

INSTITUTO TECNOLÓGICO DE ESTUDIOS SUPERIORES DE ZAMORA

MAESTRÍA EN SISTEMAS COMPUTACIONALES

ARQUITECTURA ORIENTADA A SERVICIOS

ClienteServidorWeb

**“MANUAL DE INSTALACIÓN Y PUESTA EN MARCHA
DE LA APLICACIÓN ”**

FACILITADOR

M.I.S. ROBERTO SUÁREZ ZINZÚN

SUSTENTAN

ZAMORA

**ING. LUIS FERNANDO NAREZ RUIZ
L.I.A. JUAN CARLOS HERNANDEZ BARRERA**

ZAMORA, MICHOACÁN A 24 DE NOVIEMBRE DEL 2020

1.- Repositorio de la aplicación de React.

https://github.com/charly11001/aplicacion_acualizada_JCHB.git

2.- Repositorio de la API REST en loopback.

https://github.com/charly11001/loopback_actualizado_JCHB.git

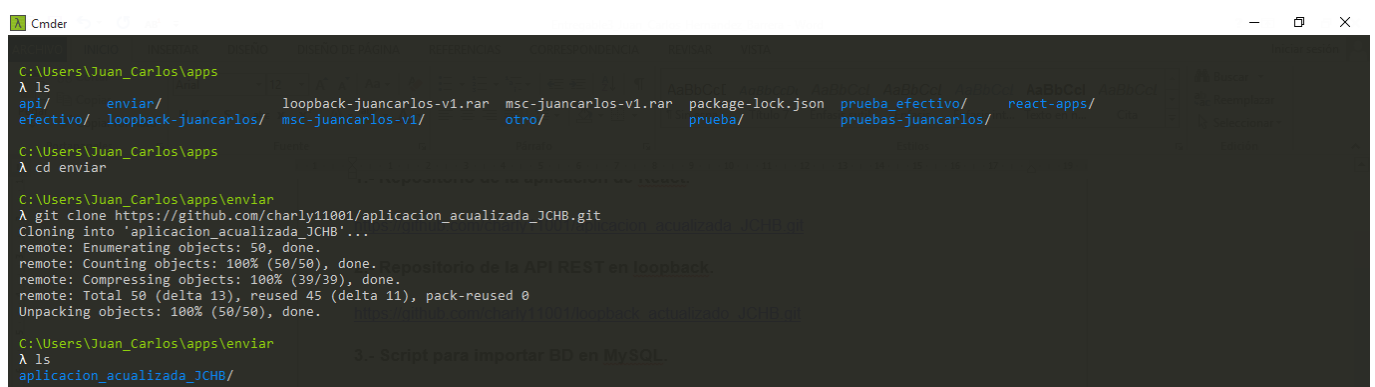
3.- Script para importar BD en MySQL.

Se encuentra dentro de la carpeta data con el nombre de **miscursosos.sql** que a su vez se encuentra dentro de la aplicación React en el repositorio que se mencionó en el punto número 1.

4.- Procedimiento para importar la aplicación.

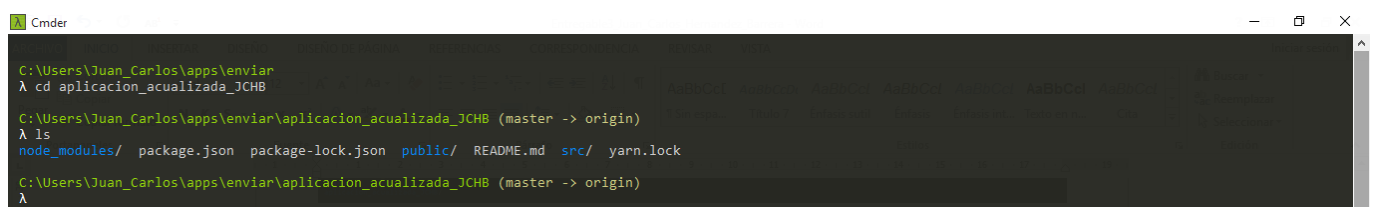
a) Importación y puesta en punto de la aplicación de React

Primero hay que ingresar a la carpeta donde se desea clonar la aplicación de react a través del repositorio compartido en Github, llevar a cabo la clonación escribiendo el siguiente comando: **git clone https://github.com/charly11001/aplicacion_acualizada_JCHB.git**, como se muestra a continuación.



```
C:\Users\Juan_Carlos> cd enviar/
C:\Users\Juan_Carlos> git clone https://github.com/charly11001/aplicacion_acualizada_JCHB.git
Cloning into 'aplicacion_acualizada_JCHB'...
remote: Enumerating objects: 50, done.
remote: Counting objects: 100% (50/50), done.
remote: Compressing objects: 100% (39/39), done.
remote: Total 50 (delta 13), reused 45 (delta 11), pack-reused 0
Unpacking objects: 100% (50/50), done.
```

Posteriormente, ingresar a la carpeta que se creó (la cual contiene todas las carpetas y archivos de nuestra aplicación), como se muestra en la siguiente imagen.



```
C:\Users\Juan_Carlos> cd aplicacion_acualizada_JCHB
C:\Users\Juan_Carlos> ls
node_modules/ package.json package-lock.json public/ README.md src/ yarn.lock
```

Después hay que instalar la aplicación con el comando **npm install**, como se muestra a continuación.

```
C:\Users\Juan_Carlos\apps\enviar\aplicacion_acualizada_JCHB (master -> origin)
λ npm install

> core-js@2.6.11 postinstall C:\Users\Juan_Carlos\apps\enviar\aplicacion_acualizada_JCHB\node_modules\babel-runtime\node_modules\core-js
> node -e "try(require('..postinstall')){catch(e){}}"

Thank you for using core-js ( https://github.com/zloirock/core-js ) for polyfilling JavaScript standard library!

The project needs your help! Please consider supporting of core-js on Open Collective or Patreon:
> https://opencollective.com/core-js
> https://www.patreon.com/zloirock

Also, the author of core-js ( https://github.com/zloirock ) is looking for a good job -)

> core-js@3.6.5 postinstall C:\Users\Juan_Carlos\apps\enviar\aplicacion_acualizada_JCHB\node_modules\core-js
> node -e "try(require('..postinstall')){catch(e){}}"

> core-js-pure@3.6.5 postinstall C:\Users\Juan_Carlos\apps\enviar\aplicacion_acualizada_JCHB\node_modules\core-js-pure
> node -e "try(require('..postinstall')){catch(e){}}"

npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.13 (node_modules\webpack-dev-server\node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.13: wanted {"os":"darwin","arch":"any"} (current: {"os":"win32","arch":"x64"})
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.13 (node_modules\watchpack-chokidar2\node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.13: wanted {"os":"darwin","arch":"any"} (current: {"os":"win32","arch":"x64"})
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.13 (node_modules\jest-haste-map\node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.13: wanted {"os":"darwin","arch":"any"} (current: {"os":"win32","arch":"x64"})
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@2.1.2 (node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@2.1.2: wanted {"os":"darwin","arch":"any"} (current: {"os":"win32","arch":"x64"})

added 1679 packages from 808 contributors and audited 1683 packages in 303.42s

64 packages are looking for funding
  run `npm fund` for details

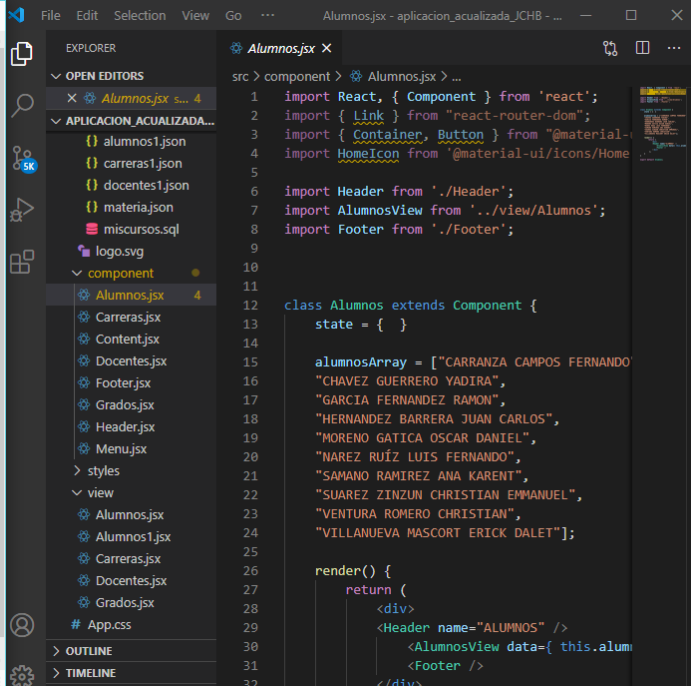
found 4982 low severity vulnerabilities
  run `npm audit fix` to fix them, or `npm audit` for details

C:\Users\Juan_Carlos\apps\enviar\aplicacion_acualizada_JCHB (master -> origin)
λ |
```

Después hay que escribir el comando **code .** para ingresar a Visual Studio Code y poder ver el código fuente de nuestra aplicación, como se muestra a continuación.

