

UNIVERSIDAD NACIONAL DE INGENIERÍA

FACULTAD DE CIENCIAS

Documentación del proyecto



Título de proyecto: Tienda virtual de productos farmacéuticos con reconocimiento de voz

Curso: Interacción Humano-Computador (CC451A)

Integrantes:

Reyes Gutierrez Jose Daniel 20162732A

Vargas Torres Eros Aylthon 20182203D

Oliva Valdivia Abel Alejandro 20102198I

Yacolca Huaman Jesus Miguel 20183004E

Profesor:

Ciro Nunez

Ciclo: 2021-2

Resumen

FarmUni es una página de ventas de productos tal como los que venderían MiFarma Farmacia Universal o Inkafarma solo que lo novedoso en nuestra aplicación es que implementamos una navegación por voz que va a permitir dirigirse a opciones dadas en la página ya sea perfil, tienda virtual, menú principal solo activando el botón de comenzar reconocimiento de voz y pronunciando el comando respectivo. Las otras funcionalidades que tiene FarmUni son como las de una página de ventas normal ya sean agregar lo que se quiere comprar a un carrito de compras, pagar lo que se agregó al carrito de compras , y por ultimo un apartado de sugerencias donde el cliente pueda darnos alguna recomendación.

Vista Principal

- Mediante nuestro interface de voz implementada gracias a la librería react speech recognizer, el usuario podrá navegar en la página, interactuando con nuestra aplicación.
- Esta interface de voz podrá ser usada en la tienda virtual como filtrador de productos, e incluso se podrá hacer una compra con una simple frase.

Usuarios de la Aplicación:

- **Usuario cliente:**
Este tipo de usuario accederá a nuestra aplicación y podrá navegar gracias a nuestro reconocimiento de voz e incluso seleccionar productos.
Este tipo de usuario puede ser: usuario registrado o usuario sin registrar.
- **Usuario Administrador:**
Este usuario tendrá permisos especiales incluyendo las funciones CRUD para la lista de productos en nuestra tienda Virtual.

Algunas de nuestras actividades:

- Para el usuario sin registrar se crea la opción de registrarse para que puedan ser partícipes de la tienda virtual, en la cual deben colocar sus datos personales.
- El usuario registrado tiene la opción de interactuar con la página web. Esta interacción se basa en poder realizar compras o quitar productos dependiendo de su tipo de registro que tenga. Según este se subdivide en 2 grupos más: usuario cliente y administrador.
- El usuario cliente es capaz de realizar compras en la plataforma mediante un pago en efectivo o su tarjeta de crédito
- También puede contribuir dando sugerencias sobre la experiencia de usuario en la página.
- El usuario administrador tiene la capacidad de agregar o quitar productos de la tienda según la situación lo amerite.

Adicionalmente para una mayor diversidad de usuarios se implementará una interfaz de voz, la cual permitirá una interacción más sencilla y rápida en la página. Un usuario podrá moverse con facilidad a través de la página, además de poder realizar diversas acciones tales como elegir el producto (medicamento) a comprar.

Alcance del proyecto

Llegar a mejorar y optimizar el código ya hecho, hacerlo más accesible y mucho más rápido para pueda ser de ayuda todos nuestros usuarios, en especial a aquellos con alguna discapacidad.

Librerías Usadas

- axios
- bootstrap
- modules
- react
- react-bootstrap
- react-dom
- react-router-dom
- react-scripts
- react-validation
- sweetalert2
- sweetalert2-react-content
- validator
- web-vitals
- react speech recognizer

Librerías Adicionales

- faker
- @fortawesome/fontawesome-svg-core
- @fortawesome/free-solid-svg-icons
- @fortawesome/react-fontawesome

Restricciones de nuestro Proyecto:

- 1) Nos enfocamos más en frontend más que en el backend para poder hacer los requerimientos del reconocimiento de voz.
- 2) Nos centramos en el cliente más que administrador, en otras palabras no se implementaron funciones CRUD para la tienda virtual modo administrador.
- 3) No se implementó una base de datos para almacenar los mensajes de las sugerencias, lo cual se hará más adelante.

- 4) No se llegó a completar un carrito de compras, donde el usuario cliente pueda ver sus productos comprados y el historial de compra.
- 5) En la tienda virtual nos faltó un filtrador de productos y una función agregar que funcionara con el código de voz.
- 6)

Pruebas Hechas

- Sujeto Pruebas 1: Karina Torres:

Correo: jtorres@mail.upla.edu.pe

Comentario o Sugerencia: Me pareció novedoso y muy interesante, si tuviera que corregir o aumentar algo sería el color, y que los productos se pudieran elegir de manera individual con el micrófono.

- Sujeto Pruebas 2: Daniel Reyes:

Correo: dreyesg@outlook.es

Comentario o Sugerencia: Me hubiera gustado que haya una función de agregar al carrito mediante comandó de voz.

- Sujeto Pruebas 3: Isabel Vega

Correo: vegaisaperu01@gmail.com

Comentario o Sugerencia: Me pareció interesante la aplicación web, sobretodo en tiempos de pandemia, sería de mucha utilidad tener categorías de los productos a comprar.

Link de los videos de las pruebas Hechas

<https://drive.google.com/drive/folders/12bwuAm6x3lM7fCktRdaT7sK7KsBttpwK>

Forma de Inicializar el Proyecto Farmuni

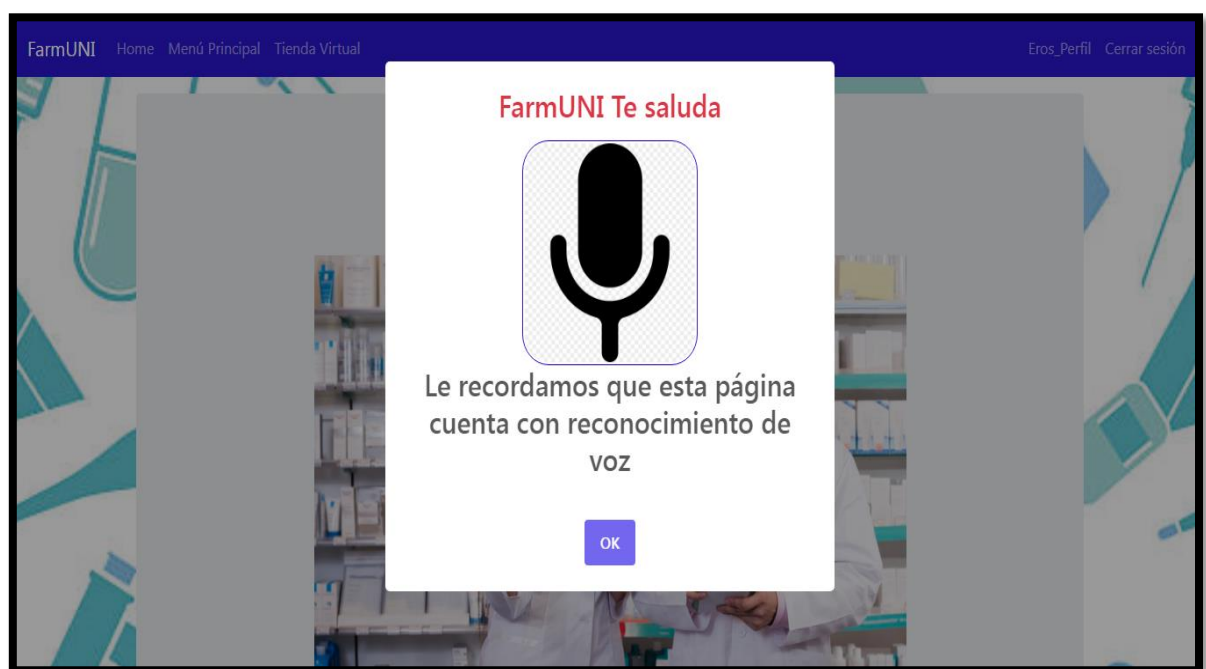
- Acceder a dos consolas diferentes y entra con cd backend y cd frontend respectivamente
- En la primera consola ejecutar los comando yarn install
- Una vez terminada de cargar las dependencias del backend, usar el comando yarn start , esto conectara la base de datos a mongo atlas.
- En la segunda consola ejecutar los comando yarn install
- Una vez terminada de cargar las dependencias del frontend, usar el comando yarn start, esto iniciara la página.

