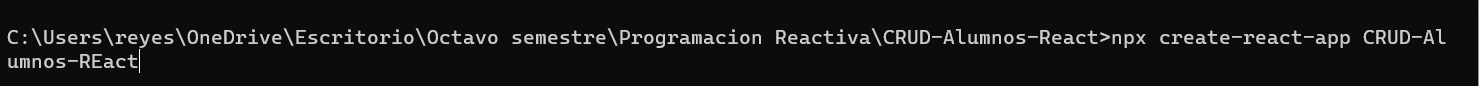
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nombre de la práctica** | **CRUD desde react usando estados** | | | **No.** | **1** |
| **Asignatura:** | **Programación Reactiva** | **Carrera:** | **Ingeniería en Sistemas Computacionales** | **Duración de la práctica (Hrs)** | **2 horas** |

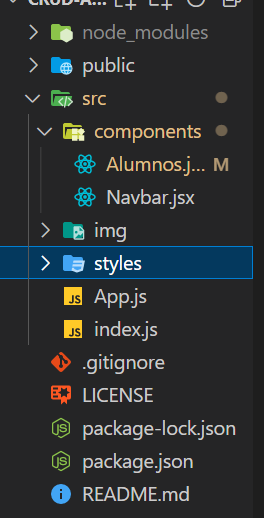
**Integrantes:**

**Brandon Lee Reyes Moreno**

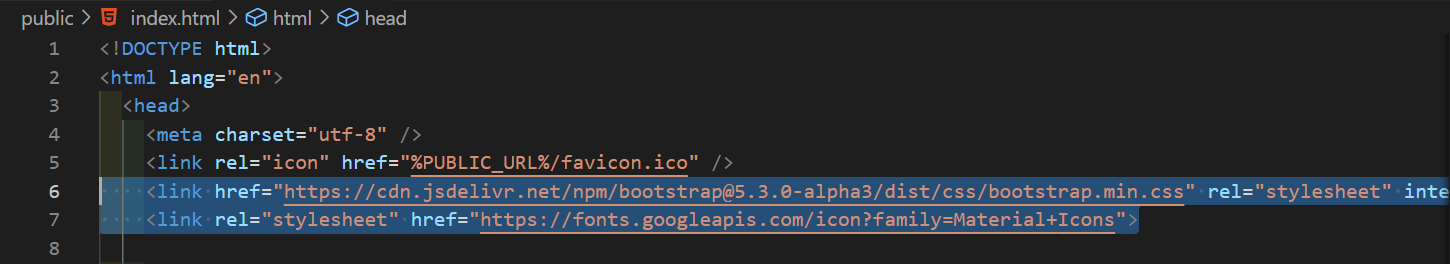
1. **Comenzamos creando nuestro proyecto de react**

****

1. **Limpiamos nuestro proyecto quitándole los css y los logos y js que sobran para esto y agregamos una carpeta para ahí agregar nuestros componentes, otra para agregar nuestro estilo y otra para guardar nuestras imágenes**