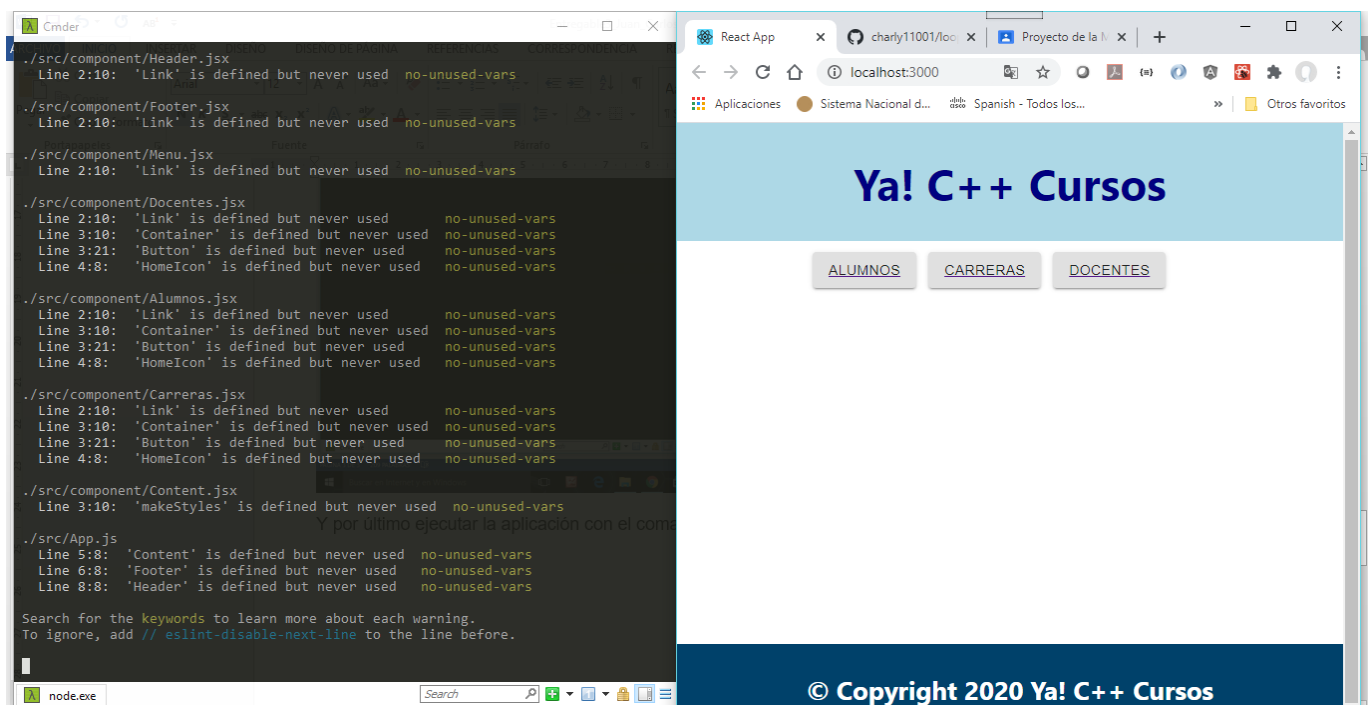
```
C:\Users\Juan_Carlos\apps\enviar\aplicacion_acualizada_JCHB (master -> origin)
λ code .

C:\Users\Juan_Carlos\apps\enviar\aplicacion_acualizada_JCHB (master -> origin)
λ |
```



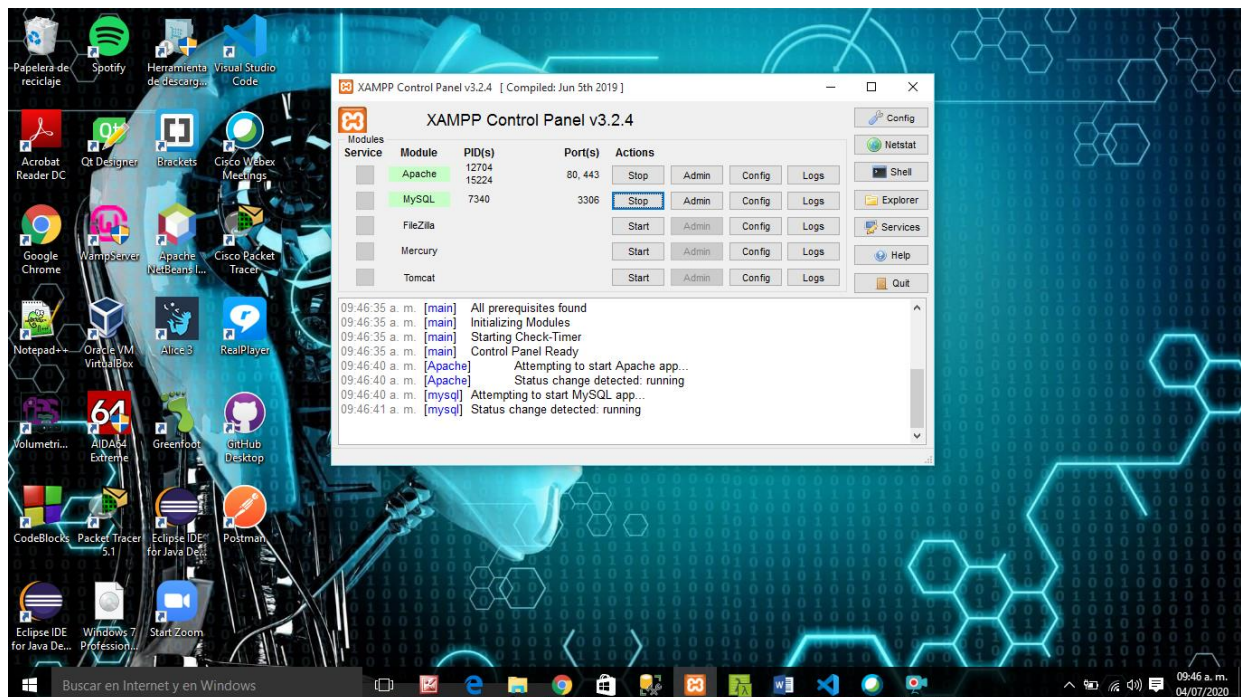
The screenshot shows the Visual Studio Code interface. On the left, the Explorer sidebar displays the project structure with folders like 'component' and 'view', and files like 'Alumnos.jsx', 'Carreras.jsx', 'Content.jsx', 'Docentes.jsx', 'Footer.jsx', 'Grados.jsx', 'Header.jsx', 'Menu.jsx', 'styles', 'App.css', 'Alumnos1.jsx', 'Carreras1.jsx', 'Docentes1.jsx', 'Grados1.jsx', and 'NPM SCRIPTS'. The main editor area shows the code for 'Alumnos.jsx', which includes imports for React, Component, Link, Container, Button, HomeIcon, and Header. It defines a class 'Alumnos' extending 'Component' with a state object and a render method that returns a JSX element with a header, a list of students, and a footer.

Y por último ejecutar la aplicación con el comando **npm start**.

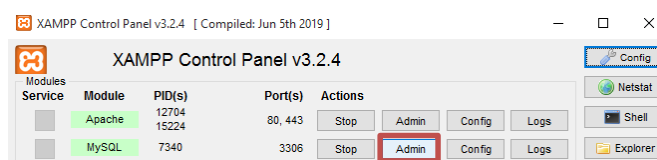


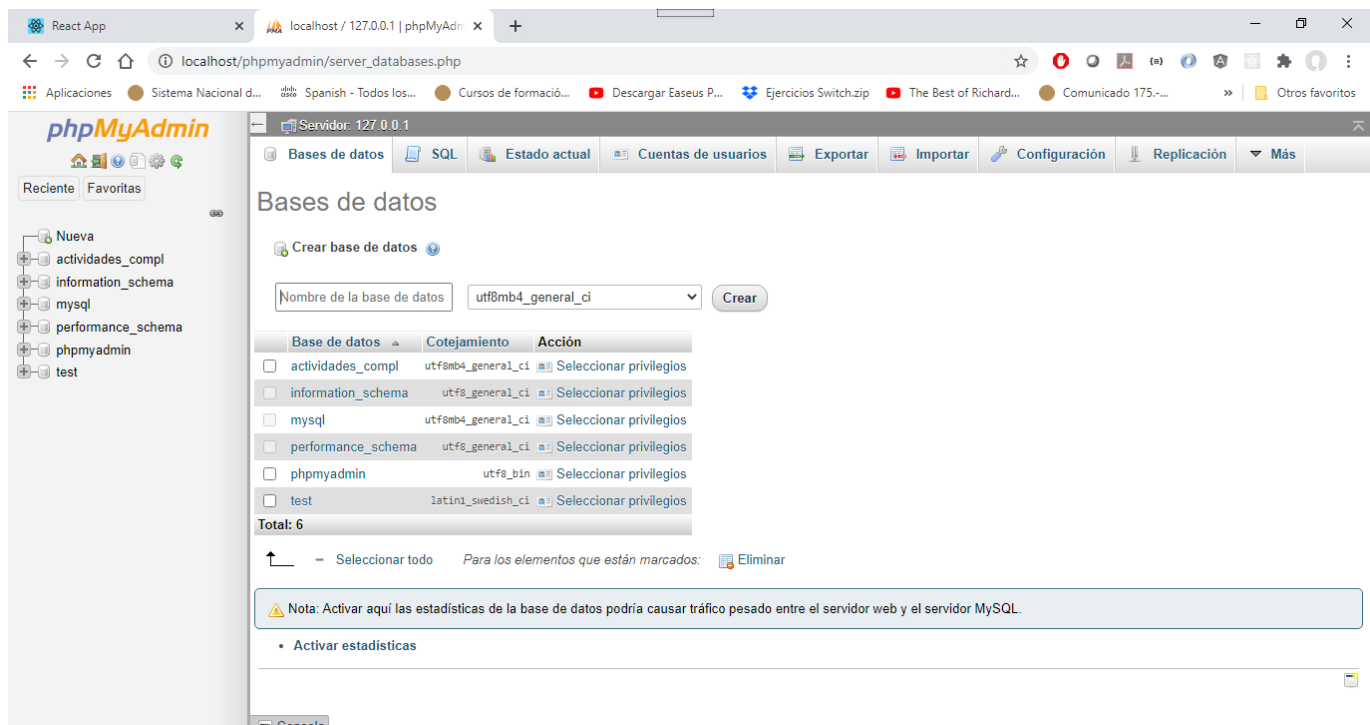
b) Importación de base de datos en MySQL.