Interface e Imágenes

1) Inicio de pantalla fase 1



2) Inicio de pantalla fase 2



3) Página Login

FarmUNI Home Iniciar sesión Registrarse

Nombre de usuario

Contraseña

Iniciar sesión

4) Página Registro

FarmUNI Home Iniciar sesión Registrarse

Nombre

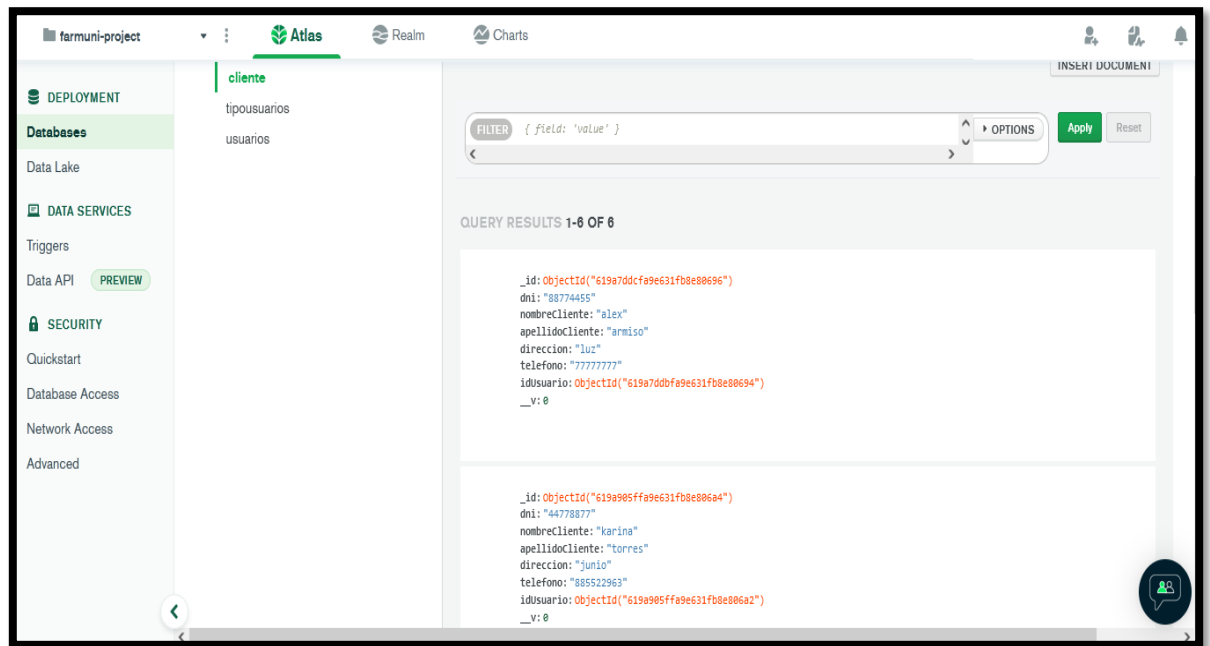
Apellido

Correo

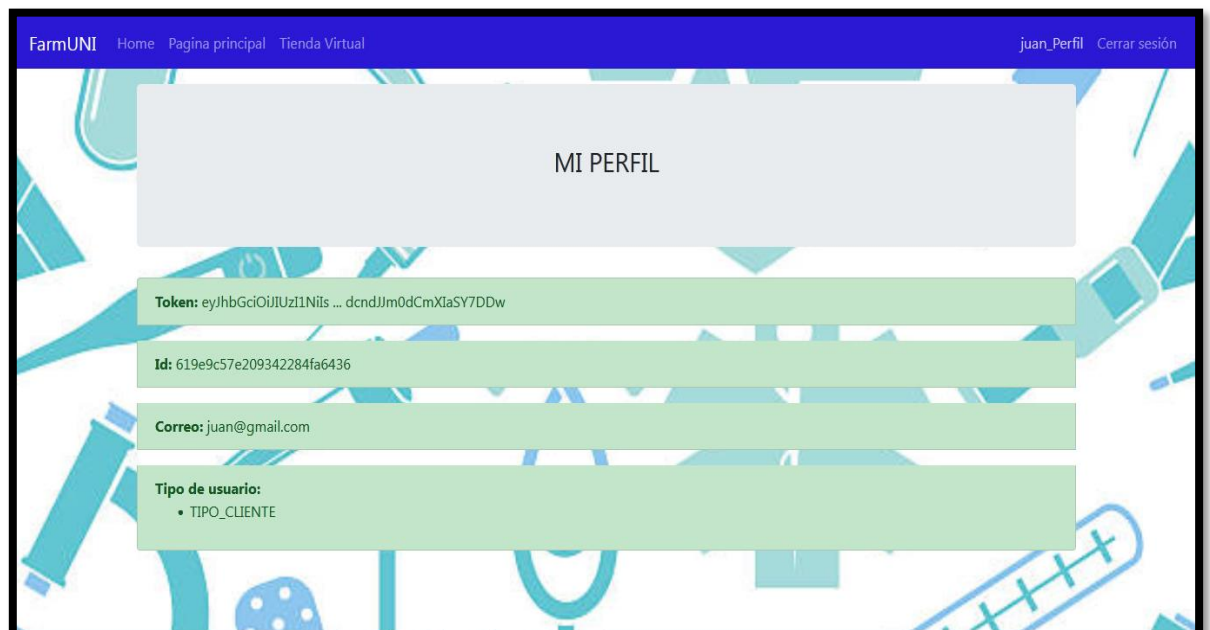
Telefono

DNI

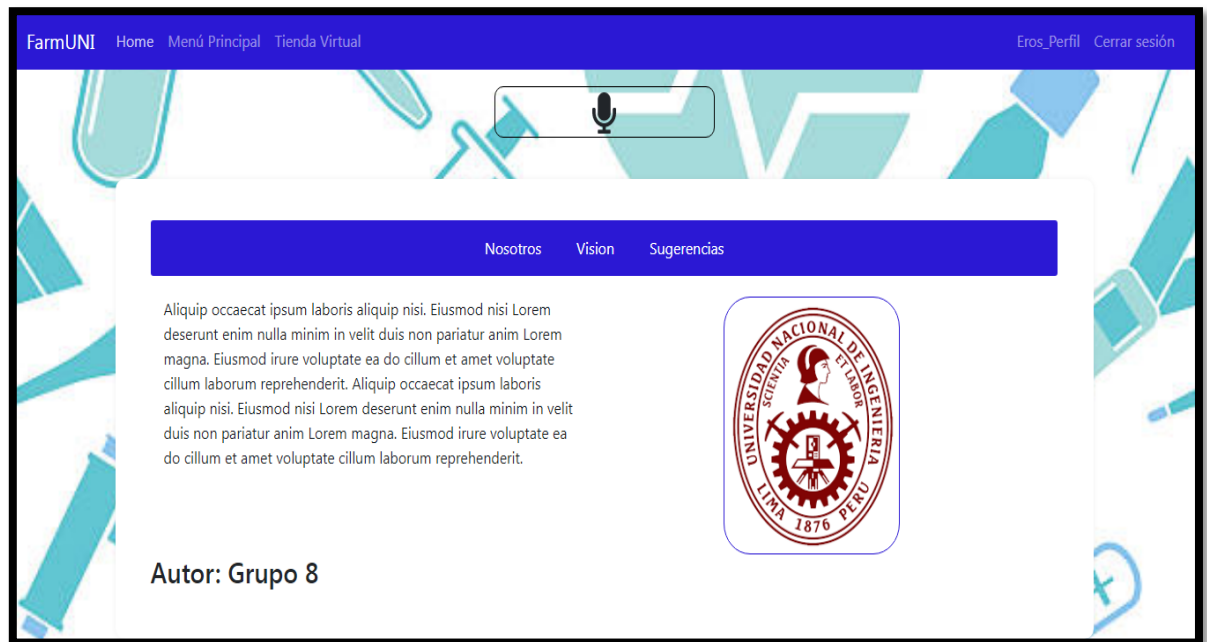
5) Conexiones a Mongo Atlas:



