

ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA



DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

CARRERA:

ING. DE SOFTWARE

NRC:

8500

ASIGNATURA:

SISTEMAS DE BASES DE DATOS AVANZADOS

TEMA:

PROYECTO

FECHA:

29/08/2022

TECNOLOGÍAS USADAS

SERVIDOR:

Node JS, Express

CLIENTE:

React, NodeJs

Dependencias usadas:

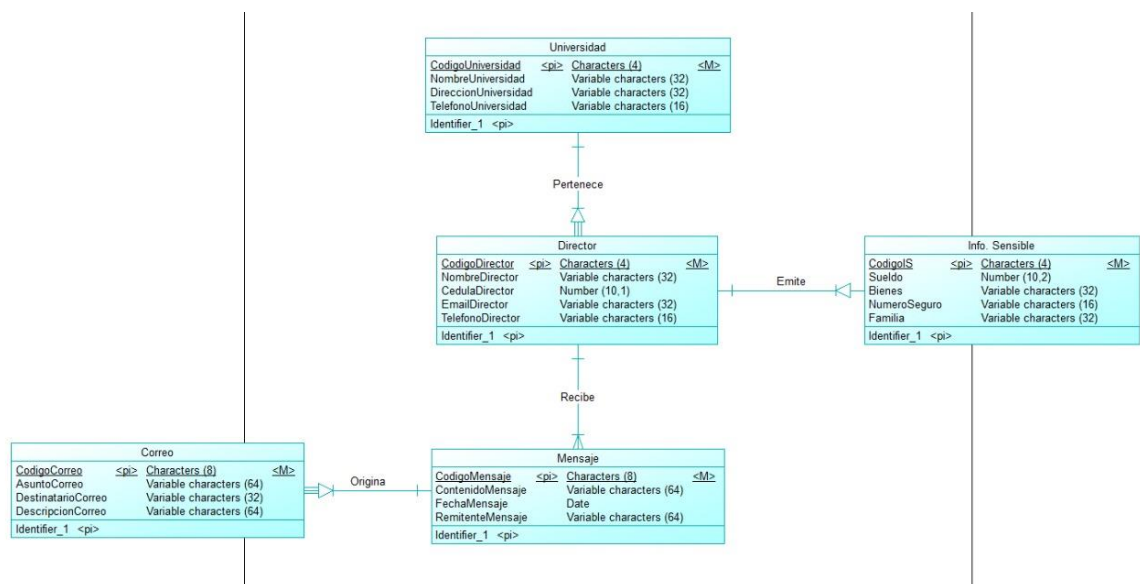
1. "aos": "^2.3.4",
2. "axios": "^0.27.2",
3. "body-parser": "^1.20.0",
4. "react": "^18.2.0",
5. "react-dom": "^18.2.0",
6. "react-router-dom": "^6.3.0",
7. "react-scripts": "5.0.1",
8. "sweetalert2": "^11.4.24",
9. "uniquid": "^1.1.4",
10. "web-vitals": "^2.1.4"

