

Visual Studio Code interface showing a project named 'my-app' with a file explorer on the left and a code editor on the right. The code editor displays the 'Formulario.js' file, which contains JavaScript code for a form submission function.

File Explorer (Left):

- my-app
 - node_modules
 - public
 - favicon.ico
 - index.html
 - logo192.png
 - logo512.png
 - manifest.json
 - robots.txt
 - src
 - components
 - Cita.js
 - Formulario.js
 - App.js
 - index.css
 - index.js
 - serviceWorker.js
 - .gitignore
 - package-lock.json
 - package.json
 - README.md
 - package-lock.json
 - package.json

Code Editor (Right):

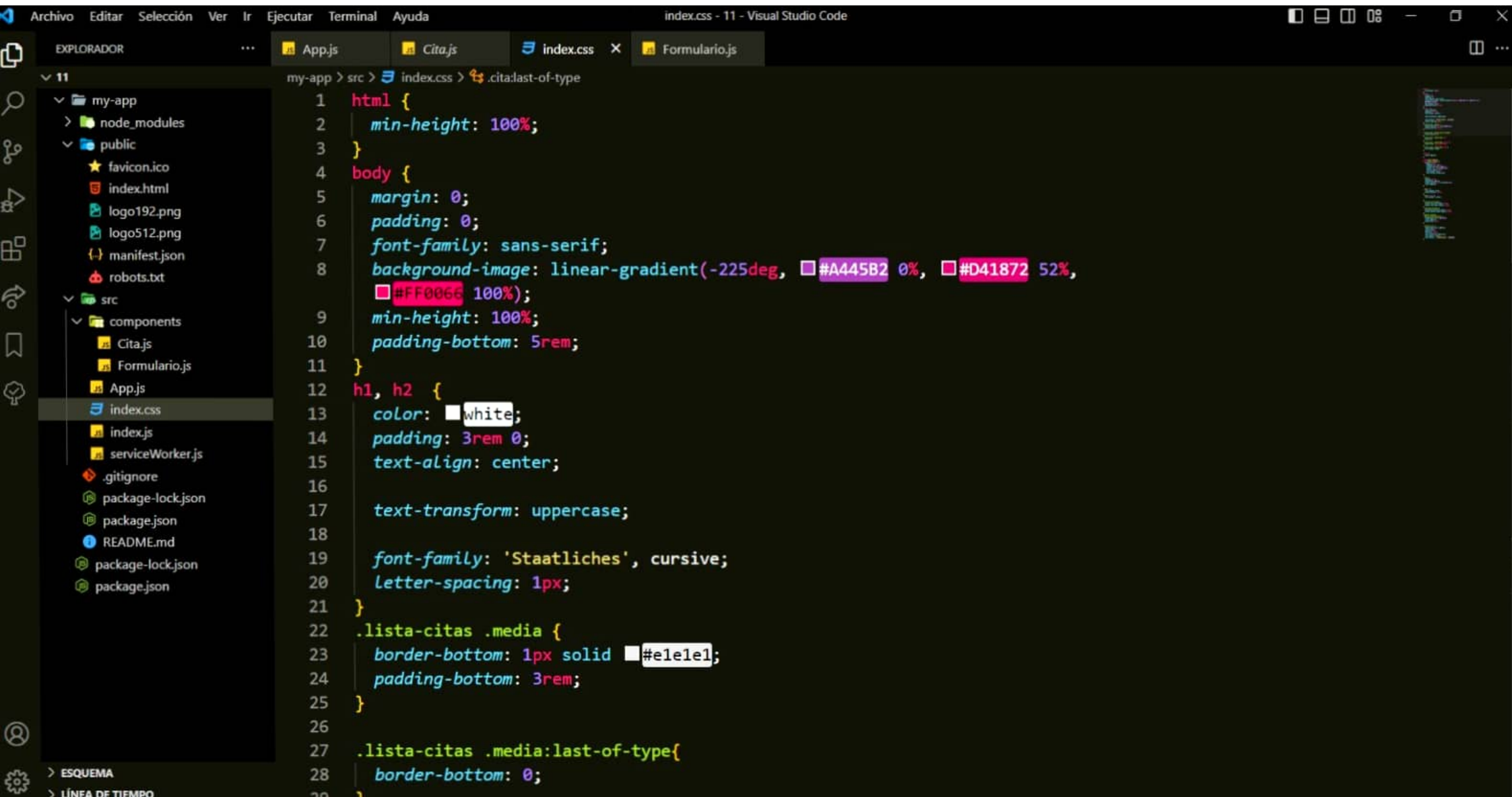
```
1 import React, { Fragment, useState } from 'react'
2
3 import PropTypes from 'prop-types'
4
5 const Formulario = ({ crearCita }) => {
6   // Crear state de Citas
7   const [cita, actualizarCita] = useState({
8     mascota: '',
9     propietario: '',
10    fecha: '',
11    hora: '',
12    sintomas: '',
13  })
14  const [error, actualizarError] = useState(false)
15
16  // Funcion que se ejecuta cada vez que el usuario escribe en un input
17  const actualizarState = (e) => {
18    actualizarCita({
19      ...cita,
20      [e.target.name]: e.target.value,
21    })
22  }
23
24  // Extraer los valores
25  const { mascota, propietario, fecha, hora, sintomas } = cita
26
27  // Cuando el usuario presiona agregar cita
28  const submitCita = (e) => {
29    e.preventDefault()
30  }
```