Abrir el programa Xampp e inicializar los servicios de Apache y MySQL.

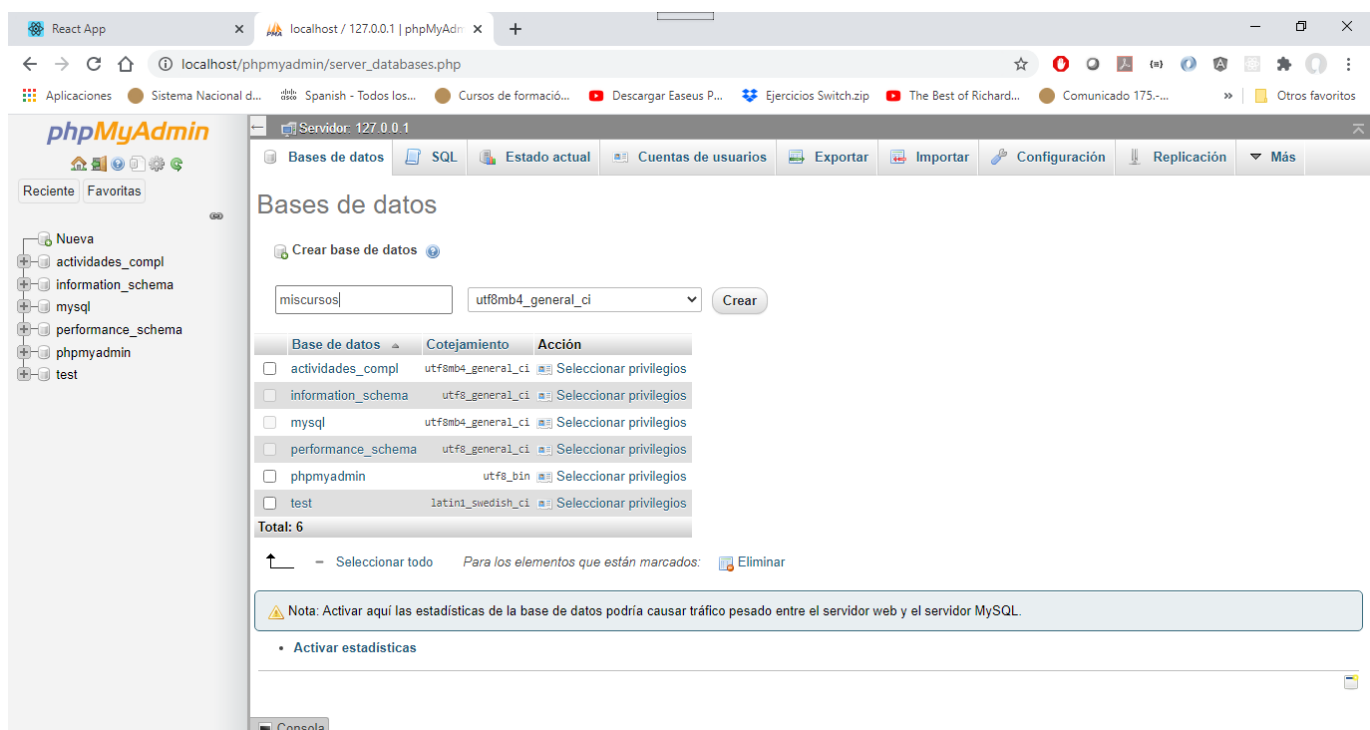


En el servicio de MySQL, hay que darle un clic al botón de Admin para abrir el phpMyAdmin.

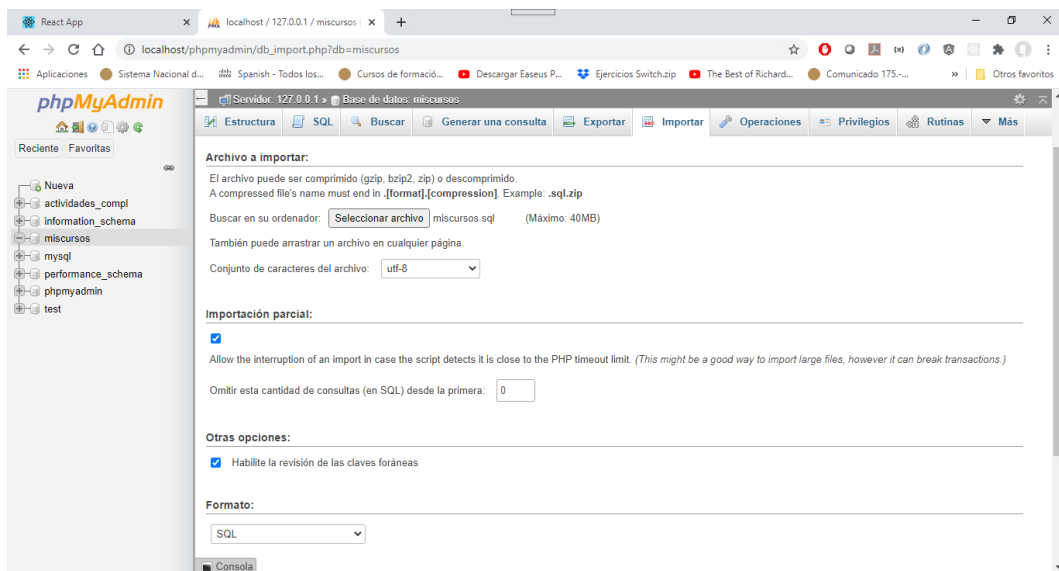




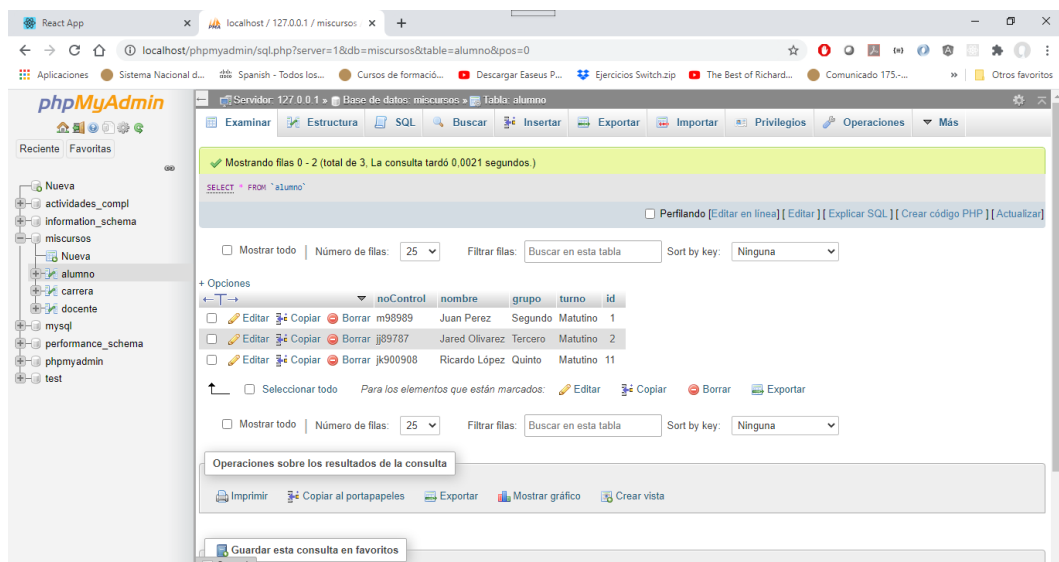
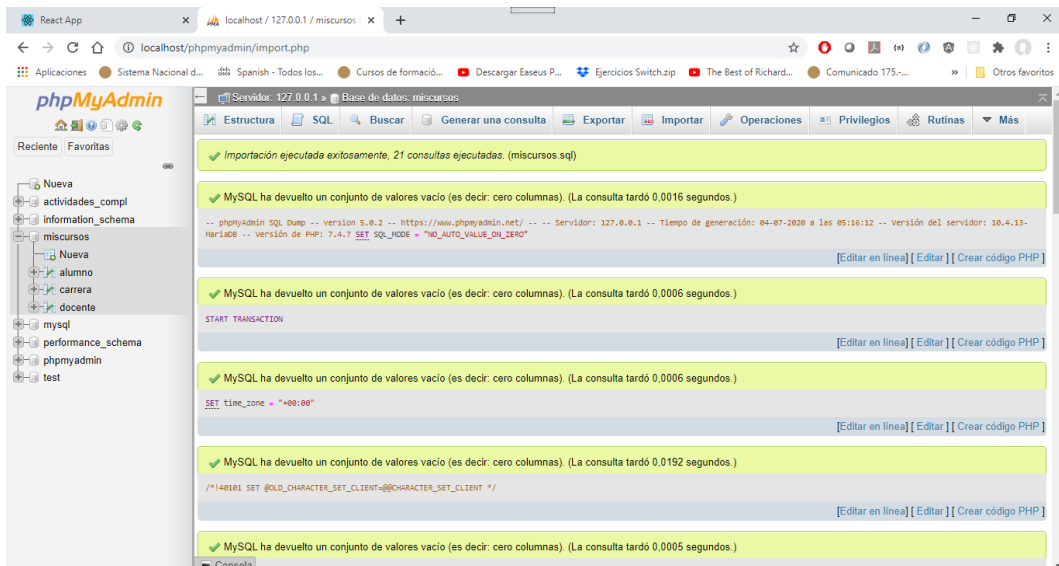
Posteriormente hay que crear la base de datos, escribiendo el nombre de la base de datos (en este caso **miscursos**) y presionar el botón de crear.