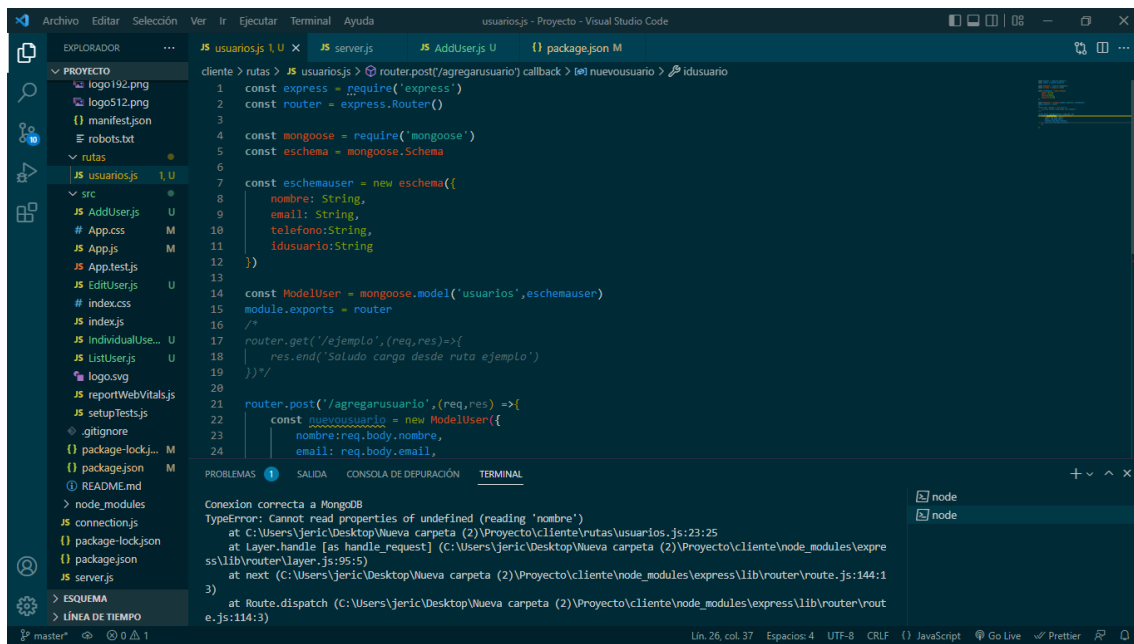
IDE a utilizar : Visual Studio Code

Sistema Operativo: Windows

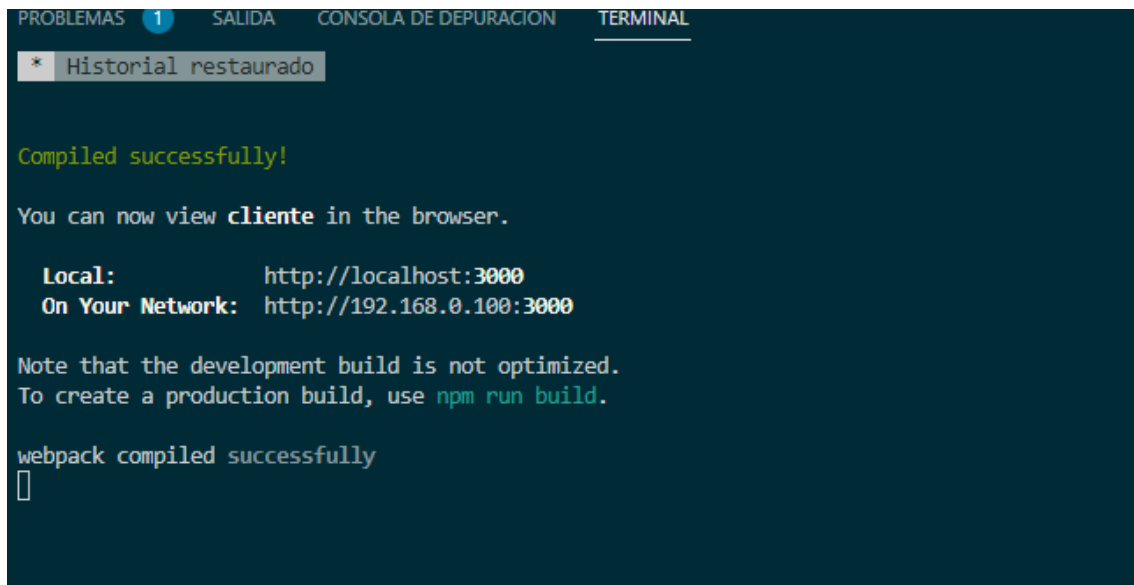
Requisitos: Visual Studio Code, Node Js, React, MongoDB Compass