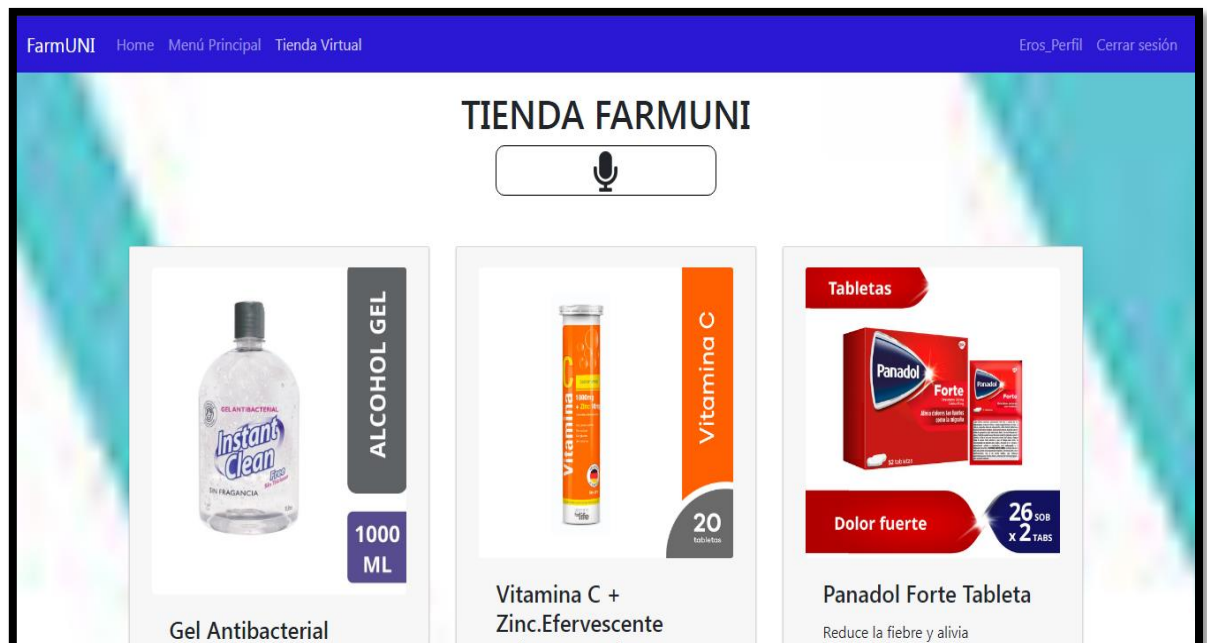
6) Perfil del Usuario

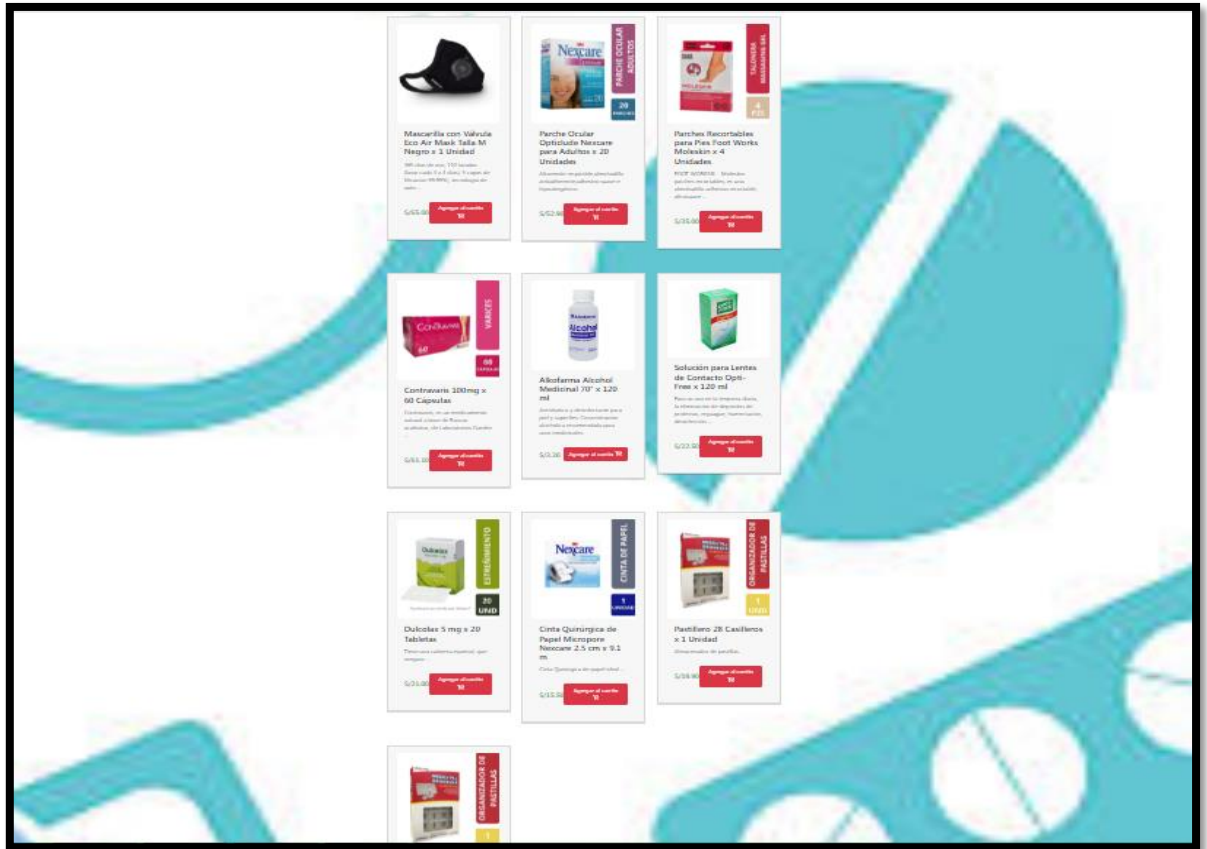


7) Menú Principal



8) Página de la tienda Virtual





9) Código de la Data ingresada manualmente : 40 productos

```

458 //producto 39
459 {
460   descripción: 'Almacenador de pastillas.',
461   nombreProducto: 'Pastillero 28 Casilleros x 1 Unidad',
462   stock: '1000',
463   etiquetas: 'No',
464   precio: '5/19.90',
465   tipoProducto: 'salud',
466   requiereReceta: 'No',
467   imgSrc: '/assets/images/productos/39.jpg',
468   idCategoria: 'null',
469 },
470 //producto 40
471 {
472   descripción: 'INSULINA GLARGINA 100 U/mL y LIXISENATIDA 33 µg/mL SoloStar Solución inyectable, en plum
473   nombreProducto: 'Soliqua (30-60) Solución Inyectable',
474   stock: '1000',
475   etiquetas: 'No',
476   precio: '5/212.90',
477   tipoProducto: 'salud',
478   requiereReceta: 'No',
479   imgSrc: '/assets/images/productos/39.jpg',
480   idCategoria: 'null',
481 },
482 ];
483

```

PROBLEMAS 1 OUTPUT TERMINAL CONSOLA DE DEPURACIÓN

Search for the **keywords** to learn more about each warning.
To ignore, add `// eslint-disable-next-line` to the line before.