Después dar un clic en el botón de importar, seleccionar el archivo que se desea importar (en este caso, se llama **miscursos.sql**), elegir el formato (SQL) y presionar el botón de continuar.

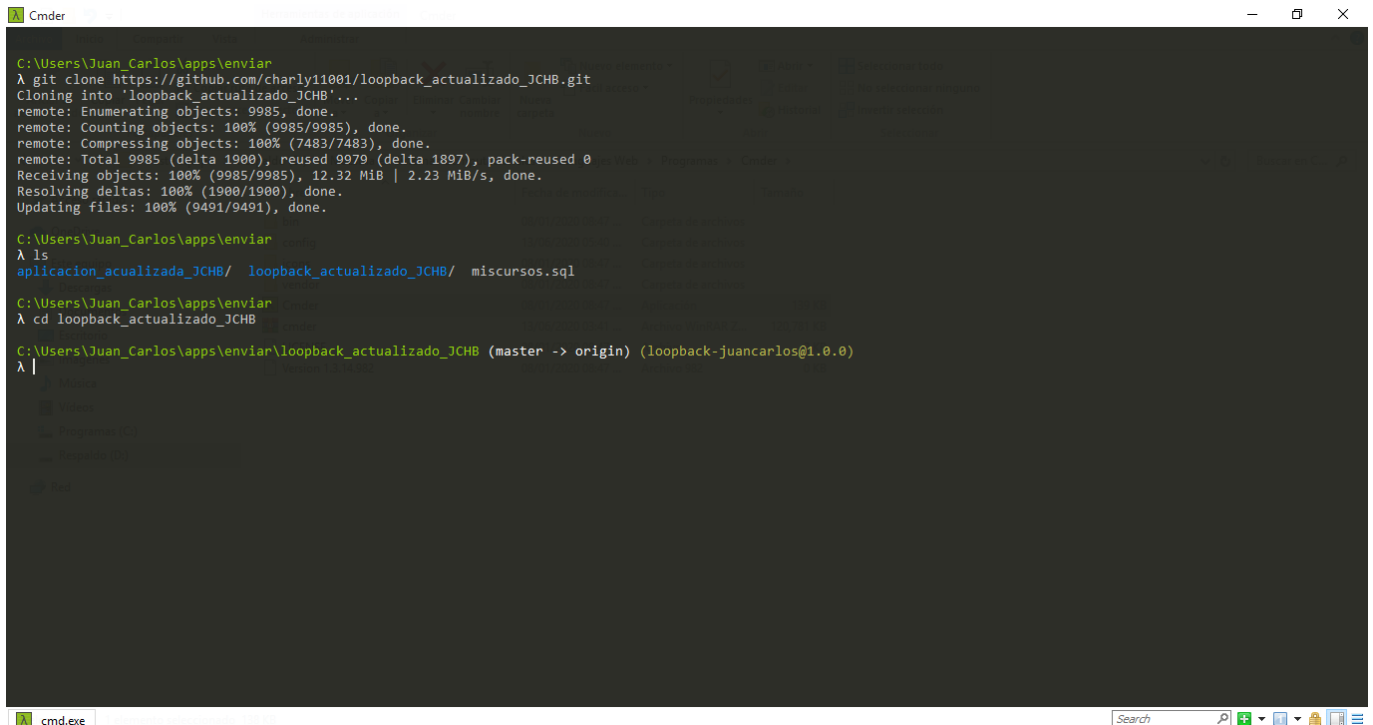


Con esta acción se habrán importado todas las tablas (incluso sus registros) de la base de datos llamada **miscursos**, como se muestra en la siguiente imagen.



c) Importación y puesta en punto de aplicación de Loopback

Primero hay que ingresar a la carpeta donde se desea clonar la aplicación de react a través del repositorio compartido en Github, llevar a cabo la clonación escribiendo el siguiente comando: **git clone https://github.com/charly11001/loopback_actualizado_JCHB.git**, después ingresar a la carpeta que se creó, como se muestra a continuación.



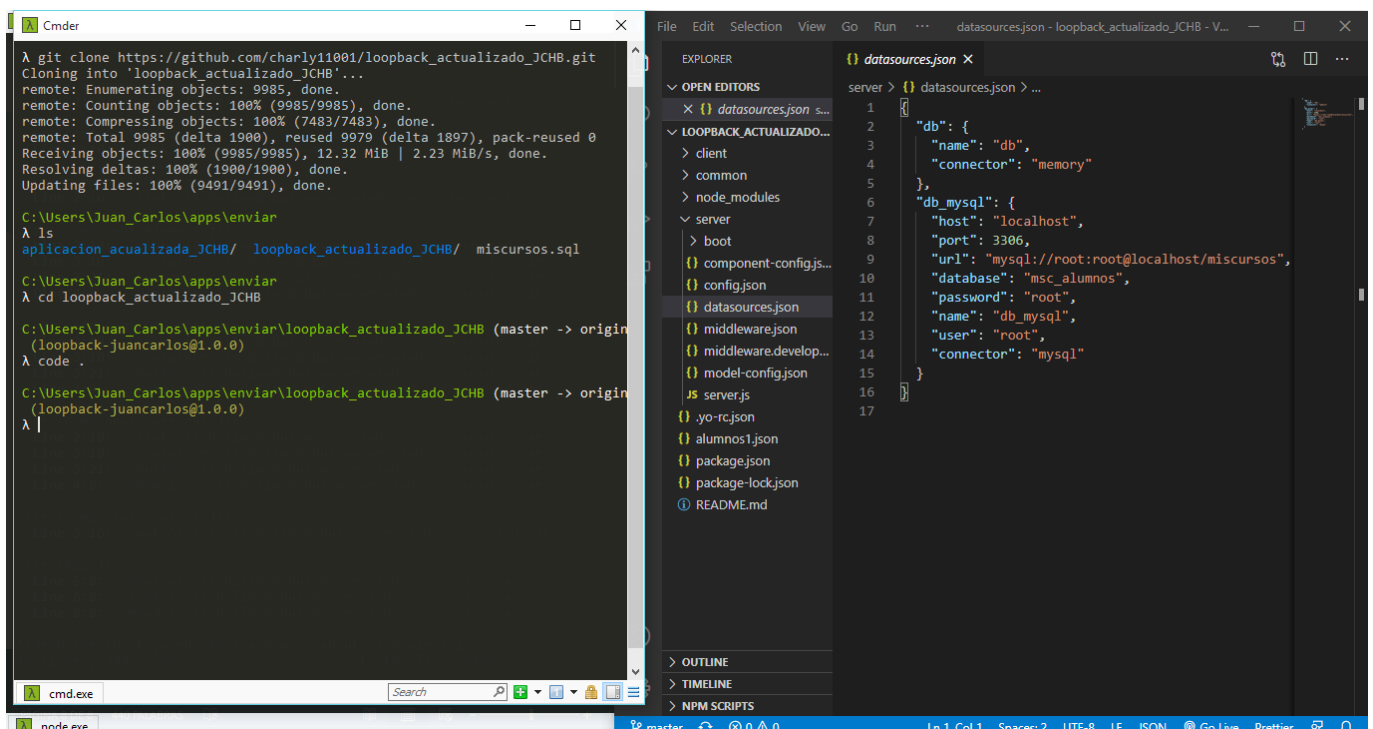
```
C:\Users\Juan_Carlos\apps\enviar
λ git clone https://github.com/charly11001/loopback_actualizado_JCHB.git
Cloning into 'loopback_actualizado_JCHB'...
remote: Enumerating objects: 9985, done.
remote: Counting objects: 100% (9985/9985), done.
remote: Compressing objects: 100% (7483/7483), done.
remote: Total 9985 (delta 1900), reused 9979 (delta 1897), pack-reused 0
Receiving objects: 100% (9985/9985), 12.32 MiB | 2.23 MiB/s, done.
Resolving deltas: 100% (1900/1900), done.
Updating files: 100% (9491/9491), done.

C:\Users\Juan_Carlos\apps\enviar
λ ls
aplicacion_acualizada_JCHB/  loopback_actualizado_JCHB/  miscursos.sql

C:\Users\Juan_Carlos\apps\enviar
λ cd loopback_actualizado_JCHB

C:\Users\Juan_Carlos\apps\enviar\loopback_actualizado_JCHB (master -> origin) (loopback-juancarlos@1.0.0)
λ |
```