****

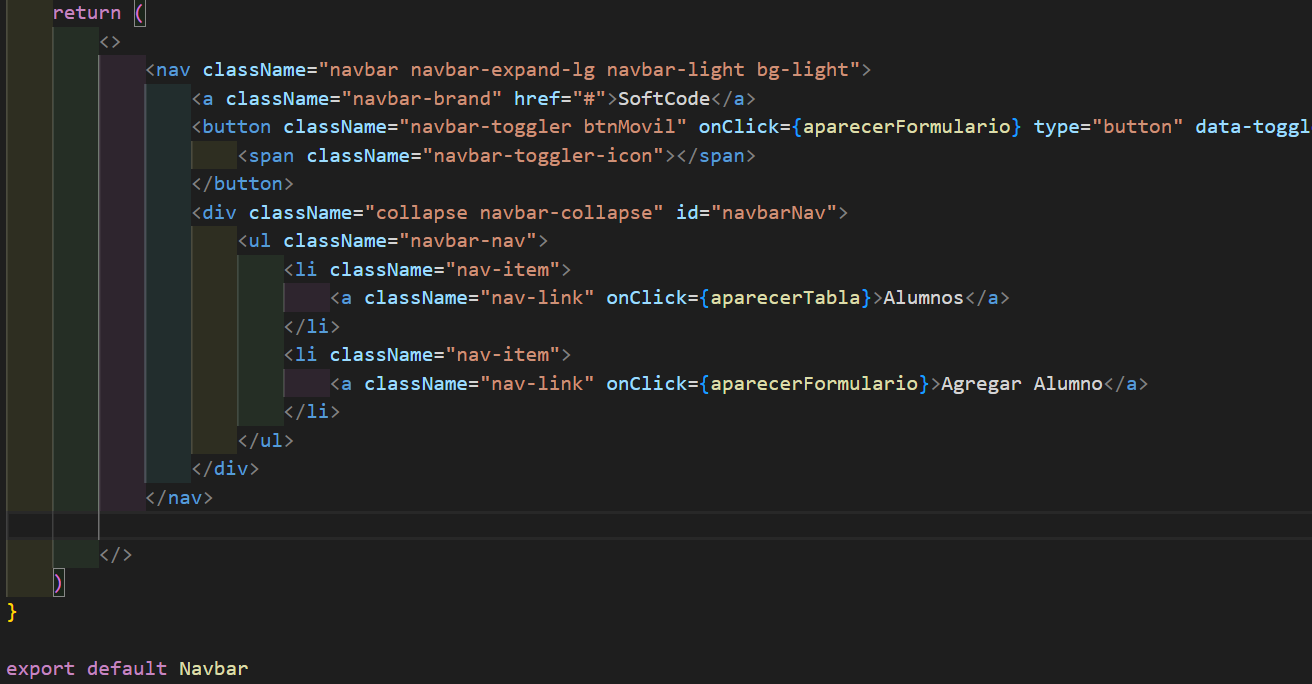
1. **Agregamos en nuestro índex el link para Bootstrap y así poner descarar estilos y también el de Google para algunos iconos**



1. **Comenzamos agregando nuestro componente del Navbar que utilizaremos para el menú principal de nuestra aplicación primero ponemos las funciones para que al darle clic a uno desaparezca el otro y viseversa**

****

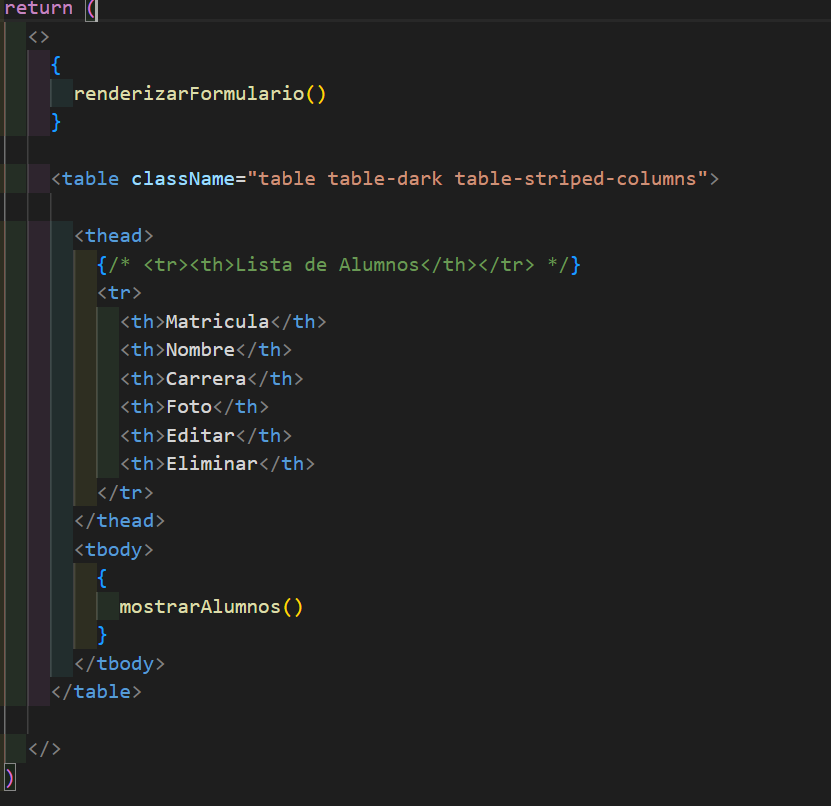
1. **Después colocamos nuestro HTML con botones que llamen a las funciones y lo exportamos**