MODELO CONCEPTUAL DE LA BASE DE DATOS

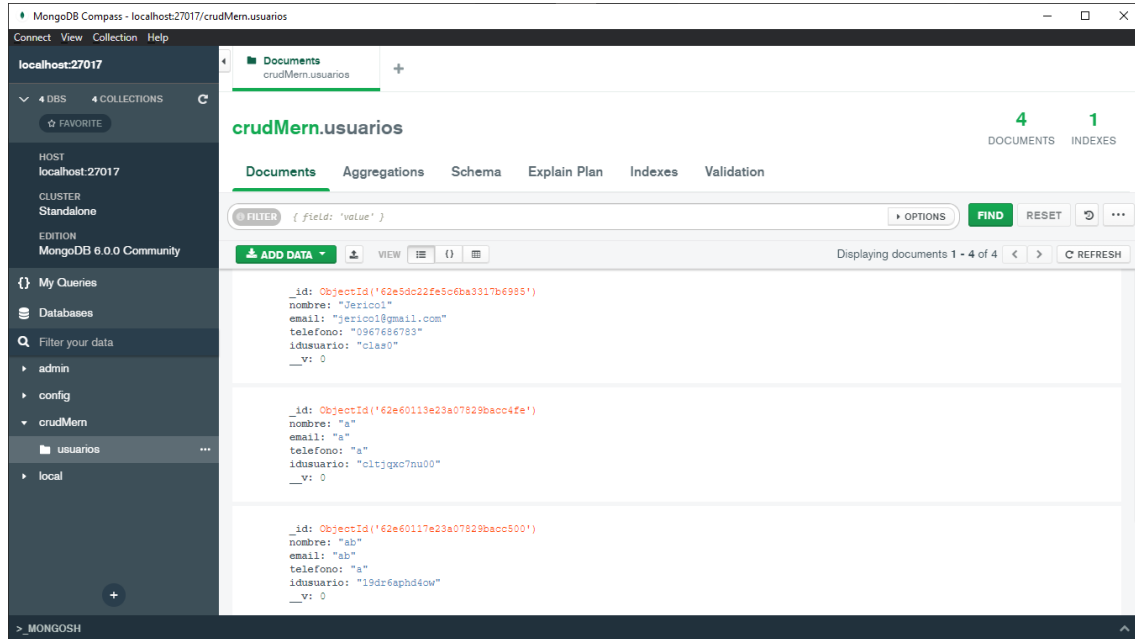




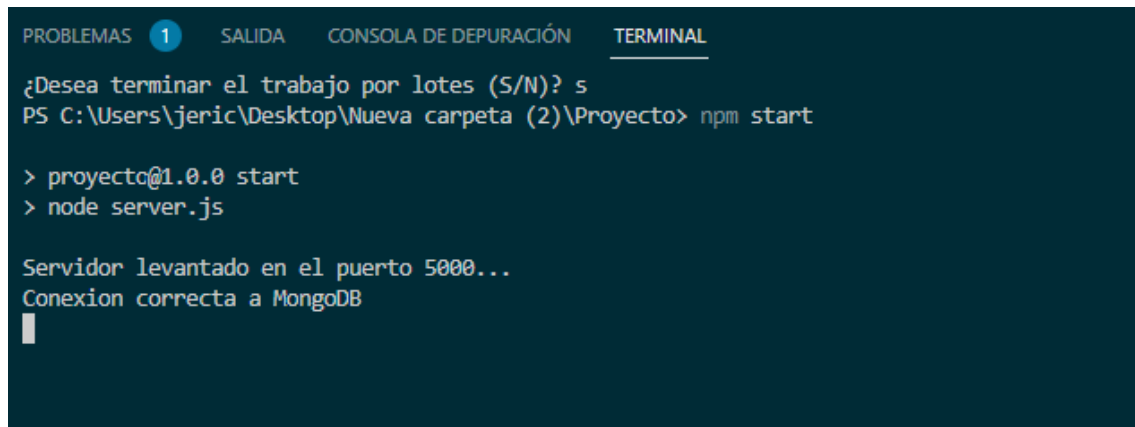
Servidor FrontEnd levantado en el puerto 3000



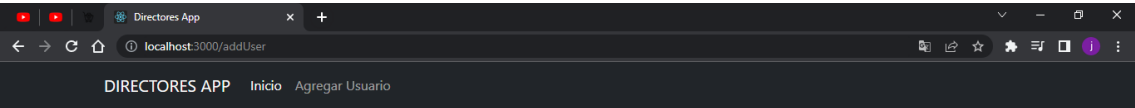
USUARIOS LEIDOS DE LA BASE DE DATOS



Servidor BackEnd levantado en el puerto 5000



Formulario para el envío de correo



Mensajes

NOMBRE

EMAIL

TELEFONO

Enviar mensaje

Home

New Customer

Customer List

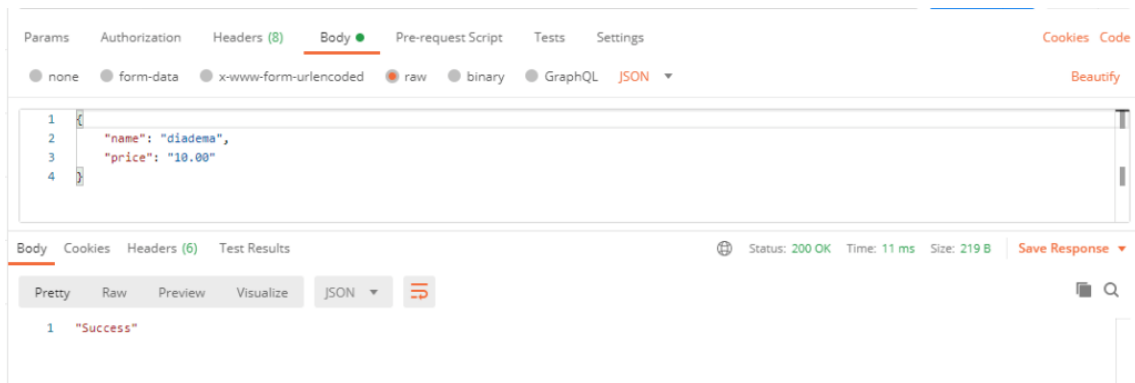
#	Name	Email	Action
1	Asdf	qwerty@mail.com	<div>EditDelete</div>
2	hivnbvnbvn	hiiQ@gmail.com	<div>EditDelete</div>
3	Kyyyyya	ffff@fgh.hhhj	<div>EditDelete</div>
4	Joe Marie Lui	sdfsd@asdad.asd	<div>EditDelete</div>
5	Thanh	P.thanhit@gmail.com	<div>EditDelete</div>
6	Jim Kurosaki	kurosakijim@unwanted.kr	<div>EditDelete</div>
7	admin	admin@admin.com	<div>EditDelete</div>
8	Teste	Gsa@gbh.nb	<div>EditDelete</div>
9	Gsf	dang@gmail.com	<div>EditDelete</div>
10	R	a@gmail.com	<div>EditDelete</div>