Status Bar (Bottom):

Lín. 47, col. 1 Espacios: 4 UTF-8 CRLF {} JavaScript Go Live Go Live ✓ Prettier

Taskbar (Bottom):

Windows taskbar with search bar and various application icons.



```
1  html {
2    min-height: 100%;
3  }
4  body {
5    margin: 0;
6    padding: 0;
7    font-family: sans-serif;
8    background-image: linear-gradient(-225deg, #A445B2 0%, #D41872 52%,
9    #FF0066 100%);
10   min-height: 100%;
11   padding-bottom: 5rem;
12 }
13 h1, h2 {
14   color: white;
15   padding: 3rem 0;
16   text-align: center;
17   text-transform: uppercase;
18   font-family: 'Staatliches', cursive;
19   letter-spacing: 1px;
20 }
21
22 .lista-citas .media {
23   border-bottom: 1px solid #e1e1e1;
24   padding-bottom: 3rem;
25 }
26
27 .lista-citas .media:last-of-type{
28   border-bottom: 0;
29 }
```

Archivo Editar Selección Ver Ir Ejecutar Terminal Ayuda Cita.js - 11 - Visual Studio Code

EXPLORADOR

- my-app
 - node_modules
 - public
 - favicon.ico
 - index.html
 - logo192.png
 - logo512.png
 - manifest.json
 - robots.txt
 - src
 - components
 - Cita.js
 - Formulario.js
 - App.js
 - index.css
 - index.js
 - serviceWorker.js
 - .gitignore
 - package-lock.json
 - package.json
 - README.md
 - package-lock.json
 - package.json

my-app > src > components > Cita.js > Cita

```
1 import React from 'react'
2 import PropTypes from 'prop-types'
3
4 const Cita = ({ cita, eliminarCita }) => (
5   <div className='cita'>
6     <p>
7       {' '}
8       Mascota: <span>{cita.mascota}</span>
9     </p>
10    <p>
11      {' '}
12      Dueño: <span>{cita.propietario}</span>
13    </p>
14    <p>
15      {' '}
16      Fecha: <span>{cita.fecha}</span>
17    </p>
18    <p>
19      {' '}
20      Hora: <span>{cita.hora}</span>
21    </p>
22    <p>
23      {' '}
24      Sintomas: <span>{cita.sintomas}</span>
25    </p>
26    <button
27      className='button eliminar u-full-width'
28      onClick={() => eliminarCita(cita.id)}
29    >
30      Eliminar &times;
```


Archivo Editar Selección Ver Ir Ejecutar Terminal Ayuda App.js - 11 - Visual Studio Code

EXPLORADOR

- my-app
 - node_modules
 - public
 - favicon.ico
 - index.html
 - logo192.png
 - logo512.png
 - manifest.json
 - robots.txt
 - src
 - components
 - Cita.js
 - Formulario.js
 - App.js
- index.css
- Formulario.js

```
1 import React, { Fragment, useState, useEffect } from 'react'
2 import Formulario from '../components/Formulario'
3 import Cita from '../components/Cita'
4
5 function App() {
6   // Citas en local storage
7   let citasIniciales = JSON.parse(localStorage.getItem('citas'))
8   if (!citasIniciales) {
9     citasIniciales = []
10  }
11
12  // Arreglos de citas
13  const [citas, guardarCitas] = useState(citasIniciales)
14
15  // use Effect para realizar ciertas operaciones cuando el state cambia
16  useEffect(() => {
17    if (citasIniciales) {
18      localStorage.setItem('citas', JSON.stringify(citas))
19    } else {
20      localStorage.setItem('citas', JSON.stringify([]));
21    }
22  }, [citas, citasIniciales]);
23
24  // Funcion que tome las citas actuales y agregue la nueva
25  const crearCita = (cita) => {
26    guardarCitas([...citas, cita])
27  }
28  // Funcion que elimina una cita por su ID
29  const eliminarCita = (id) => {
30    const nuevasCitas = citas.filter((cita) => cita.id !== id)
```

ESQUEMA
LÍNEA DE TIEMPO

Live Share Quokka

lín. 6, col. 27 Espacios: 4 UTF-8 LF {} JavaScript Go Live Go Live Prettier

ADMINISTRADOR DE PACIENTES

CREAR CITA

Nombre Mascota

Nombre Dueño

Fecha

Hora

Síntomas

AGREGAR CITA

ADMINISTRA TUS CITAS

Mascota: Lulu

Dueño: Juan Sebastian Ardila Hernandez

Fecha: 2023-03-18

Hora: 18:25

Síntomas: Pulgosa, callejera

ELIMINAR »