****

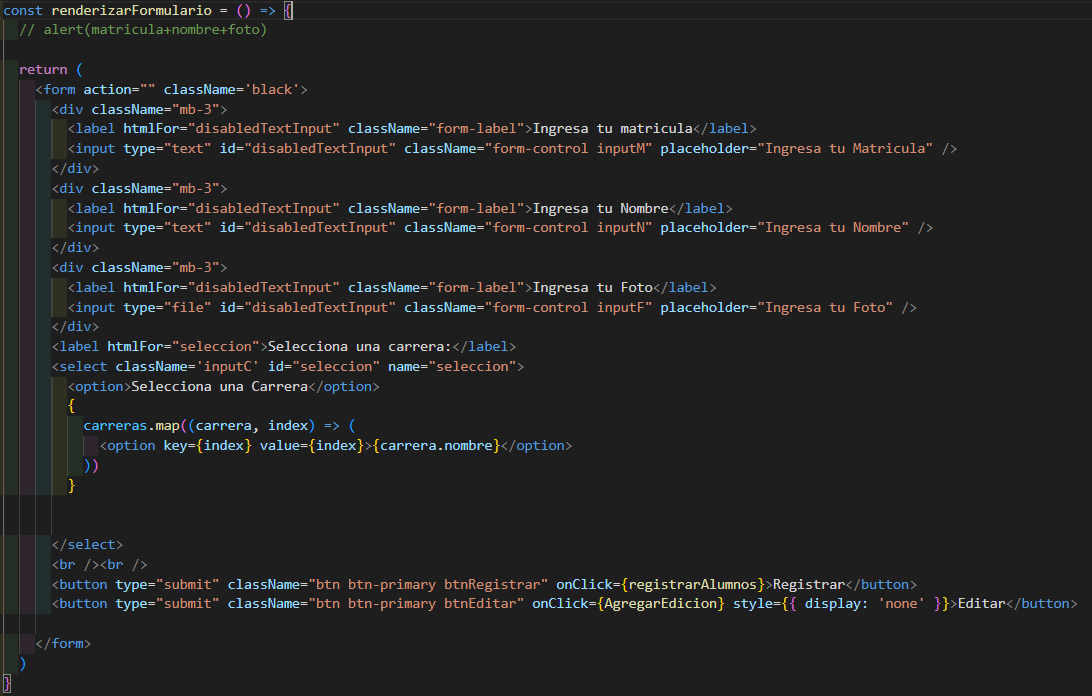
1. **Después crearemos nuestro siguiente componente donde se encontrará primero una lista de alumnos y carreras ya definidas para después poderla modificar o agregar**