node node

10) Data guardada para la creación del carrito de compras (efecto de agregar):



11) Código del reconocimiento de voz 1ra fase

```
File Edit Selection View Go Run Terminal Help
Voz.jsx - IHC - Visual Studio Code

EXPLORER
  OPEN EDITORS
    Voz.jsx app-voz-app-vo... U
  IHC
    src
      App.css
      App.js
      App.test.js
      index.css
      index.js
      logo.svg
      Main.jsx
      reportWebVitals.js
      Rutas.jsx
      setupTests.js
      Vista1.jsx
      Vista2.jsx
      Voz.css
      Voz.jsx
      .gitignore
      package-lock.json
      package.json
      README.md
      yarn.lock
    > ionic-angular-news
    > my-react-app
    > photo-gallery
  OUTLINE
  TIMELINE

Voz.jsx
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50

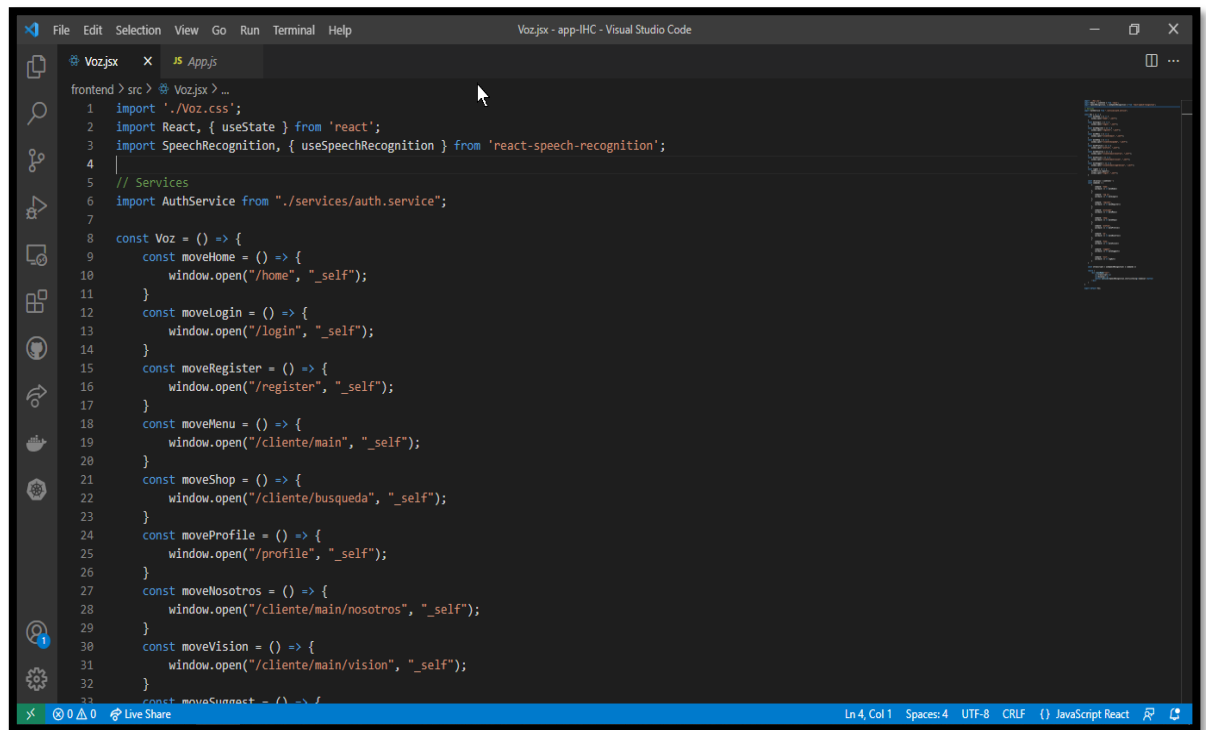
const [display] = useState('')
const commands = [
  {
    command: 'number one',
    callback: () => movervista1()
  },
  {
    command: 'number two',
    callback: () => movervista2()
  },
  {
    command: 'main page',
    callback: () => movermain()
  }
]

const {transcript} = useSpeechRecognition({ commands })

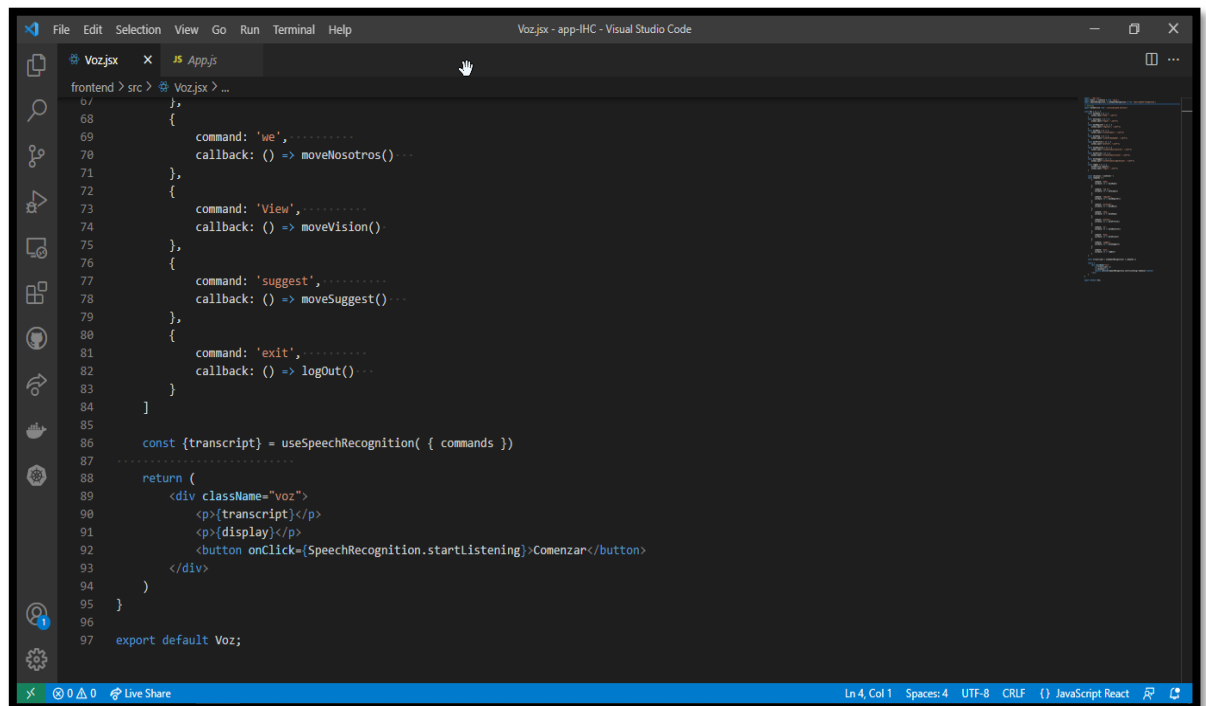
return (
  <div className="voz">
    <h3>Speak to navigate</h3>
    <p>{transcript}</p>
    <p>{display}</p>
    <button onClick={SpeechRecognition.startListening}>Comenzar</button>
    <button onClick={SpeechRecognition.stopListening}>Detener</button>
  </div>
)

export default Voz;
```

