          <Form className="mt-10">
            <div className="mb-4">
              <label
                className='text-gray-800'
                htmlFor='nombre'>Nombre:</label>
              <Field
                id="nombre"
                type="text"
                className="mt-2 block w-full p-3 bg-gray-100"
                placeholder="Nombre del cliente"
                name="nombre"/>
              {errors.nombre && touched.nombre? (
                <Alerta>{errors.nombre}</Alerta>
              ): null}
            </div>
            <div className="mb-4">
              <label
                className='text-gray-800'
                htmlFor='empresa'>Empresa Cliente:</label>
              <Field
                id="empresa"
                type="text"
                className="mt-2 block w-full p-3 bg-gray-100"
                placeholder="Nombre de la empresa"
                name="empresa"/>

```

```

        {errors.empresa && touched.empresa ? (
          <Alerta>{errors.empresa}</Alerta>
        ): null}
      </div>
      <div className="mb-4">
        <label
          className='text-gray-800'
          htmlFor='email'>Email</label>
        <Field
          id="email"
          type="email"
          className="mt-2 block w-full p-3 bg-gray-100"
          placeholder="Email del Cliente"
          name="email"/>
        {errors.email && touched.email ? (
          <Alerta>{errors.email}</Alerta>
        ): null}

      </div>
      <div className="mb-4">
        <label
          className='text-gray-800'
          htmlFor='telefono'>Teléfono</label>
        <Field
          id="telefono"
          type="tel"
          className="mt-2 block w-full p-3 bg-gray-100"
          placeholder="Teléfono del Cliente"
          name="telefono"/>
        {errors.telefono && touched.telefono ? (
          <Alerta>{errors.telefono}</Alerta>
        ): null}

      </div>
      <div className="mb-4">
        <label
          className='text-gray-800'
          htmlFor='notas'>Notas</label>
        <Field
          as="textarea"
          id="notas"
          type="text"
          className="mt-2 block w-full p-3 bg-gray-100 h-20"
          placeholder="Notas"
          name="notas"/>
      </div>
      <input
        type="submit"
        value={cliente.nombre? "Editar Cliente": "Agregar Cliente"}
        className="mt-5 w-full bg-blue-800 p-3 text-white uppercase font-bold
text-lg"/>
    </Form>
  )}}

```

```
        </Formik>

      </div>
    )
  }
  Formulario.defaultProps = {
    cliente: {}
  }
  export default Formulario
```