****

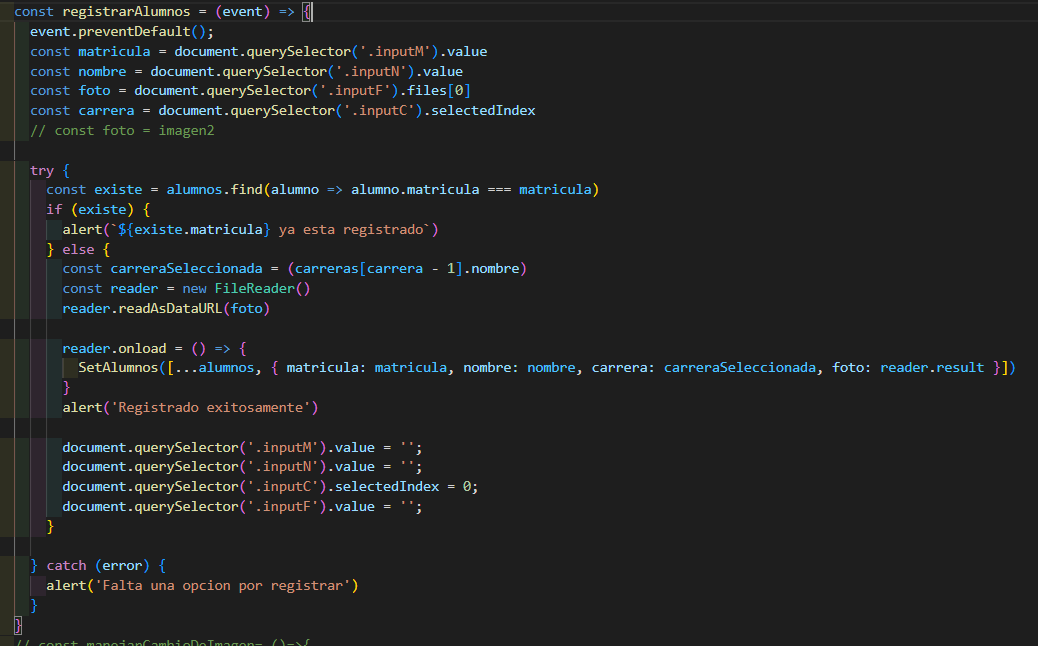
1. **Lo que va a retornar nuestro componente es la siguiente estructura HTML con llamada a funciones**

****

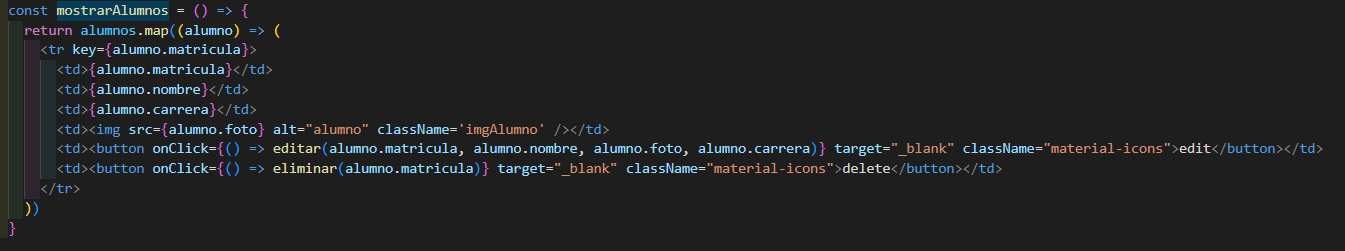
1. **Al llamar a la función de renderizar de igual manera nos regresa nuestro HTML de mi formulario con llamadas a funciones como para mostrar las carreras, y al dale al botón de registrar y aditar aparece cuando le des en botón editar en nuestra tabla**

****

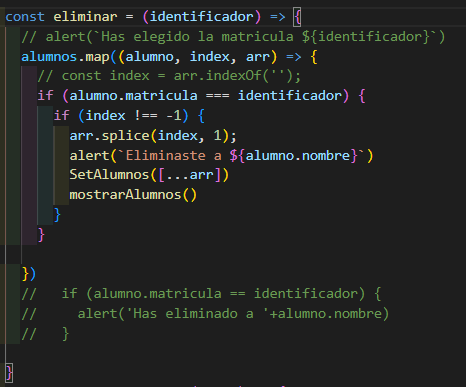
1. **Al darle en registrar nos mandara a la siguiente función donde validara que todos los campos estén llenos para poder registrarse, si llega a faltar algún campo o algo anda mal te retorna lo que te hace falta y cuando se logre registrar y todo salga bien se limpiaran nuestros campos para dejarlos en blanco**