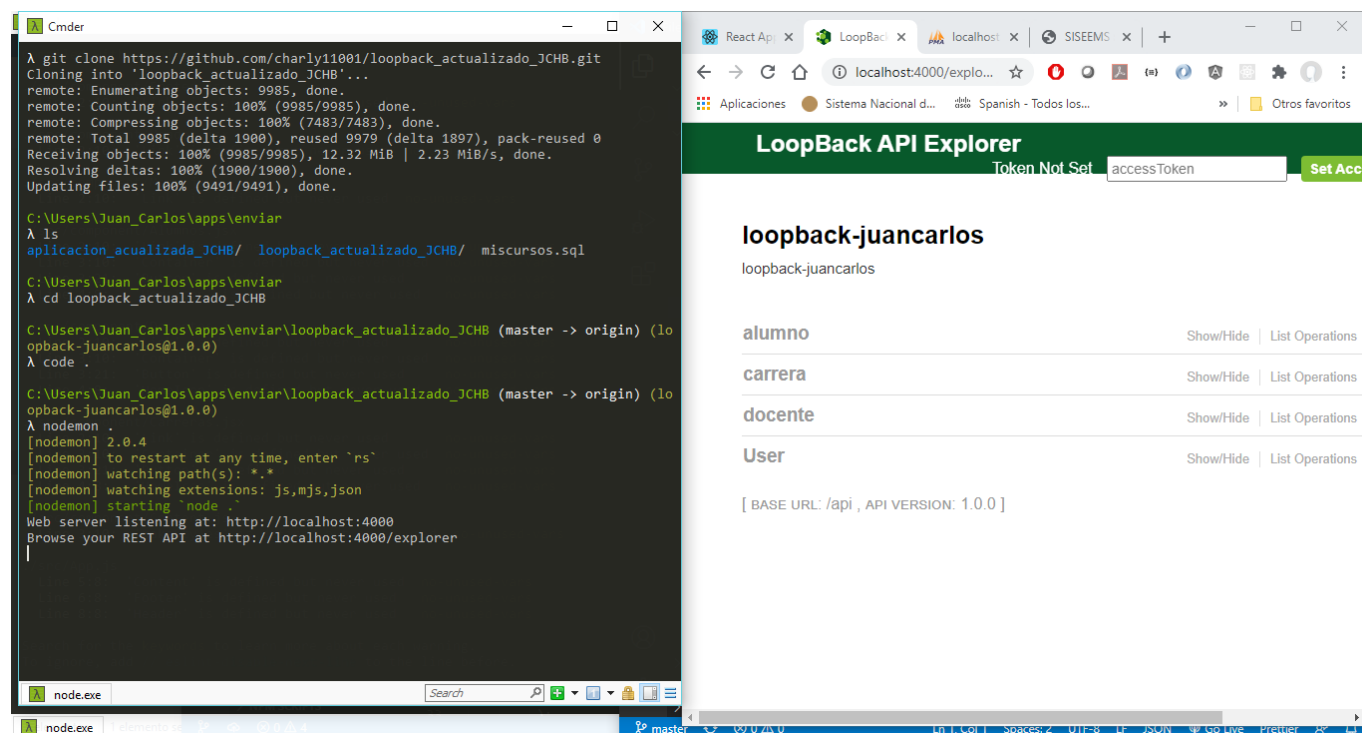
Después hay que escribir el comando `code .` para ingresar a Visual Studio Code y poder ver el código fuente de nuestra aplicación de Loopback, como se muestra a continuación.



```
File Edit Selection View Go Run ... datasources.json - loopback_actualizado_JCHB - V...
EXPLORER
  X {} datasources.json s...
  LOOPBACK_ACTUALIZADO...
    > client
    > common
    > node_modules
    > server
      > boot
      {} component-configs...
      {} config.json
      {} datasources.json
      {} middleware.json
      {} middleware.develop...
      {} model-config.json
      JS server.js
      {} .yo-rc.json
      {} alumnos1.json
      {} package.json
      {} package-lock.json
      README.md
  > OUTLINE
  > TIMELINE
  > NPM SCRIPTS

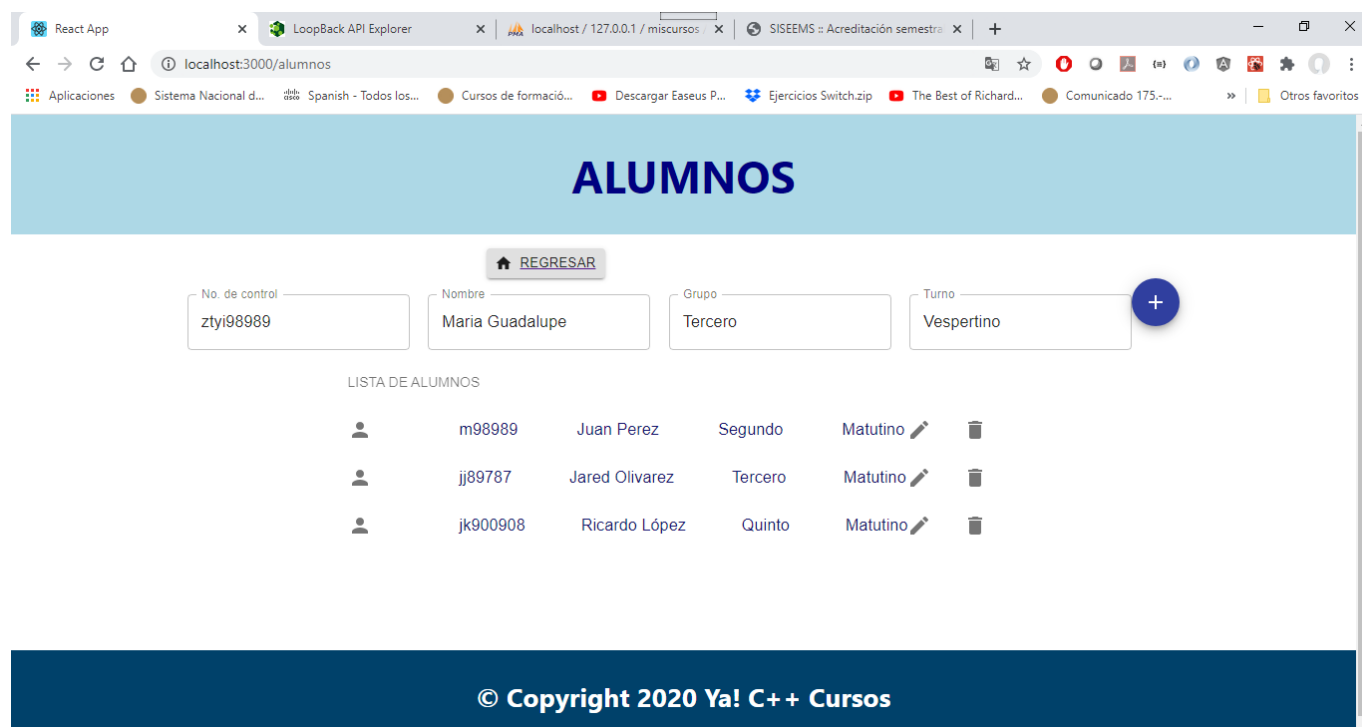
server > {} datasources.json > ...
1
2 "db": {
3   "name": "db",
4   "connector": "memory"
5 },
6 "db_mysql": {
7   "host": "localhost",
8   "port": 3306,
9   "url": "mysql://root:root@localhost/miscursos",
10  "database": "msc_alumnos",
11  "password": "root",
12  "name": "db_mysql",
13  "user": "root",
14  "connector": "mysql"
15 }
16
17
```