Name

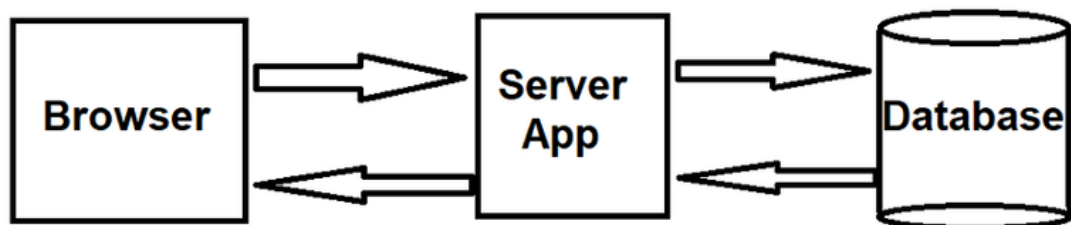
Email address

Enter email

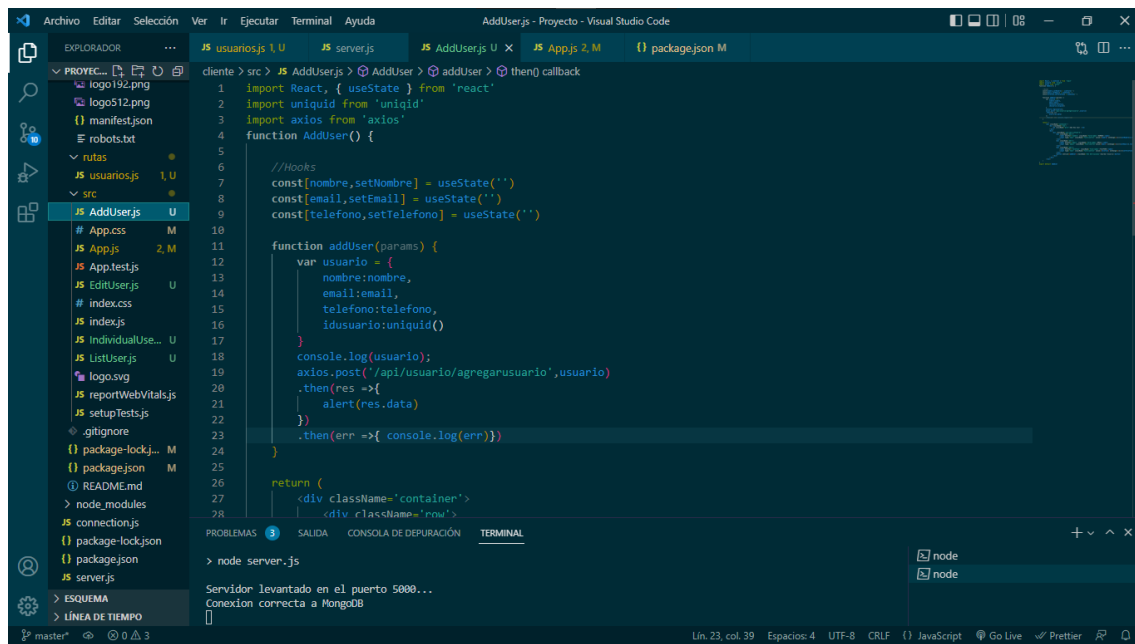
Add



ARQUITECTURA DE CLIENTE, SERVIDOR, BD



Evidencias del código fuente

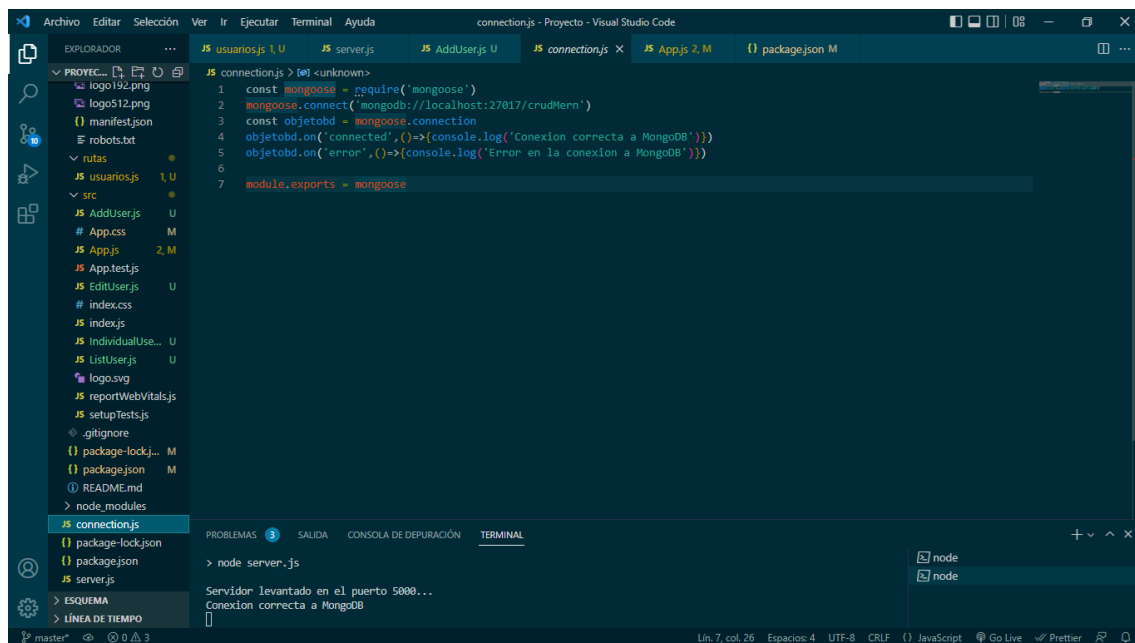


The screenshot shows the Visual Studio Code editor with the file explorer on the left displaying the project structure. The main editor window shows the `AddUser.js` file. The code defines a React component `AddUser` that uses `useState` for form fields and `axios` to send a POST request to the `/api/usuario/agregarusuario` endpoint. The terminal at the bottom shows the command `node server.js` being executed, with output indicating the server is running on port 5000 and connected to MongoDB.

```
1 import React, { useState } from 'react'
2 import uniqid from 'uniqid'
3 import axios from 'axios'
4 function AddUser() {
5
6   //Hooks
7   const[nombre,setNombre] = useState('')
8   const[email,setEmail] = useState('')
9   const[telefono,setTelefono] = useState('')
10
11   function addUser(params) {
12     var usuario = {
13       nombre:nombre,
14       email:email,
15       telefono:telefono,
16       idusuario:uniqid()
17     }
18     console.log(usuario);
19     axios.post('/api/usuario/agregarusuario',usuario)
20       .then(res =>{
21         alert(res.data)
22       })
23       .then(err =>{ console.log(err)})
24   }
25
26   return (
27     <div className='container'>
28       <div className='row'>
```

PROBLEMAS 3 SALIDA CONSOLA DE DEPURACIÓN TERMINAL

```
> node server.js
Servidor levantado en el puerto 5000...
Conexion correcta a MongoDB
```