****

1. **Al retornar nuestro componente en la función de mostrar Alumnos nos retorna una tabla llena con nuestra lista de alumnos ordenada y con cada usuario nos mostrara un botón de eliminar y otro de editar**

****

1. **Al darle clic en nuestro botón de eliminar de nuestro alumno, nos manda como parámetro la matricula del alumno para saber que podemos eliminar. E identificándolo buscamos su posición el arreglo a eliminar, y desde donde hasta donde eliminarlo en este caso solo es uno y se actualiza nuestra tabla mandando llamar de nuevo a nuestra función de mostrar alumnos a nuestra Tabla.**

****

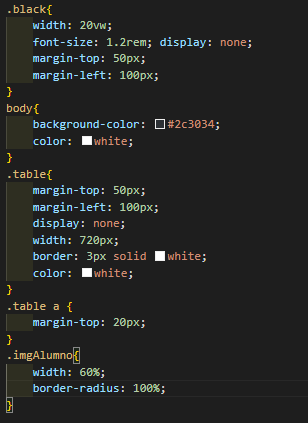
1. **Al darle al botón de editar de cada alumno nos retorna todos sus valores los cuales utilizaremos para rellenar los campos y que el usuario lo pueda modificar**

****

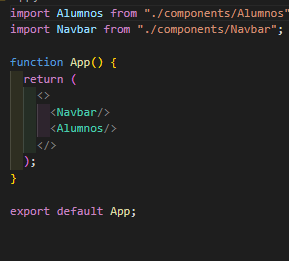
1. **Después de llenarlo y que nos aparecerá un botón de editar para confirmar los cambios y guardarlos y con ellos nos llevara a la función de agregar edición en la cual obtenemos todos los campos que se ingresaron, después lo buscamos en el arreglo para identificar el que vamos a modificar, obtenemos la posición de este, después creamos un nuevo arreglo donde colocamos estos datos y al final en nuestro array inicial lo colocamos en la posición en la cual estaba el de editar y después se limpian nuestros datos**

****

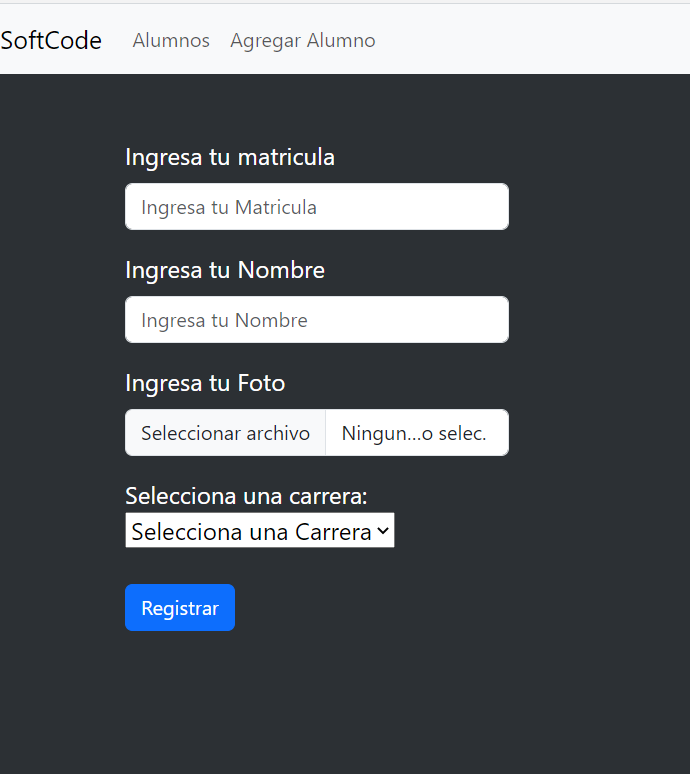
1. **Nuestros estilos son muy mus pocos ya que la mayoría fue con la utilización de Bootstrap y este css es solo para colores, tamaños margen y algunos display donde al inicio aparecerán ocultos**