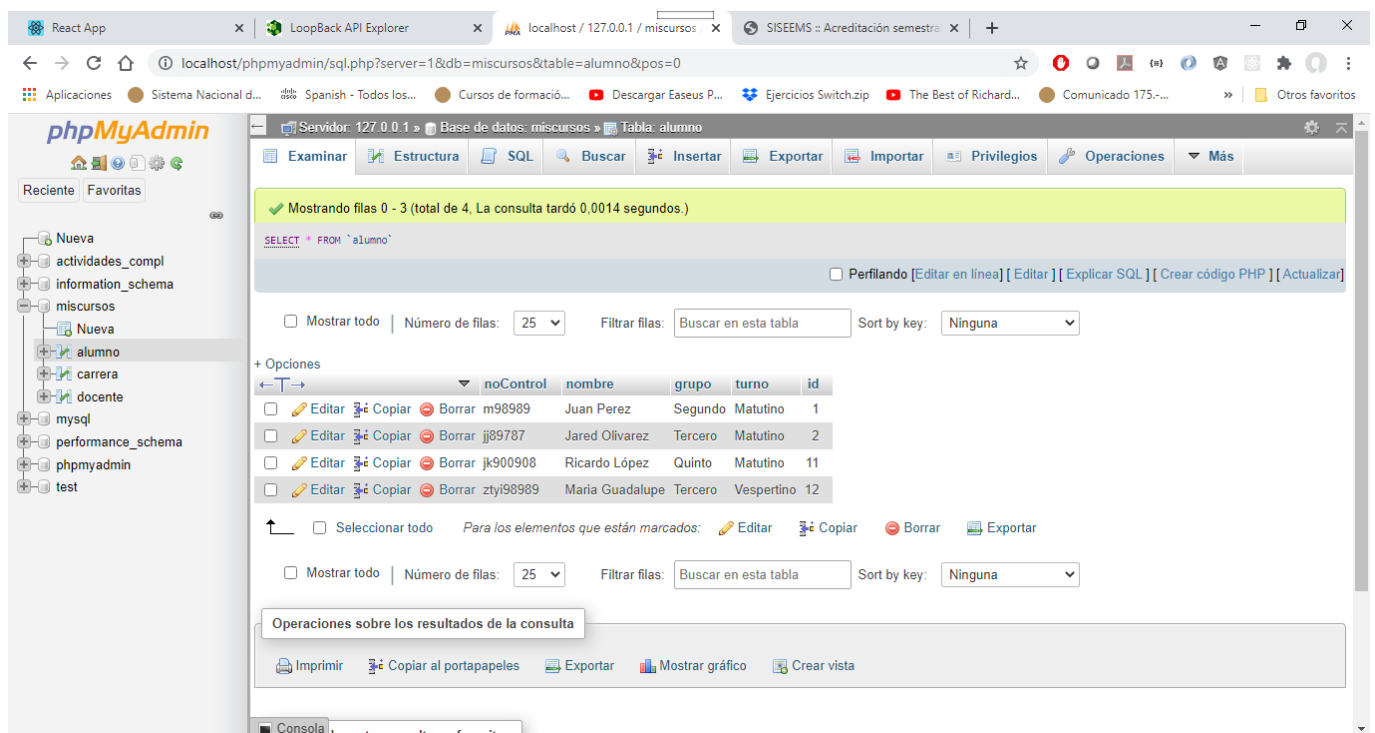
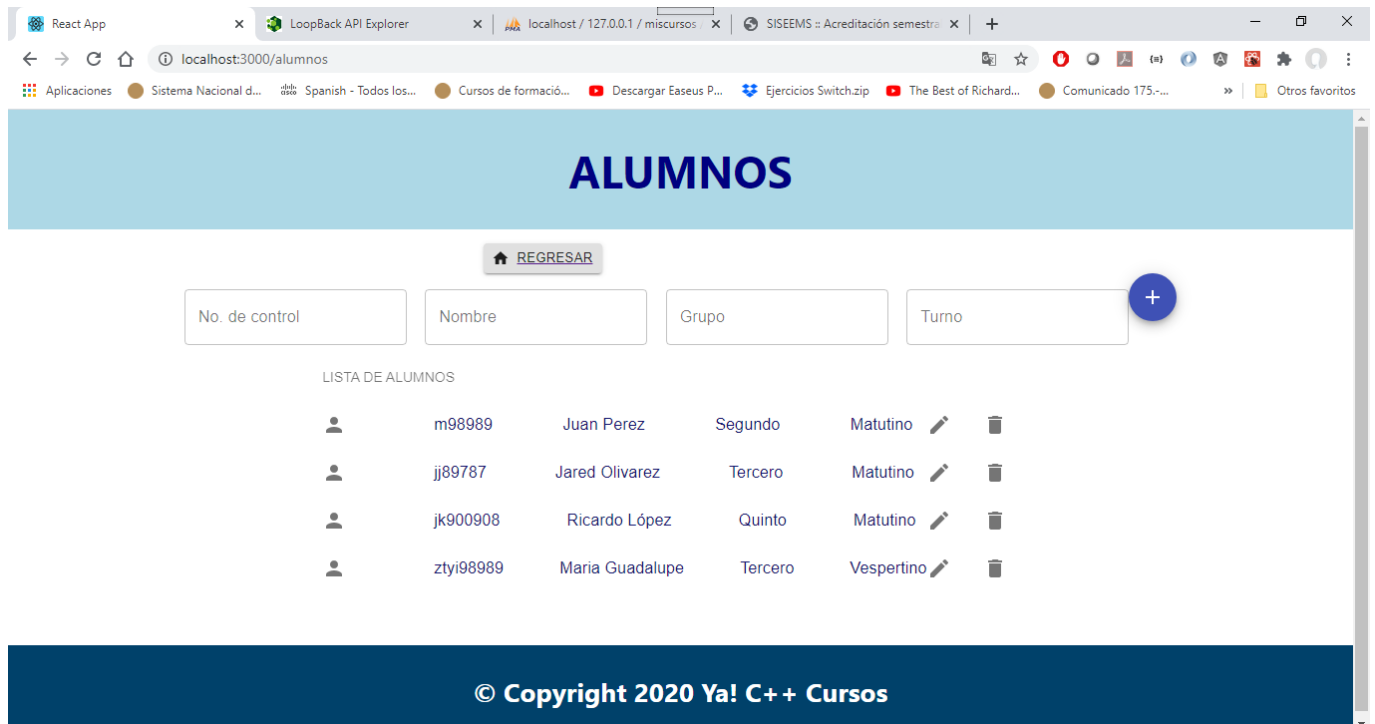
Por último ejecutamos la aplicación Loopback con el comando **nodemon** . e ingresamos a la REST API a través de nuestro navegador para explorarlo, escribiendo la dirección del servidor, en este caso: <http://localhost:4000/explorer>, como se muestra en las siguientes imágenes.



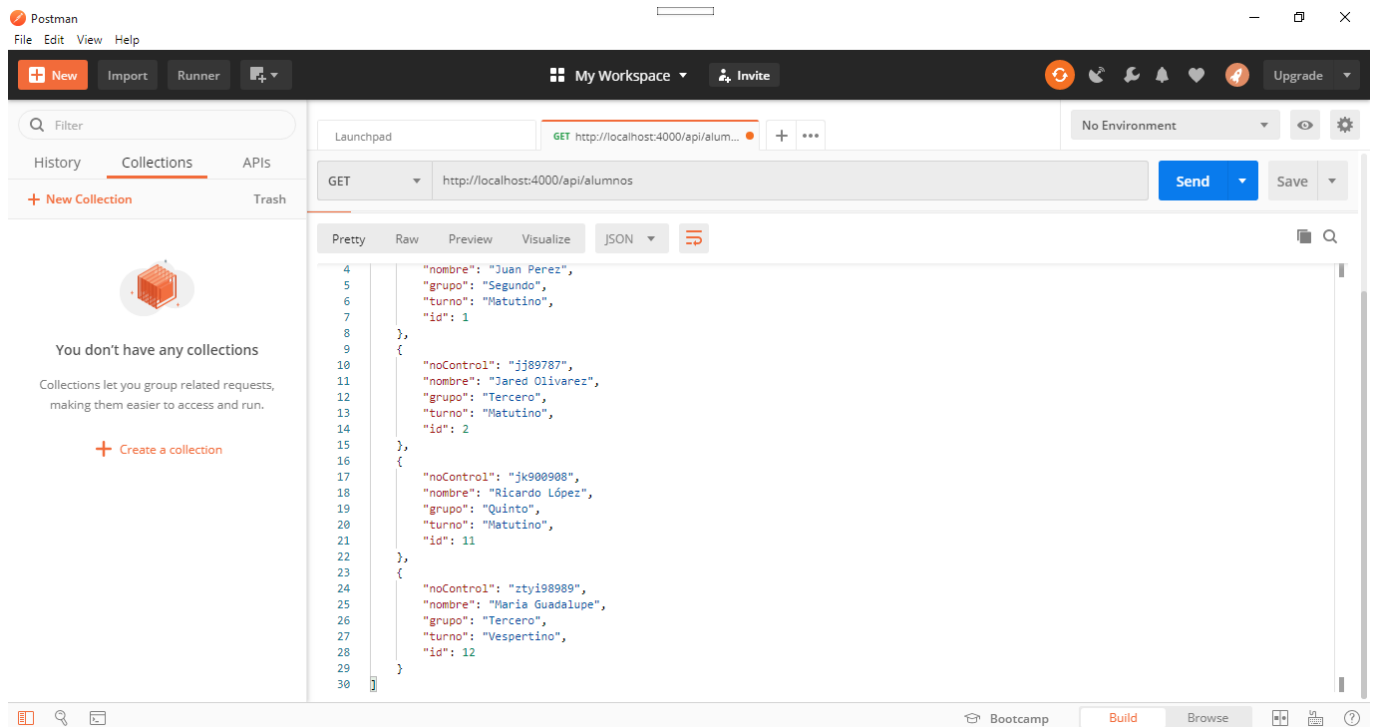
d) Pruebas de la aplicación

Desde la aplicación de React en el navegador se pueden agregar registros a la base de datos, los cuales se verán reflejados dentro de la base de datos de mySQL que creamos previamente, como se muestra en las siguientes imágenes.