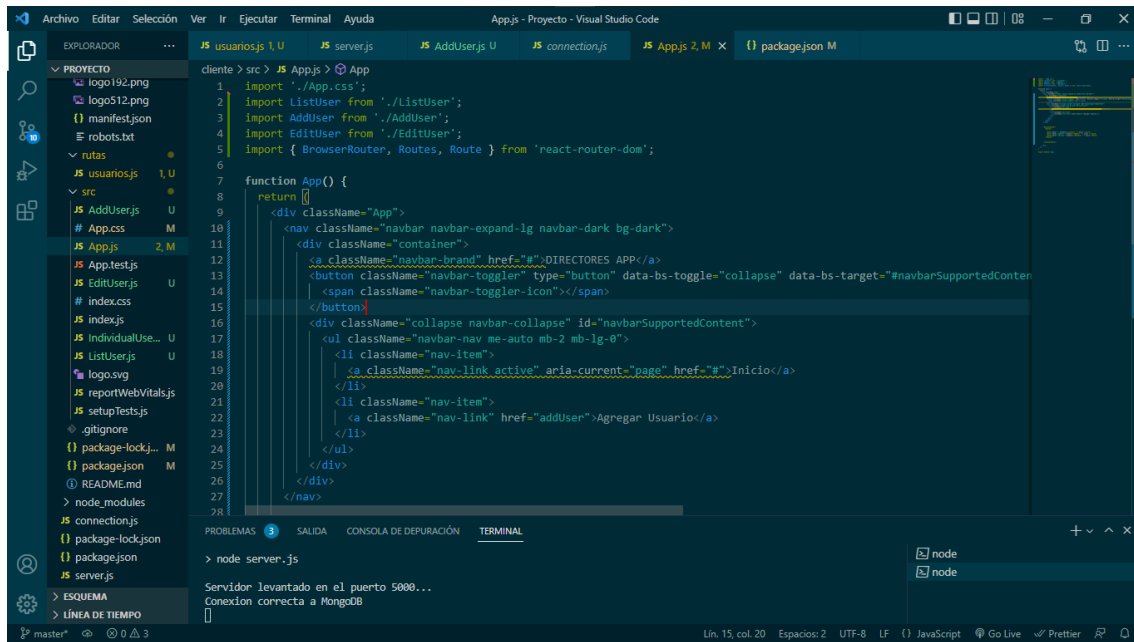


The screenshot shows the Visual Studio Code editor with the file explorer on the left displaying the project structure. The main editor window shows the `connection.js` file. The code uses `mongoose` to connect to a MongoDB database at `localhost:27017/crudMern`. The terminal at the bottom shows the command `node server.js` being executed, with output indicating the server is running on port 5000 and connected to MongoDB.

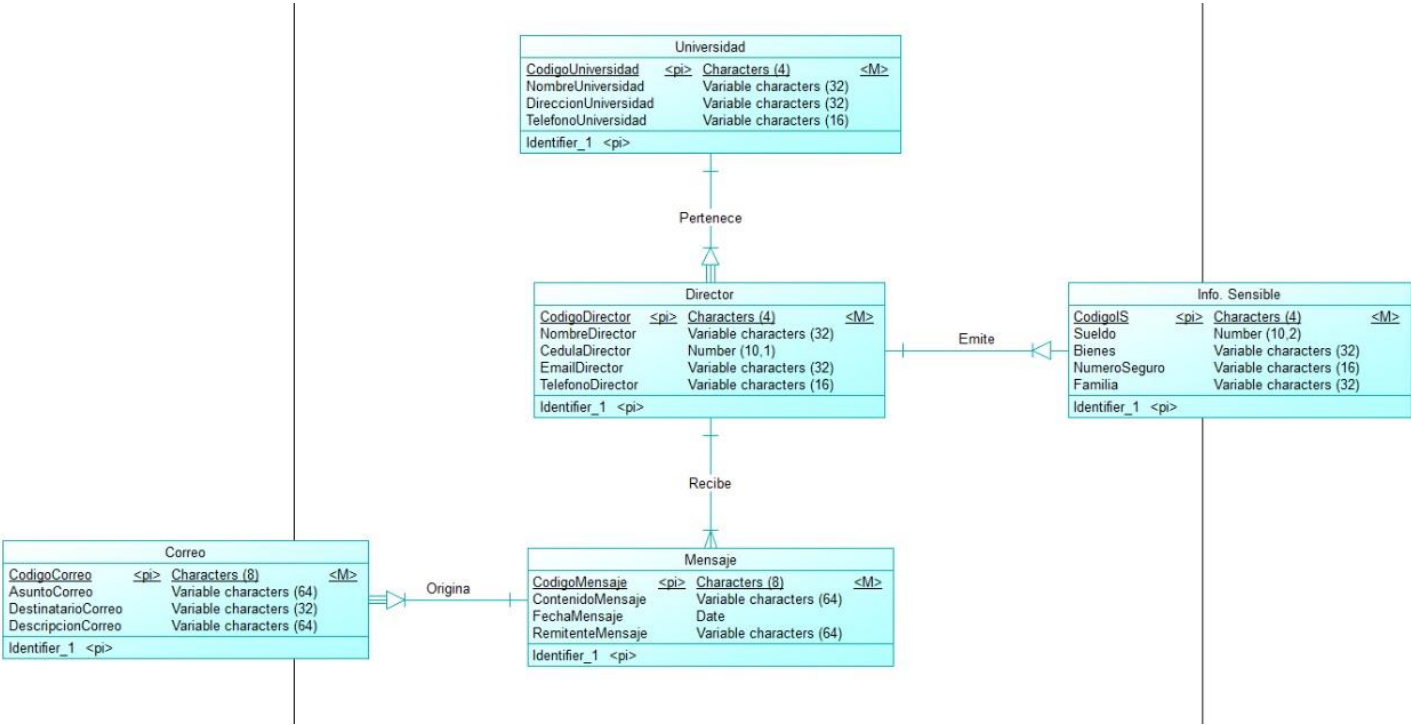
```
1 const mongoose = require('mongoose')
2 mongoose.connect('mongodb://localhost:27017/crudMern')
3 const objetodb = mongoose.connection
4 objetodb.on('connected',()=>{console.log('Conexion correcta a MongoDB')})
5 objetodb.on('error',()=>{console.log('Error en la conexion a MongoDB')})
6
7 module.exports = mongoose
```