12) Código del reconocimiento de voz 2da fase

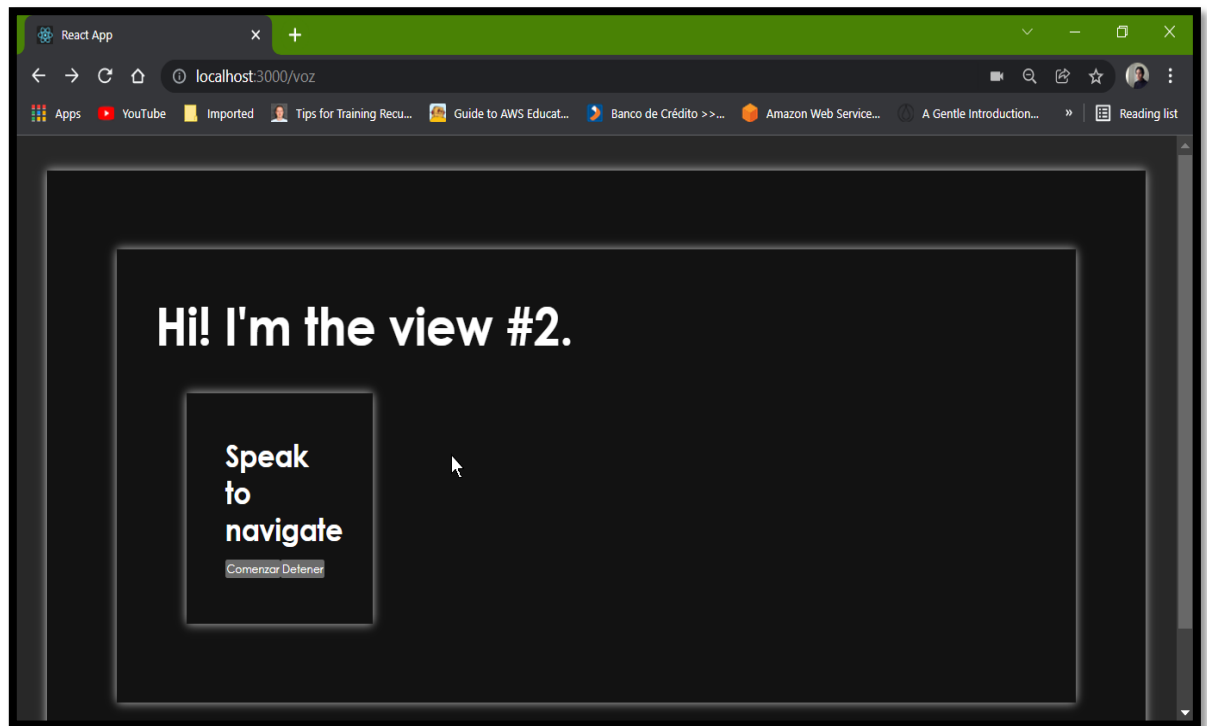


```
frontend > src > Vozjsx > ...
1 import './Voz.css';
2 import React, { useState } from 'react';
3 import SpeechRecognition, { useSpeechRecognition } from 'react-speech-recognition';
4
5 // Services
6 import AuthService from './services/auth.service';
7
8 const Voz = () => {
9   const moveHome = () => {
10     window.open("/home", "_self");
11   }
12   const moveLogin = () => {
13     window.open("/login", "_self");
14   }
15   const moveRegister = () => {
16     window.open("/register", "_self");
17   }
18   const moveMenu = () => {
19     window.open("/cliente/main", "_self");
20   }
21   const moveShop = () => {
22     window.open("/cliente/busqueda", "_self");
23   }
24   const moveProfile = () => {
25     window.open("/profile", "_self");
26   }
27   const moveNosotros = () => {
28     window.open("/cliente/main/nosotros", "_self");
29   }
30   const moveVision = () => {
31     window.open("/cliente/main/vision", "_self");
32   }
33   const moveSuggest = () => {
```



```
67   },
68   {
69     command: 'we', .....
70     callback: () => moveNosotros() ...
71   },
72   {
73     command: 'View', .....
74     callback: () => moveVision() ...
75   },
76   {
77     command: 'suggest', .....
78     callback: () => moveSuggest() ...
79   },
80   {
81     command: 'exit', .....
82     callback: () => logout() ...
83   }
84 ]
85
86 const {transcript} = useSpeechRecognition( { commands })
87 .....
88 return (
89   <div className="voz">
90     <p>{transcript}</p>
91     <p>{display}</p>
92     <button onClick={SpeechRecognition.startListening}>Comenzar</button>
93   </div>
94 )
95 }
96
97 export default Voz;
```

13) Interface del reconocimiento de voz 1ra fase



14) Interface del reconocimiento de voz 2da fase "Con el botón Comenzar"



FarmUNI

HomeMenú PrincipalTienda Virtual

Eros_PerfilCerrar sesiónComenzar

NosotrosVisionSugerencias

Si, Usted tiene alguna "QUEJA" o "SUGERENCIAS" haganoslo saber

Llene el espacio en blanco

Escriba en esta Area

Enviar

FarmUNI

HomeMenú PrincipalTienda Virtual

Eros_PerfilCerrar sesiónComenzar

MI PERFIL

Token:eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXZWZw--_qsPt5auw

Id:61ad8dc34e1cf8ba6ceb

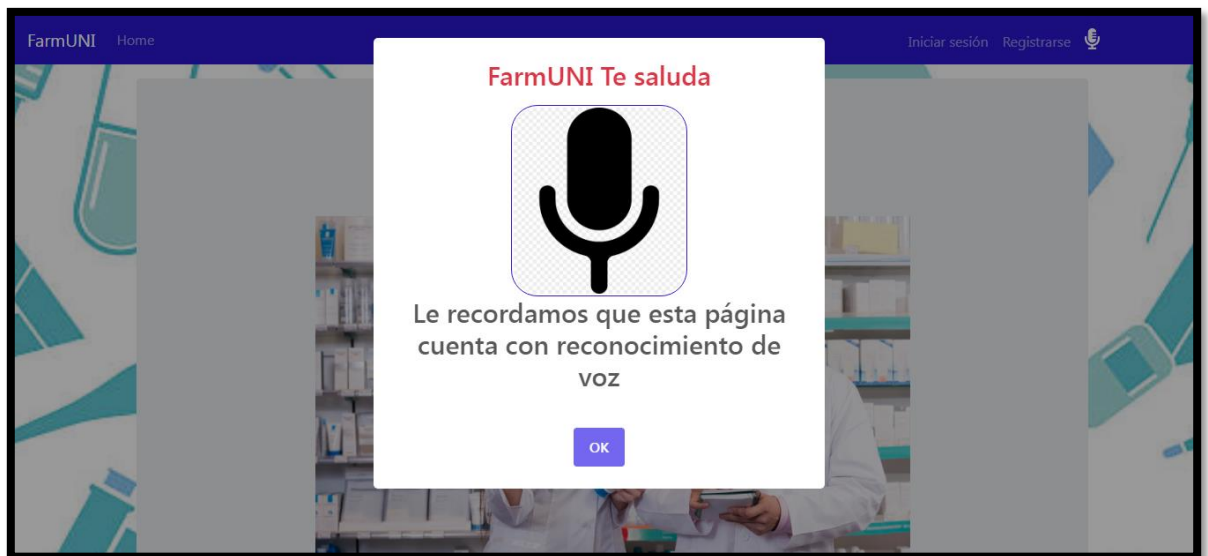
Correo:correo1@gmail.com

Tipo de usuario:

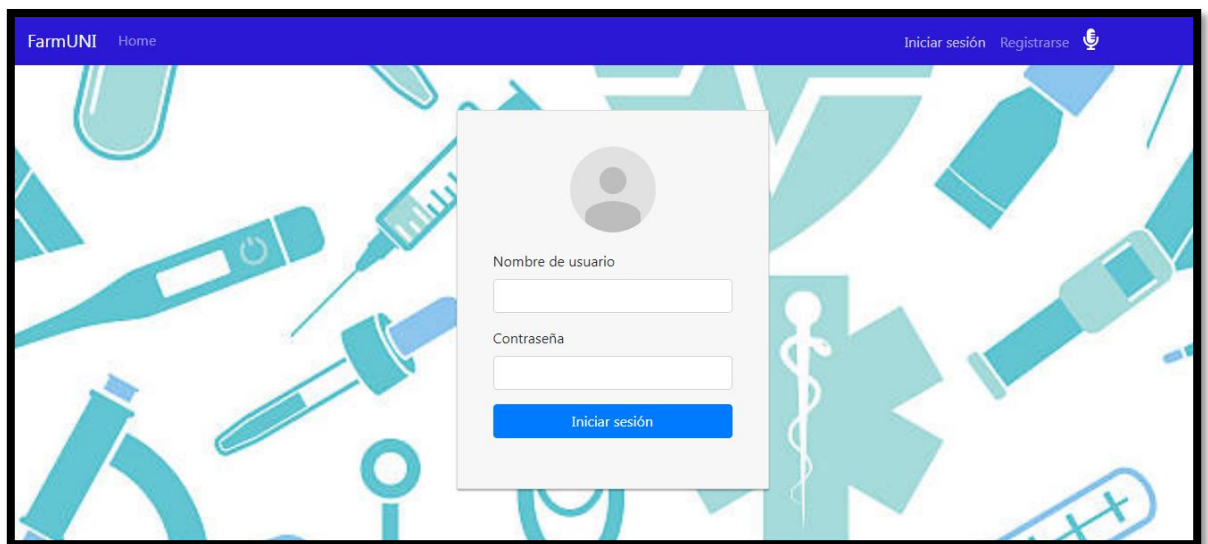
- TIPO_CLIENTE

Imágenes de Proyecto Finalizado con la implementación de la Interface de voz:

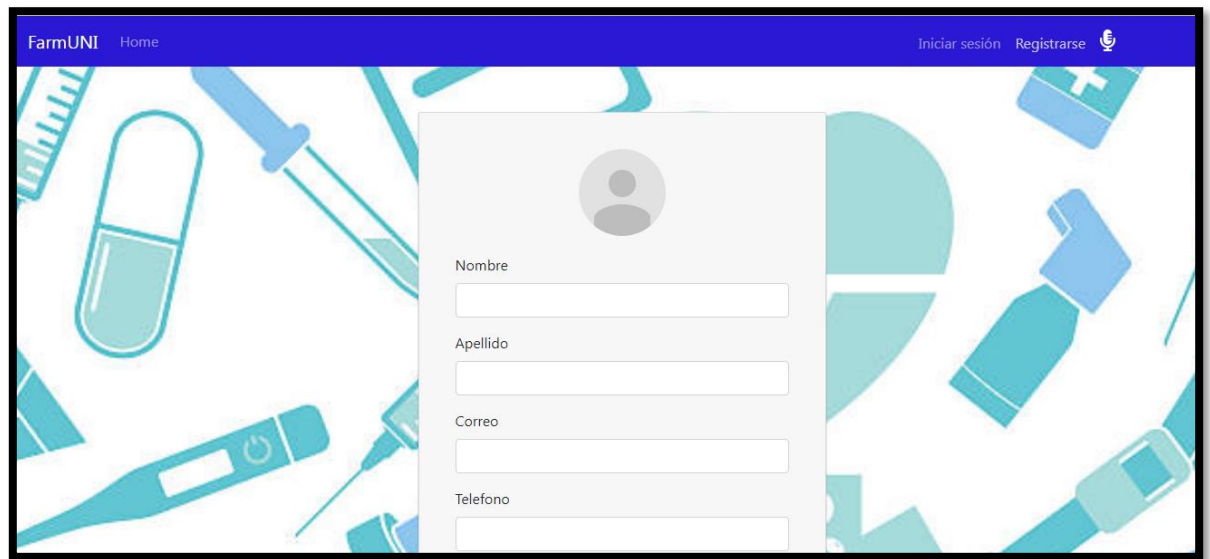
- 1) Imagen de Presentación con interface de voz “home”



- 2) Imagen de Iniciar Sesión con interface de voz “Login”

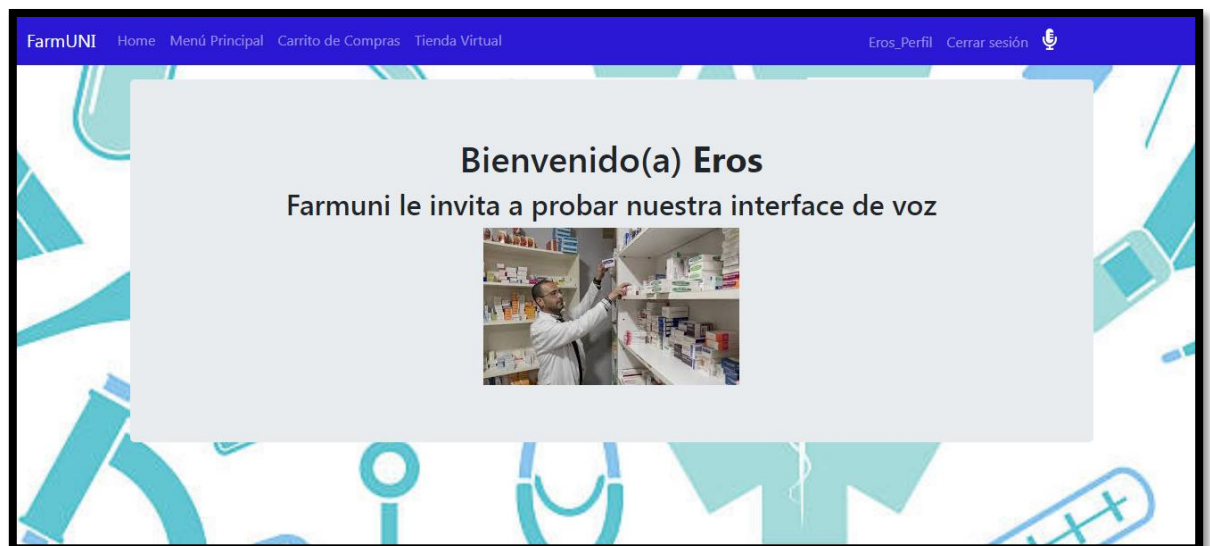


3) Imagen de Registro con interface de voz “registrarse”

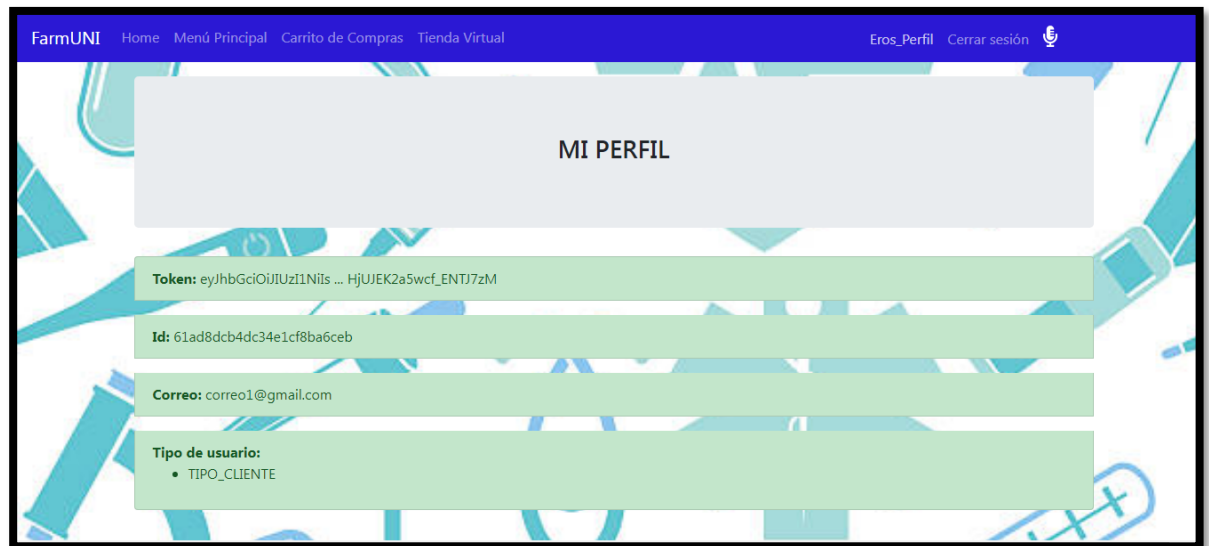


The screenshot shows the registration page of FarmUNI. The header is blue with the text "FarmUNI Home" on the left and "Iniciar sesión Registrarse" on the right, accompanied by a microphone icon. The background is a light blue pattern of medical icons. A central white form contains a grey circular profile icon placeholder and four input fields labeled "Nombre", "Apellido", "Correo", and "Telefono".