****

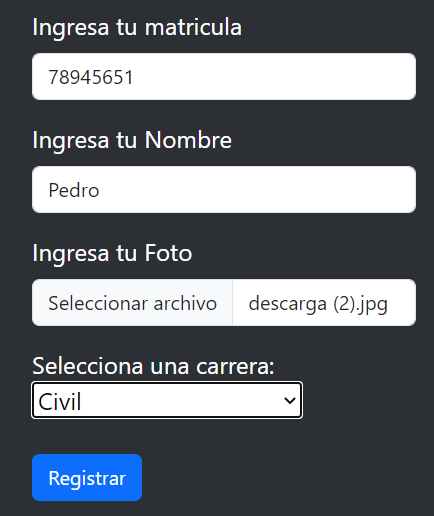
1. **Al final importaremos nuestros componentes en nuestro app.js para unir nuestros componentes**

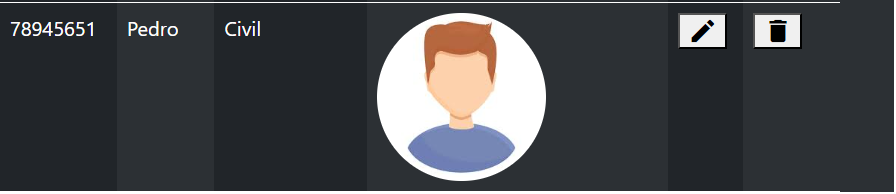
****

1. **Empezando con las pruebas, nuestra vista principal seria la siguiente**

****

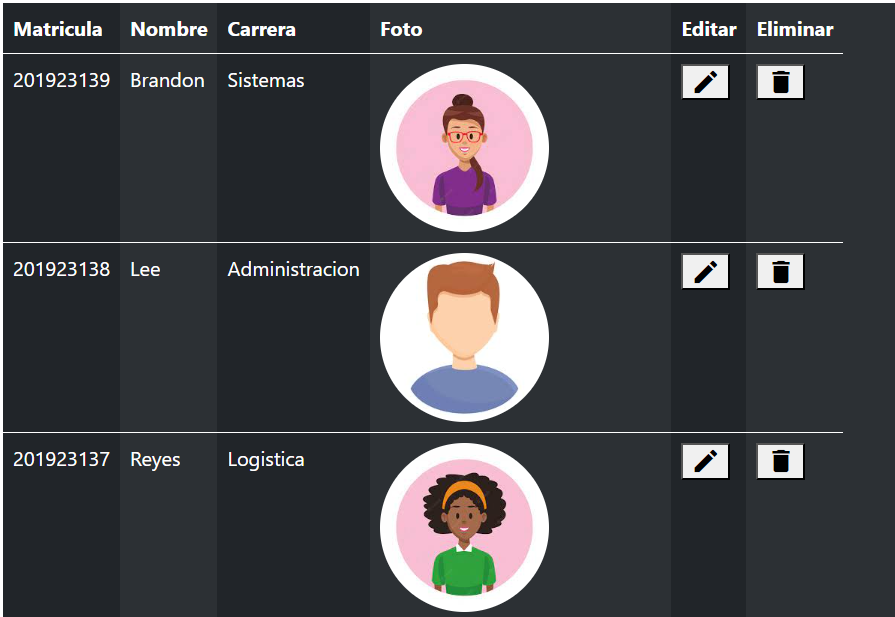
1. **Al agregar todos los datos necesarios y darle en registrar se nos mostrara en la tabla con los datos colocados**