PROBLEMAS 3 SALIDA CONSOLA DE DEPURACIÓN TERMINAL

```
> node server.js
Servidor levantado en el puerto 5000...
Conexion correcta a MongoDB
```



Modelo Conceptual



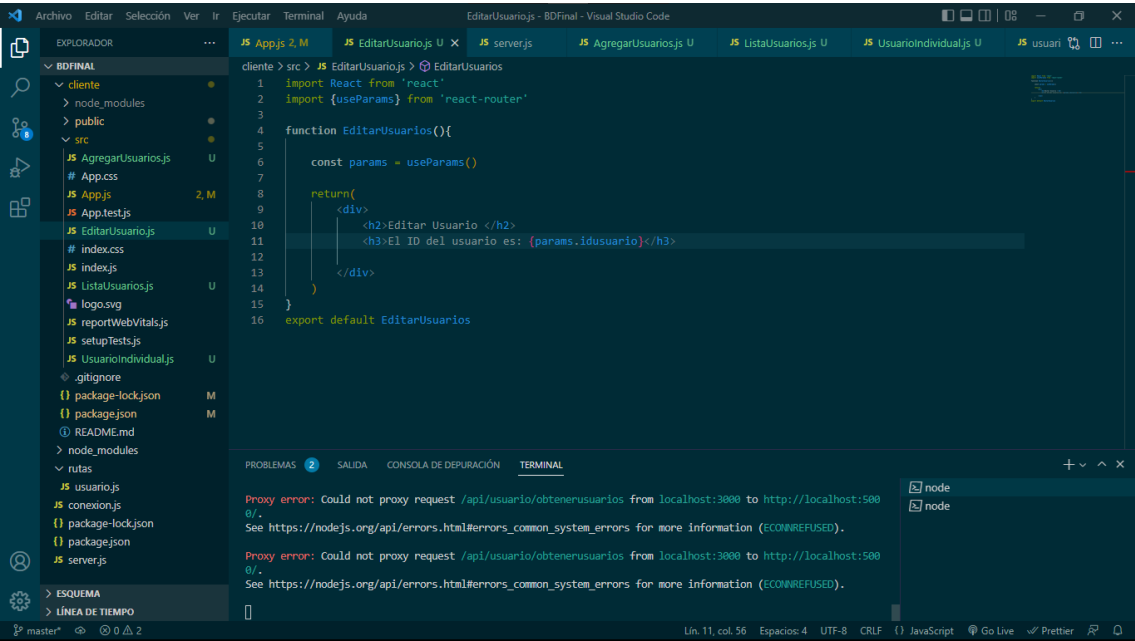
DIRECTORES APPInicioAgregar Usuario

Lista de Directores

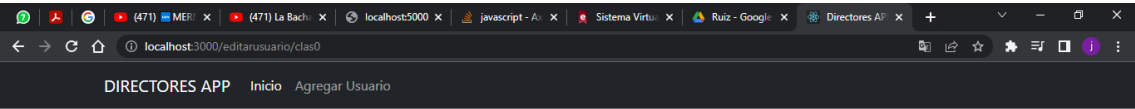
cltge8u8qaw0
Jerico
jerico@gmail.com
0967686783