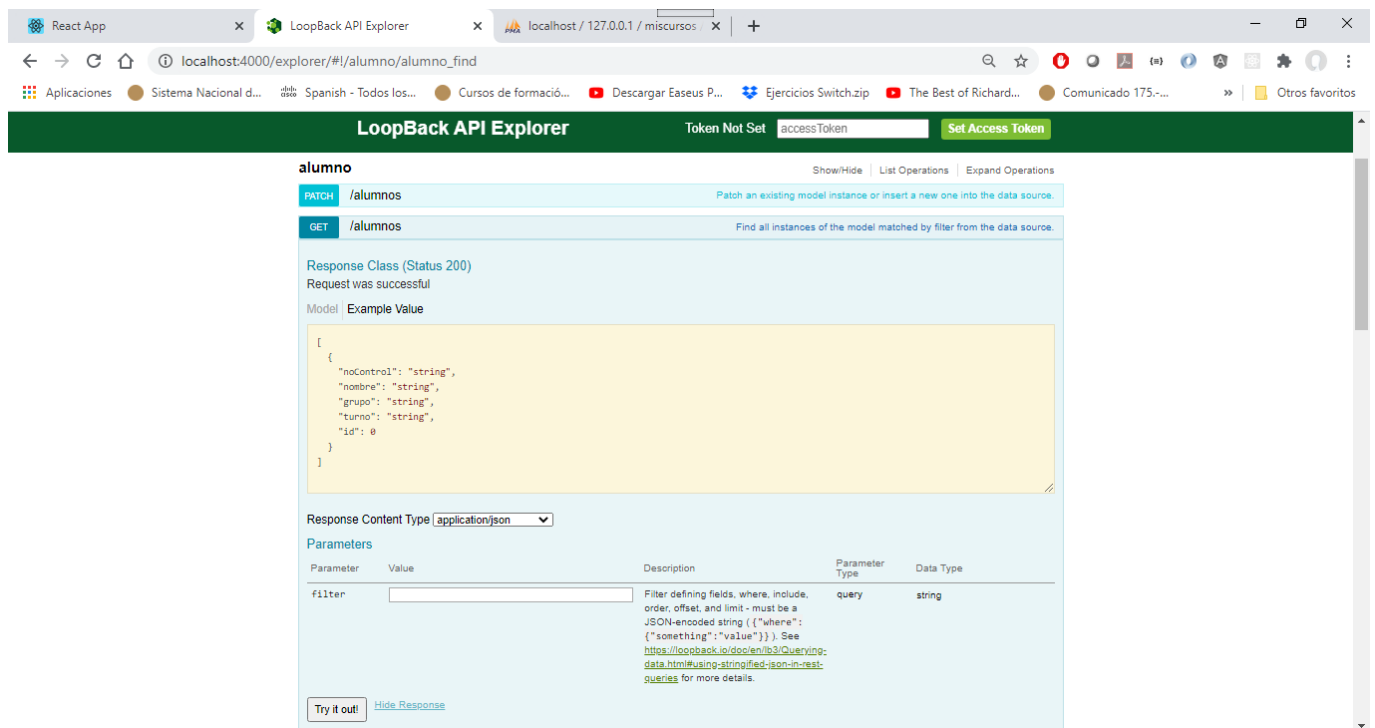


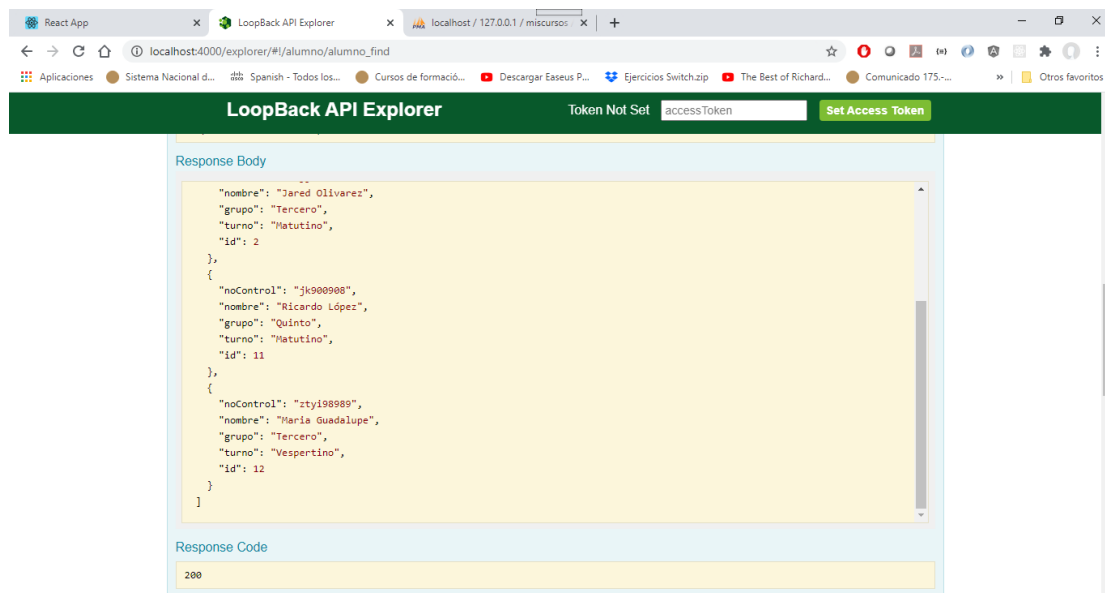


A través de la herramienta Postman realizamos un test de la API para ver los registros del componente Alumnos a través del método Get, donde se puede observar los datos de la alumna que se agregaron en la aplicación de React, como se muestra a continuación.

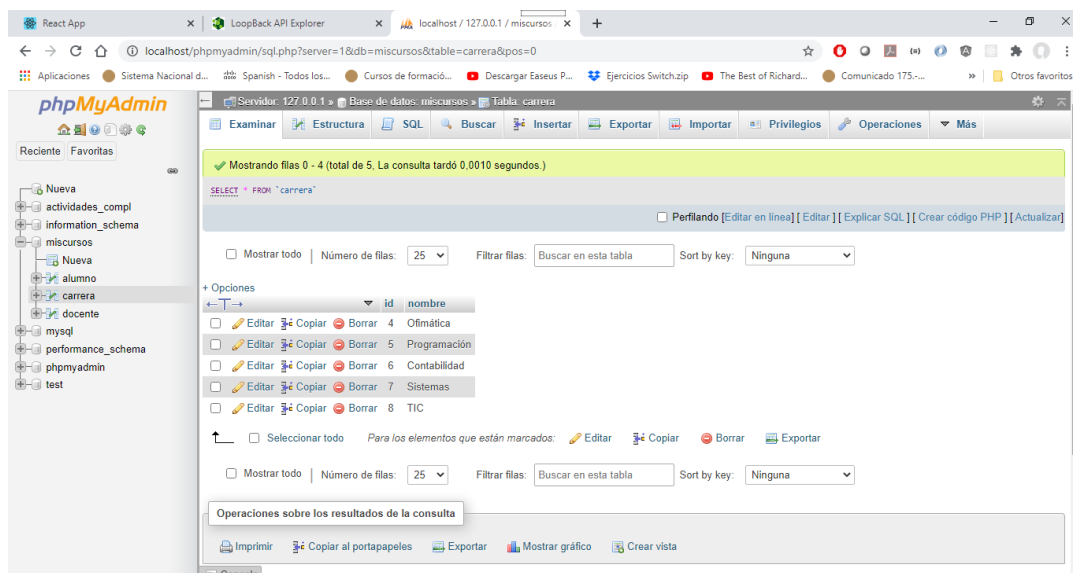
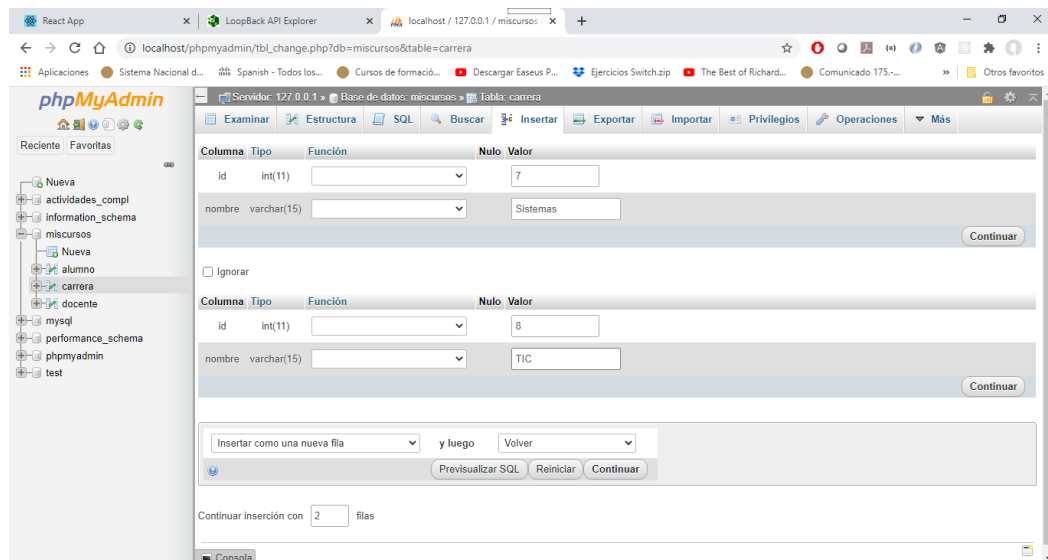