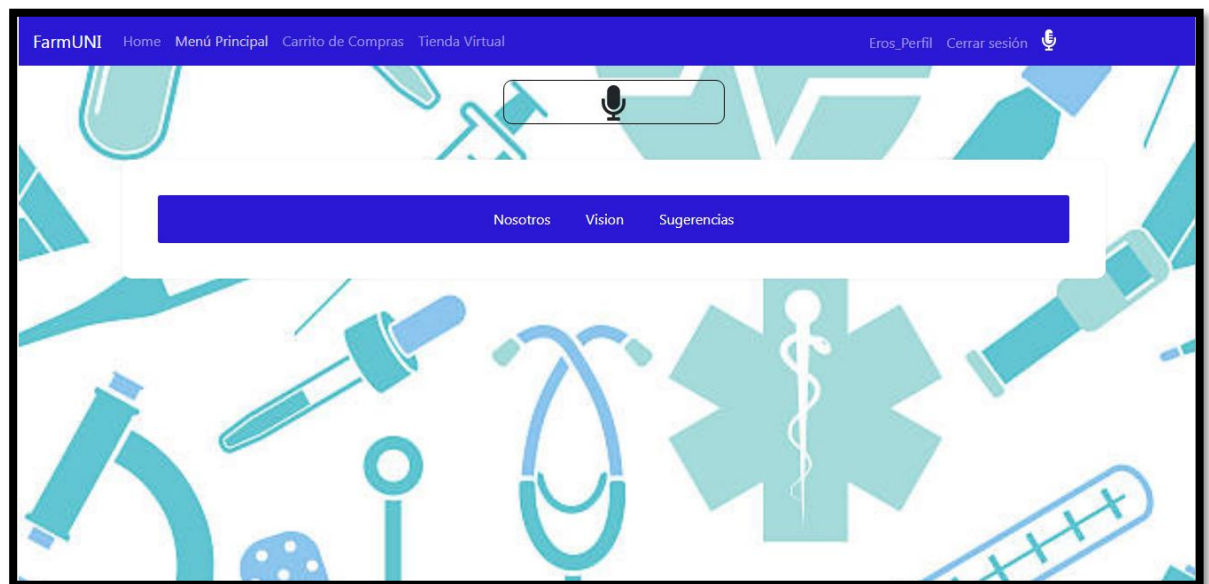
4) Imagen de Bienvenida al Usuario



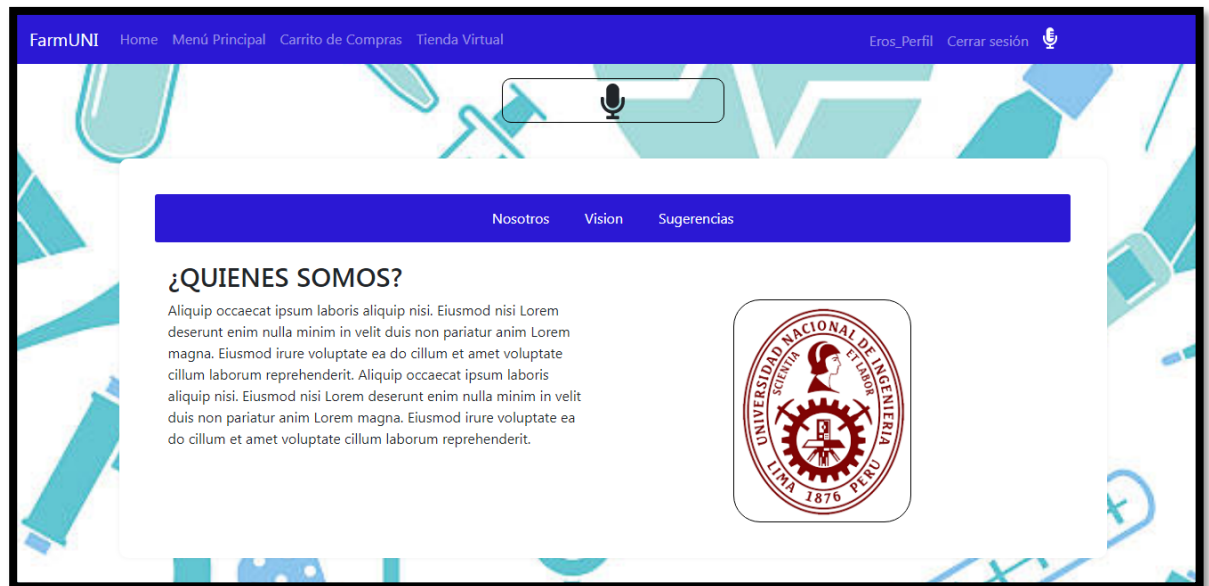
5) Imagen de perfil del cliente con interface de voz “perfil”



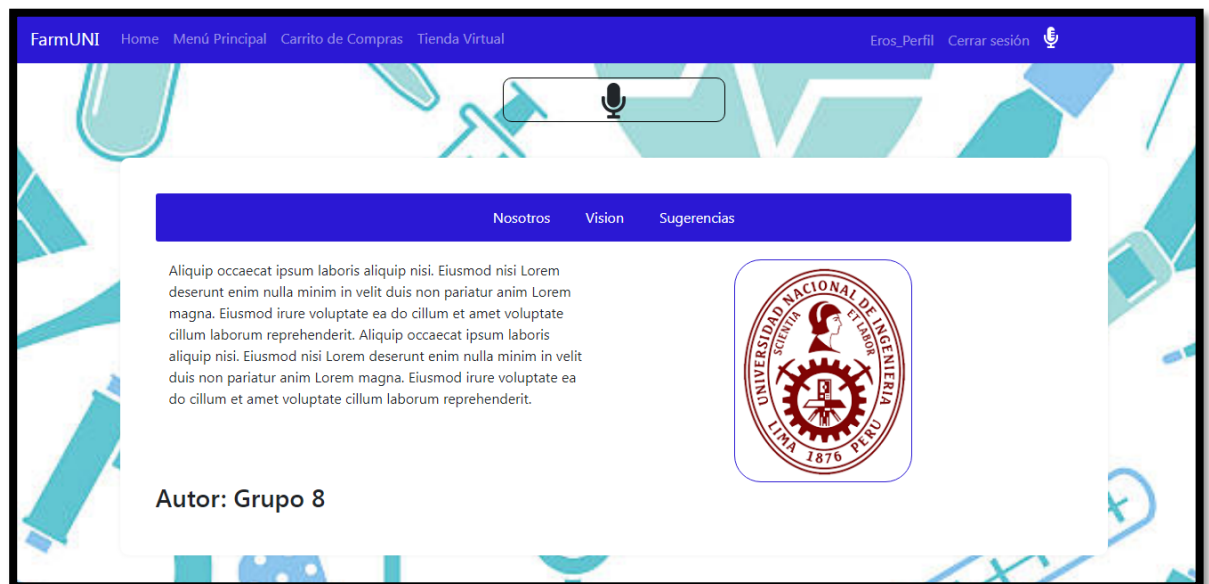
6) Imagen de página de menú principal del cliente con interface de voz “principal”