EditarBorrar

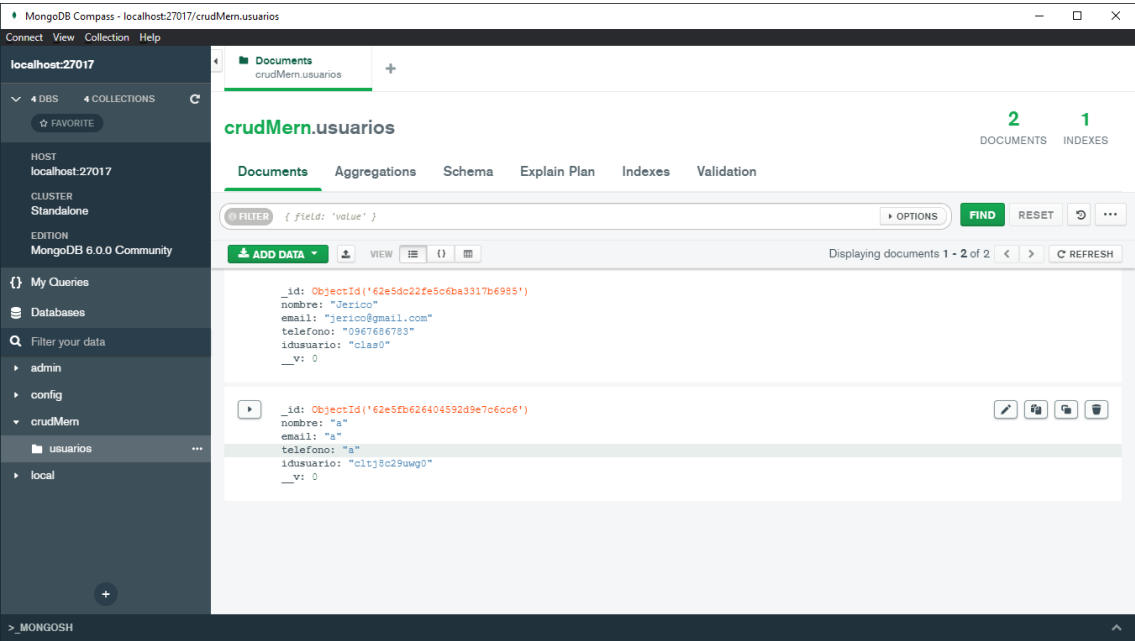
SEGURIDAD PROXY solo en el puerto 5000



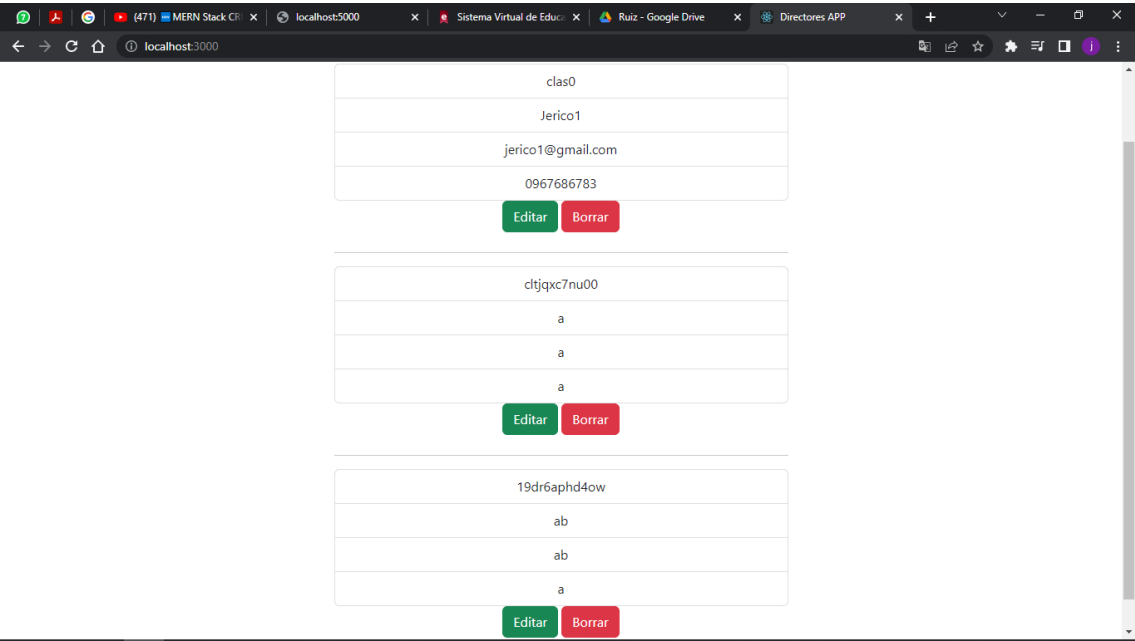
Funcion editar usuario con información sensible



BASE DE DATOS



Prueba de Registro de los usuarios y la información sensible



CÓDIGO DEL PROYECTO

```
import React, {useState} from 'react'
import uniquid from 'uniquid'
import axios from 'axios'
import Swal from 'sweetalert2'

function AgregarUsuarios() {

  //Hooks
  const[nombre,setNombre] = useState('')
  const[email,setEmail] = useState('')
  const[telefono,setTelefono] = useState('')

  function agregarUsuario() {
    var usuario = {
      nombre:nombre,
      email:email,
      telefono:telefono,
      idusuario:uniquid()
    }
    console.log(usuario)

    axios.post('/api/usuario/agregarusuario', usuario)
    .then(res =>{
      //alert(res.data)
      Swal.fire('Felicidades','El usuario se creo con éxito')
    })
    .then(err =>{ console.log(err)})
  }

  return (
    <div className='container'>
      <div className='row'>
        <h2 className='mt-4'>Crear nuevo Usuario </h2>
      </div>
      <div>
        <div className='col-sm-6 offset-3'>
          <div className='mb-3'>
            <label htmlFor='nombre' className='form-label'>NOMBRE</label>
            <input type='text' className='form-control' value={nombre} onChange={(e)=>{setNombre(e.target.value)}}></input>
          </div>
          <div className='mb-3'>
            <label htmlFor='email' className='form-label'>EMAIL</label>
```

```

        <input type='email' className='form-
control' value={email} onChange={(e)=>{setEmail(e.target.value)}}></input>
      </div>
      <div className='mb-3'>
        <label htmlFor='telefono' className='form-
label'>TELEFONO</label>
        <input type='text' className='form-control'
value={telefono} onChange={(e)=>{setTelefono(e.target.value)}}></input>
      </div>
      <button onClick={agregarUsuario} className='btn btn-
success'>Guardar Usuario</button>
    </div>
  </div>
</div>
)
}export default AgregarUsuarios