**** ****

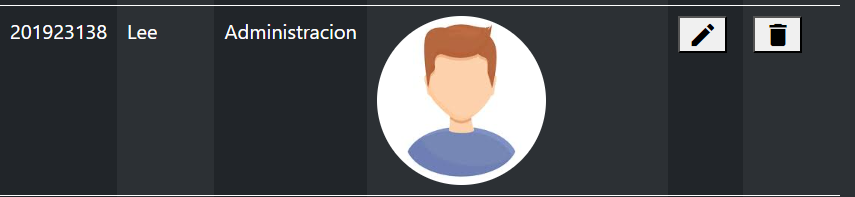
****

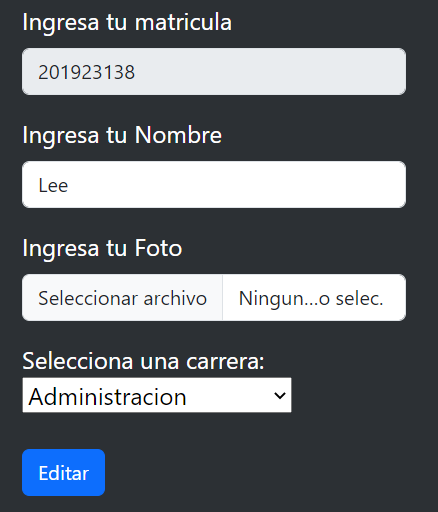
1. **Al darle en el botón de liminar de nuestro usuario solo desaparecerá de la lista**

****

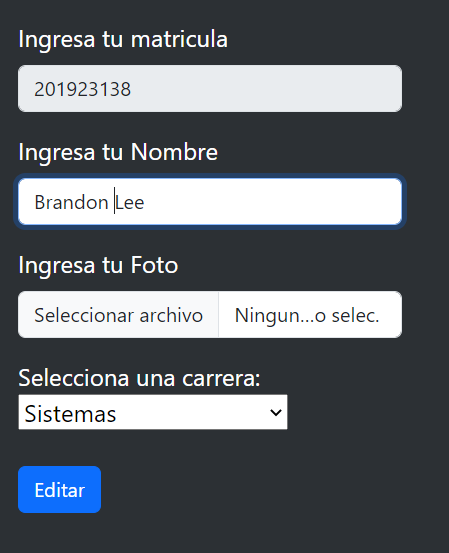
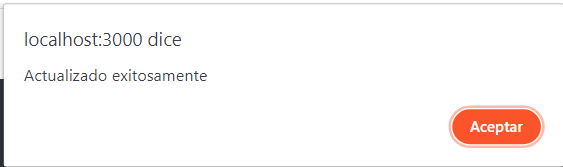
****

1. **Al darle en nuestro botón de editar nos mandara nuestro formulario lleno con los datos guardados y bloqueada la matricula**

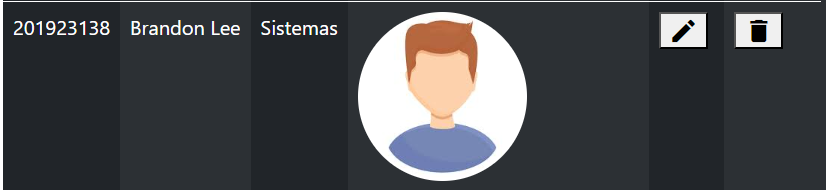
****

****

1. **Después cambiamos los datos que queramos actualizar y le damos en editar**

**** ****

1. **Y en nuestra tabla nos aparecerán los campos ya actualizados**



**CONCLUSION**

**Nuestras conclusiones sobre esta práctica que es un CRUD desde react es muy útil al utilizar los estados, ya que se pueden actualizar dependiendo las necesidades del cliente aunque para este tipo de practica a mi parecer no es tan útil ya que al volver a iniciar nuestro proyecto ya no existirá y no se guardara ningún valor que se haya registrado anteriormente, esto ya seria con un desarrollo más avanzado con una base que registre todo lo que es usuario cree, elimine o edite sin afectarlo en un futuro.**