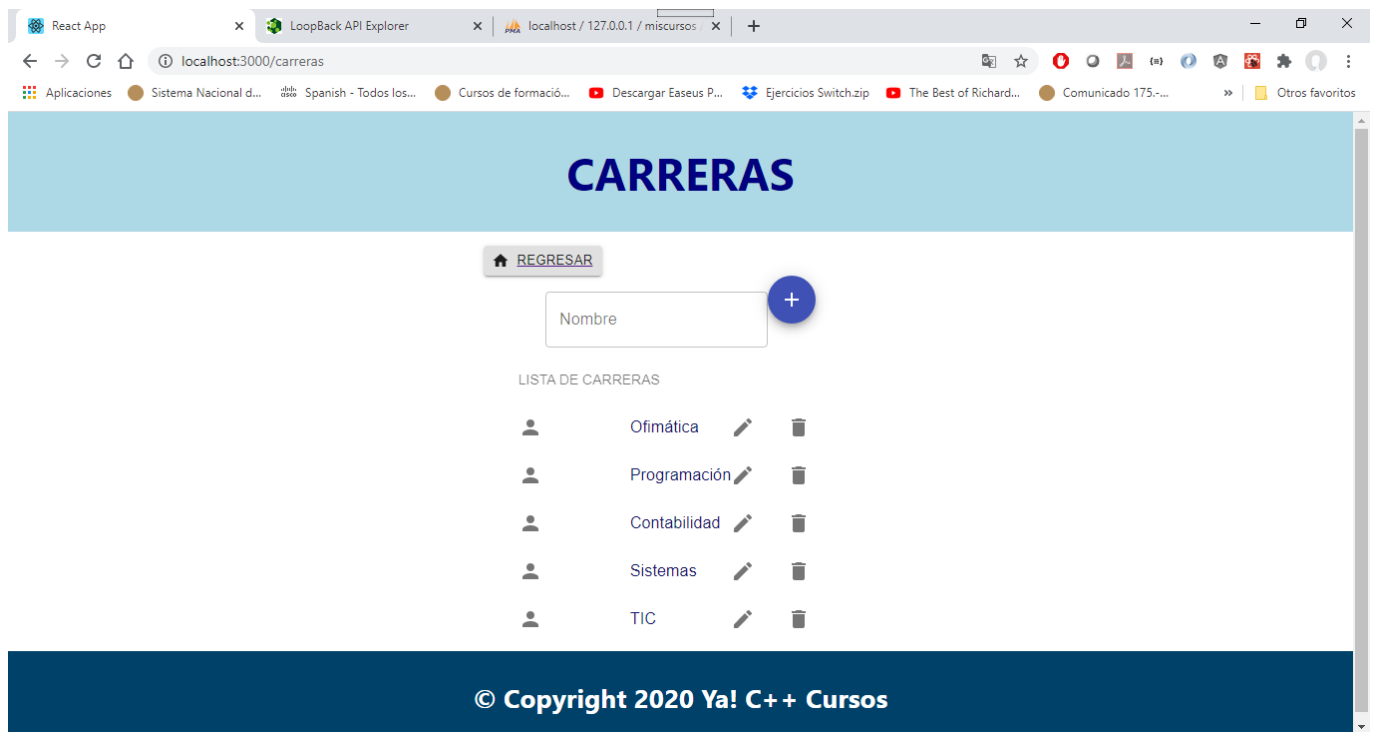
La misma consulta la podemos hacer desde la API Loopback, dando clic en alumno, después en Get /alumnos y dando clic en el botón Try it out!, como se muestra en las siguientes imágenes.



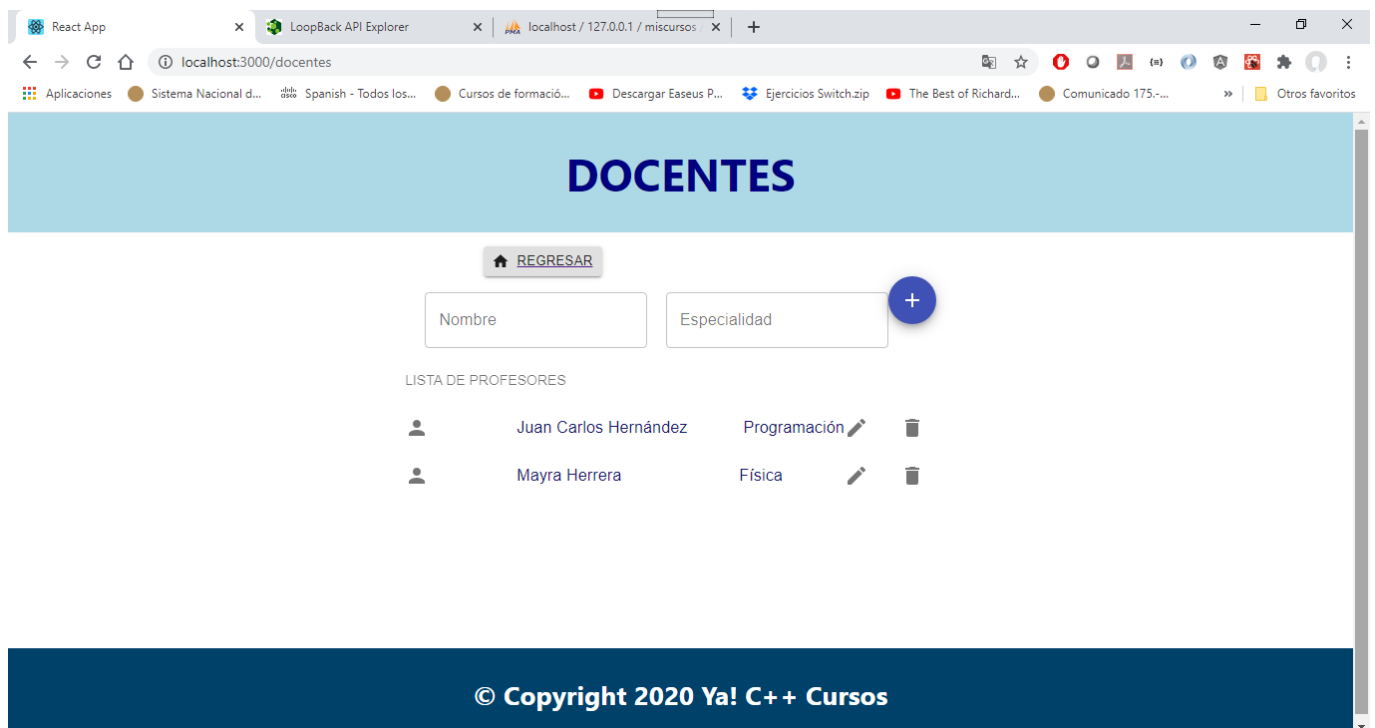


También se pueden agregar registros directamente desde la base de datos de MySQL, los cuales se verán reflejados en la aplicación de React, como se muestra a continuación.





Por último, se puede ingresar a cada elemento del menú principal de la aplicación de React, para comprobar que están funcionando sin ningún problema para insertar, editar y eliminar datos, como se muestra en las siguientes imágenes.



React App x LoopBack API Explorer x localhost / 127.0.0.1 / miscursos x +

localhost:3000/alumnos

Aplicaciones Sistema Nacional d... Spanish - Todos los... Cursos de formació... Descargar Easeus P... Ejercicios Switch.zip The Best of Richard... Comunicado 175-... Otros favoritos

ALUMNOS

[REGRESAR](#)

No. de control Nombre Grupo Turno +

LISTA DE ALUMNOS

	m98989	Juan Perez	Segundo	Matutino		
	jj89787	Jared Olivarez	Tercero	Matutino		
	jk900908	Ricardo López	Quinto	Matutino		
	ztyi98989	Maria Guadalupe	Tercero	Vespertino		

© Copyright 2020 Ya! C++ Cursos

Buscar en Internet y en Windows

04:25 p. m.
04/07/2020

React App x LoopBack API Explorer x localhost / 127.0.0.1 / miscursos x +

localhost:3000/carreras

Aplicaciones Sistema Nacional d... Spanish - Todos los... Cursos de formació... Descargar Easeus P... Ejercicios Switch.zip The Best of Richard... Comunicado 175-... Otros favoritos

CARRERAS

[REGRESAR](#)

Nombre +

LISTA DE CARRERAS

	Ofimática		
	Programación		
	Contabilidad		
	Sistemas		
	TIC		

© Copyright 2020 Ya! C++ Cursos

Buscar en Internet y en Windows

04:26 p. m.
04/07/2020

Listo!!!