```

LISTA DE USUARIOS DIRECTORES

```

import axios from 'axios'
import React, {useState,useEffect}from 'react'
import UsuarioIndividual from './UsuarioIndividual'

function ListaUsuarios(){
  const[datausuarios,setdatausuario] = useState([])

  useEffect(()=>{
    axios.get('api/usuario/obtenerusuarios').then(res =>{
      console.log(res.data)
      setdatausuario(res.data)
    }).catch(err=>{
      console.log(err);
    })
  },[])

  //Mapear listadeusuario en objetousuario
  const listausuarios = datausuarios.map(usuario =>{
    return(
      <div>
        <UsuarioIndividual usuario = {usuario}/>
      </div>
    )
  })

  return(
    <div>

```

```

        <h2>Lista de Directores </h2>
        {listausuarios}
      </div>
    )
  }
export default ListaUsuarios

```

USUARIO DIRECTOR INDIVIDUAL

```

import React,{useEffect} from 'react'
import { Link,useNavigate } from 'react-router-dom'
import axios from 'axios'
import AOS from 'aos'
import 'aos/dist/aos.css'

function UsuarioIndividual({ usuario }) {

  const navegar = useNavigate()

  //Para animacion de scroll al bajar
  useEffect(() => {
    AOS.init()
  }, [])

  //Funcion para borrar usuario
  function borrarusuario(idusuario) {
    axios.post('/api/usuario/borrarusuario', { idusuario: idusuario
  }).then(res => {
    console.log(res.data)
    alert(res.data)
    navegar(0)
  }).catch(err => {
    console.log(err)
  })
}

  return (
    <div className='container'>
      <div className='row'>

        <div className='col-sm-6 offset-3' data-aos='flip-right'>

          <ul className='list-group'>
            <li className='list-group-
item'>{usuario.idusuario}</li>
            <li className='list-group-
item'>{usuario.nombre}</li>
            <li className='list-group-
item'>{usuario.email}</li>

```

```

        <li className='list-group-
item'>{usuario.telefono}</li>

    </ul>
    <Link to={`/editarusuario/${usuario.idusuario}`}><li
className='btn btn-success'> Editar</li></Link>
    &nbsp;
    <button className='btn btn-danger' onClick={() => {
borrarusuario(usuario.idusuario) }}>Borrar</button>
    <hr className='mt-4'></hr>

</div>

</div>

</div>
)
}
export default UsuarioIndividual

```

RUTA DE USUARIO

```

const express = require('express')
const router = express.Router()

const mongoose = require('mongoose')
const esquema = mongoose.Schema

const esquemausuario = new esquema({
  nombre: String,
  email: String,
  telefono:String,
  idusuario:String
})

const ModeloUsuario = mongoose.model('usuarios',esquemausuario)
module.exports = router

router.post('/agregarusuario',(req,res)=>{
  const nuevousuario = new ModeloUsuario({
    nombre:req.body.nombre,
    email:req.body.email,
    telefono:req.body.telefono,
    idusuario:req.body.idusuario
  })
})

```

```

    })
    nuevousuario.save(function(err){
      if(!err){
        res.send('Usuario agregado correctamente')
      }else{
        res.send(err)
      }
    })
  })
})

router.get('/obtenerusuarios',(req,res)=>{
  ModeloUsuario.find({},function(docs,err){
    if(!err){
      res.send(docs)
    }else{
      res.send(err)
    }
  })
})

router.post('/obtenerdatausuario',(req,res)=>{
  ModeloUsuario.find({idusuario:req.body.idusuario},function(docs,err){
    if(!err){
      res.send(docs)
    }else{
      res.send(err)
    }
  })
})

router.post('/actualizarusuario',(req,res)=>{
  ModeloUsuario.findOneAndUpdate({idusuario:req.body.idusuario},{
    nombre:req.body.nombre,
    email:req.body.email,
    telefono: req.body.telefono
  },(err)=>{
    if(!err){
      res.send('Usuario Actualizado Correctamente')
    }else{
      res.send(err)
    }
  })
})

router.post('/borrarusuario',(req,res)=>{
  ModeloUsuario.findOneAndDelete({idusuario:req.body.idusuario},(err)=>
  {
    if(!err){
      res.send('Usuario Eliminado Correctamente')
    }
  })
})

```



```

        }else{
            res.send(err)
        }
    })
})

```

CONEXIÓN SERVIDOR BACKEND

```

const express = require('express')
const app = express()

//Importar conexion mongoDB
const archivoBD = require('./conexion')

//Importar rutas y modelo
const rutausuario =require('./rutas/usuario')

//Importar body-parser
const bodyParser = require('body-parser')
app.use(bodyParser.json())
app.use(bodyParser.urlencoded({extended:'true'}))

app.use('/api/usuario',rutausuario)

app.get('/',(req,res)=>{
    res.end('Bienvenido al servidor backend NODE. Corriendo...')
})

//Configuracion server basico
app.listen(5000, function(){
    console.log('Servidor levantado en el puerto 5000...');
})

```

CONEXIÓN BASE DE DATOS

```

const mongoose = require('mongoose')
mongoose.connect('mongodb://localhost:27017/crudMern')
const objetobd = mongoose.connection
objetobd.on('connected',()=>{console.log('Conexion correcta a MongoDB')})
objetobd.on('error',()=>{console.log('Error en la conexion a MongoDB')})

module.exports = mongoose

```

“La innovación distingue entre un líder y un seguidor.”

- STEVE JOBS