7) Imagen de página de menú principal (nosotros) con interface de voz “nosotros”



8) Imagen de página de menú principal (visión) con interface de voz “visión”



- 9) Imagen de página de menú principal (sugerencias) con interface de voz “sugerencias”



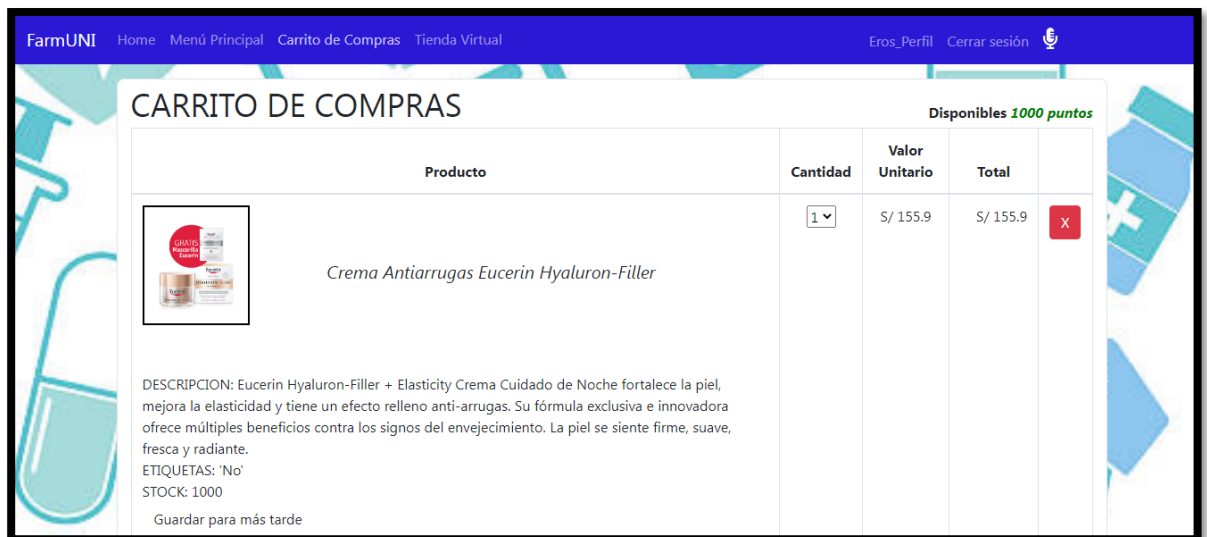
- 10) Imagen de la tienda virtual con interface de voz “tienda”



11) Imágenes enviadas al localStorage



12) Imágenes del carrito de compras



ANEXOS

Data falsa para las pruebas:

```
JS CarritoCompras.js 3, U    JS mockData.js U X
farmuni_app > frontend > src > componentes > BoardCliente > JS mockData.js > ...
1  // Mock Data
2  import faker from "faker";
3
4  export const mockData = (count) => {
5    const result = [];
6    for (let i = 0; i < count; i++) {
7      // Datos
8      const mockProducto = {
9        descripcion: faker.commerce.productDescription(),
10       nombreProducto: faker.commerce.productName(),
11       stock: faker.datatype.number(1000),
12       etiquetas: faker.lorem.text(),
13       precio: faker.commerce.price(1.5, 1000, 2, "S/ "),
14       tipoProducto: faker.lorem.text(10),
15       requiereReceta: faker.lorem.text(5),
16       imgSrc: `/assets/images/productos/${i + 1}.jpg`,
17       idCategoria: faker.random.alphaNumeric(),
18     };
19
20     result.push(mockProducto);
21   }
22
23   return result;
```

Confirmación de las librerías Usadas:

```
JS CarritoCompras.js 3, U {} package.json U X
farmuni_app > frontend > {} package.json > {} browserslist > [ ] development > 2
1 {
2   "name": "frontend",
3   "version": "0.1.0",
4   "private": true,
5   "dependencies": {
6     "@fortawesome/fontawesome-svg-core": "^1.2.36",
7     "@fortawesome/free-solid-svg-icons": "^5.15.4",
8     "@fortawesome/react-fontawesome": "^0.1.16",
9     "@testing-library/jest-dom": "^5.11.8",
10    "@testing-library/react": "^11.2.2",
11    "@testing-library/user-event": "^12.6.0",
12    "axios": "^0.21.1",
13    "bootstrap": "^4.6.0",
14    "faker": "^5.5.3",
15    "modules": "^0.4.0",
16    "react": "^17.0.1",
17    "react-bootstrap": "^1.4.3",
18    "react-dom": "^17.0.1",
19    "react-google-maps": "^9.4.5",
20    "react-router-dom": "^5.2.0",
21    "react-scripts": "4.0.1",
22    "react-speech-recognition": "^3.9.0",
23    "react-validation": "^3.0.7",
24    "styled-components": "^5.3.3",
25    "sweetalert2": "^11.2.0",
26    "sweetalert2-react-content": "^4.2.0",
27    "validator": "^13.5.2",
28    "web-vitals": "^0.2.4"
29  },
  Depuración de >
30  "scripts": {
31    "start": "react-scripts start",
32    "build": "react-scripts build",
33    "test": "react-scripts test",
34    "eject": "react-scripts eject"
35  },
36  "eslintConfig": {
37    "extends": [
38      "react-app",
39      "react-app/jest"
40    ]
41  },
42  "browserslist": {
```

Código de la interface de voz corregida al español

Parte 1:

```
JS CarritoCompras.js 3, U  Voz.jsx U X
farmuni_app > frontend > src > Voz.jsx > [Voz]
1  import './Voz.css';
2  import React, { useState } from 'react';
3  import SpeechRecognition, { useSpeechRecognition } from 'react-speech-recognition';
4
5
6  import styled from 'styled-components';
7  import { FontAwesomeIcon } from '@fortawesome/react-fontawesome'
8  import { faMicrophoneAlt } from '@fortawesome/free-solid-svg-icons'
9
10 // Services
11 import AuthService from './services/auth.service';
12
13 const Voz = () => {
14   const moveHome = () => {
15     window.open("/home", "_self");
16   }
17   const moveLogin = () => {
18     window.open("/login", "_self");
19   }
20   const moveRegister = () => {
21     window.open("/register", "_self");
22   }
23   const moveMenu = () => {
24     window.open("/cliente/main", "_self");
25   }
26   const moveShop = () => {
27     window.open("/cliente/busqueda", "_self");
28   }
29   const moveProfile = () => {
30     window.open("/profile", "_self");
31   }
32   const moveNosotros = () => {
33     window.open("/cliente/main/nosotros", "_self");
34   }
35   const moveVision = () => {
36     window.open("/cliente/main/vision", "_self");
37   }
38   const moveSuggest = () => {
39     window.open("/cliente/main/sugerencias", "_self");
40   }
41   const logOut = () => {
42     AuthService.logout();
43   }
44 }
```

Parte 2:

```
JS CarritoCompras.js 3, U  Vozjsx U X
farmuni_app > frontend > src > Vozjsx > Voz

46
47 const [display] = useState('')
48 const commands = [
49   {
50     command: 'home',
51     callback: () => moveHome()
52   },
53   {
54     command: 'login',
55     callback: () => moveLogin()
56   },
57   {
58     command: 'registrarse',
59     callback: () => moveRegister()
60   },
61   {
62     command: 'principal',
63     callback: () => moveMenu()
64   },
65   {
66     command: 'tienda',
67     callback: () => moveShop()
68   },
69   {
70     command: 'perfil',
71     callback: () => moveProfile()
72   },
73   {
74     command: 'nosotros',
75     callback: () => moveNosotros()
76   },
77   {
78     command: 'visión',
79     callback: () => moveVision()
80   },
81   {
82     command: 'sugerencias',
83     callback: () => moveSuggest()
84   },
85   {
86     command: 'salir',
87     callback: () => logOut()
88   }
89 ]
```