Validando el Formulario

- La API requiere ambos datos, la moneda para hacer la conversión y la criptomoneda
- Coloco un handleSubmit en el onSubmit del form
- Creo la función y le añado e.preventDefault()
- Tengo a disposición moneda y criptomoneda de la desestructuración del state del customHook
- Uso el if y el includes con un string vacio para hacer la validación
- Dónde pone alert va a ser un componente que muestre un mensaje de error

```
const handleSubmit =(e)=>{
  e.preventDefault()
  if([monedas, criptomoneda].includes('')){
    alert('Error')
  }return
}
```

- Creo un nuevo state en Formulario.jsx con error, setError y lo inicio en False
- Entonces ahi le coloco el setError(true)
- En el return le coloco si error es true entonces un párrafo

```
return (
  <>
    {error && <p>Todos los campos son obligatorios</p>}

    <form onSubmit={handleSubmit}>
      <SelectMonedas/>
      <SelectCriptoMoneda/>



        <InputSubmit type="sumbit"
        value="cotizar" />
    </form>


  </>
)
```

- Creo un componente que voy a llamarlo Error

```
import React from 'react'
import styled from '@emotion/styled'

const Texto= styled.div`
    background-color: red;
    color: #FFF;
    padding: 15px;
    font-size: 22px;
    text-transform: uppercase;
    text-align: center;

`
const Error = ({children}) => {
  return (
    <Texto>{children}</Texto>
  )
}

export default Error
```

- Le pongo el contenido del párrafo en el componente para que vayan como children y se coloquen dentro del componente

```
{error && <Error>Todos los campos son obligatorios</Error>}
```

Ahora debe ocultarlo cuando la validación sea correcta

```
  const handleSubmit =(e)=>{
    e.preventDefault()
    if([moneda, criptomoneda].includes('')){
      setError(true)
      return
    }

    setError(false)
  }
```

- Hay que pasar moneda y criptomoneda a App.jsx porque debajo van a mostrarse los resultados
- Va a haber un segundo llamado a la Api para la conversión de moneda a cripto