

SPRINT 3 – DESARROLLANDO EL FRONTEND

EQUIPO: U29/10

MOCKUP DE LAS INTERFACES DE USUARIO:

Captura de pantalla de los mockups o diseños planteados para el frontend del proyecto. La herramienta para el desarrollo de estas está abierta a preferencia, se sugiere draw.io

The image shows a user registration form mockup. At the top, there is a dark grey navigation bar (Navbar) with links for Home, Features, Pricing, and About. A search bar is located on the right side of the navbar. Below the navbar, the form is organized into two columns. The left column contains fields for: Id Usuario, Clave, Nombre, Tipo de documento, Dirección, and Email address (with the placeholder email@example.com). The right column contains fields for: Usuario, Tipo de usuario, Apellidos, Número de documento, Teléfono, Estado, and Password. At the bottom of the form, there is a prominent blue button labeled "Sign in".

Registro de proveedores

Cedula

Nombre completo

Dirección

Teléfono

Email address

Contraseña

Confirmar contraseña

Registrarse

Registro de productos

Id producto

Nombre

Código

Unidad de medida

Valor de la compra

Cantidad

Categoría

Registrar producto

REGISTRO DE CALIFICACIÓN

id Usuario

id Producto

id Calificación

Descripción

Notas

[Regístrese](#)

EVIDENCIA DEL FRONTEND EN FUNCIONAMIENTO:

Capturas de pantalla de las vistas del frontend en funcionamiento, consumiendo servicios del backend

localhost:3000/usuarios

Extensión: Entrar a... PARA ESCUCHAR... Educación Comfena... Facebook - Entra o... Iniciar sesión en Ou... Semillero de litigio... UNLAVIRTUAL...

M DULCE Online

Usuario

Usuario: Clave:

Rol: Administrador Cliente Transportador

Nombres: Apellidos:

Tipo de Documento: Número de Documento:

Dirección: Teléfono:

Email:

Estado: Activo Inactivo

[Guardar](#) [Limpiar](#)

Usuario	Clave	Rol	Nombres	Apellidos	Tipo de Documento	Número de Documento	Dirección	Teléfono	Email	Estado	Acciones
Regicas	lg1234	Administrador	Liebeth	Grisales Castro	CC	1010206480	Calle 40 a sur No. 34b-1	3330802	regicas@gmail.com	Activo	Eliminar Seleccionar
JhonT	12345	Usuario	Jhon	Lopez Tabares	CC	43420730	Calle 40A Sur No. 34A3	3330802	juhonn@gmail.com	Activo	Eliminar Seleccionar

Windows taskbar with icons for File Explorer, Settings, Chrome, Word, Teams, Excel, Outlook, Camera, and other applications. System tray shows time 21:51 and date 06/11/2022.

localhost:3000/proveedores

Extensión: Entrar a... PARA ESCUCHAR... Educación Comfena... Facebook - Entra o... Iniciar sesión en Ou... Semillero de litigio... UNLAVIRTUAL...

M DULCE Online

Proveedor

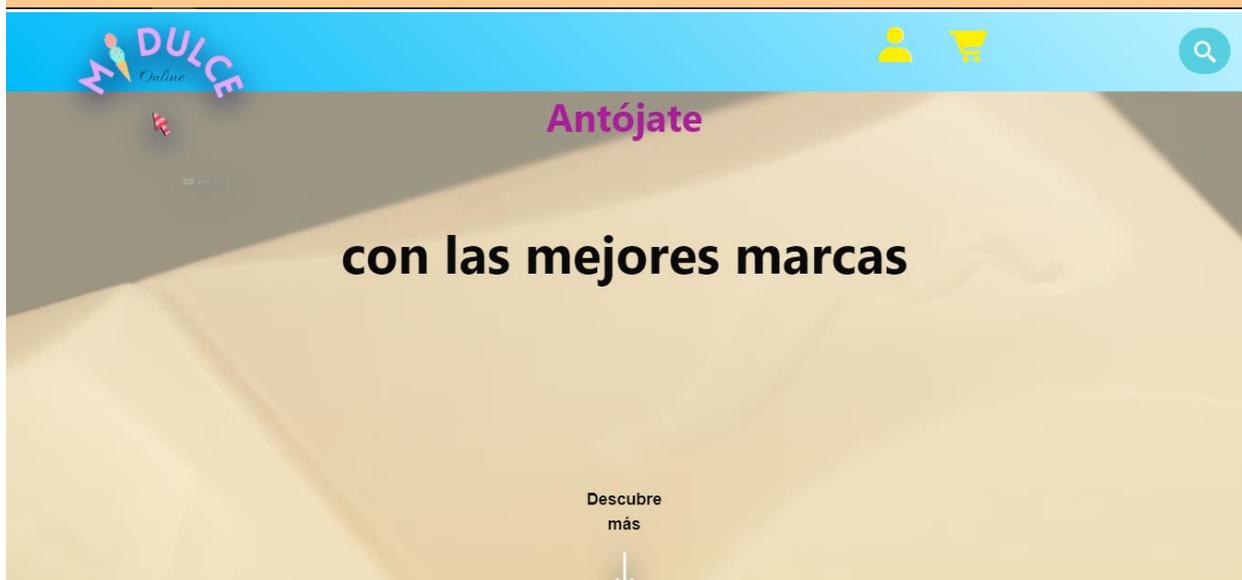
Nombres: Tipo de Documento:

Número de Documento: Dirección:

Teléfono: Email:

[Guardar](#) [Limpiar](#)

Nombres	Tipo de Documento	Número de Documento	Dirección	Teléfono	Email	Acciones
---------	-------------------	---------------------	-----------	----------	-------	----------





Bon Bon Bum Surtido
Colombina 24 Und x 456 g

\$8.600 [Ver más](#) \$909.000

Añadir al carrito



Coffee Delight Blando
Colombina 10 Und x 80 g

\$12.900 [Ver más](#) \$15.800

Añadir al carrito



Chicle Splot Sandia Acid
Colombina 10 Und x 80 g

\$6.400 [Ver más](#) \$10.800

Añadir al carrito



Chocolate Blanco y almendras
Mont Blanc 12 Und x 80 g

\$70.000 [Ver más](#) \$90.000

Añadir al carrito



Chocolate Blanco con Fresas
Mont Blanc 10 Und x 80 g

\$87.700 [Ver más](#) \$90.000

Añadir al carrito



Chocolate y Macadamia
Mont Blanc 10 Und x 80 g

\$23.950 [Ver más](#) \$90.800

Añadir al carrito



Galleta deditos
Nestle 10 Und x 80 g

\$9.890 [Ver más](#) \$2.000

Añadir al carrito



Galleta milo
Nestle 10 Und x 80 g

\$8.650 [Ver más](#) \$9.190

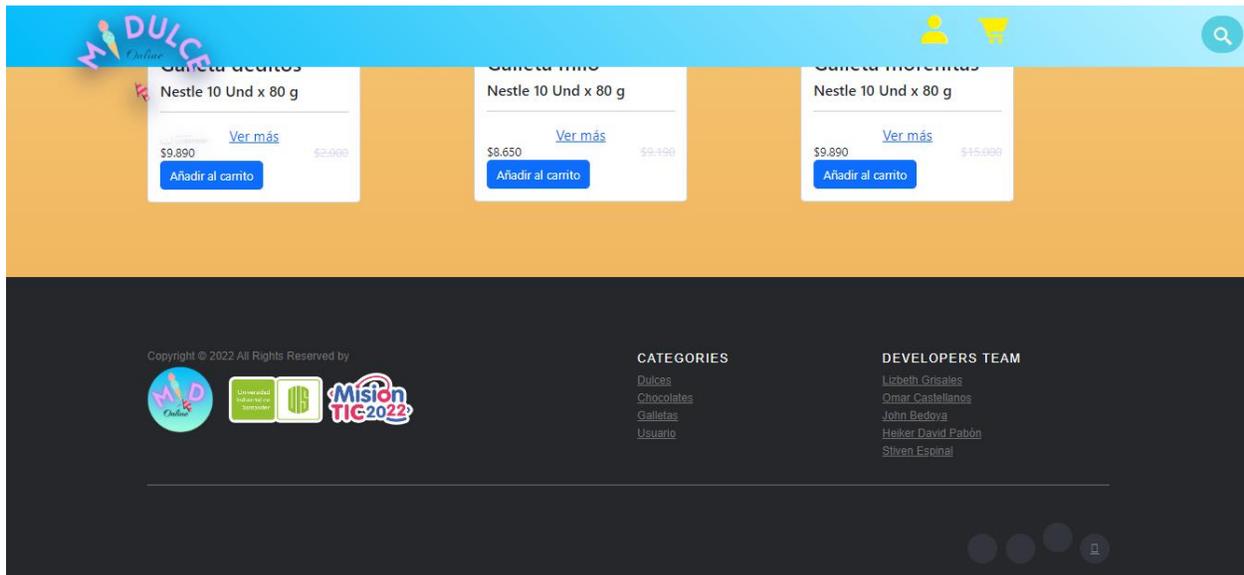
Añadir al carrito



Galleta morenitas
Nestle 10 Und x 80 g

\$9.890 [Ver más](#) \$15.000

Añadir al carrito

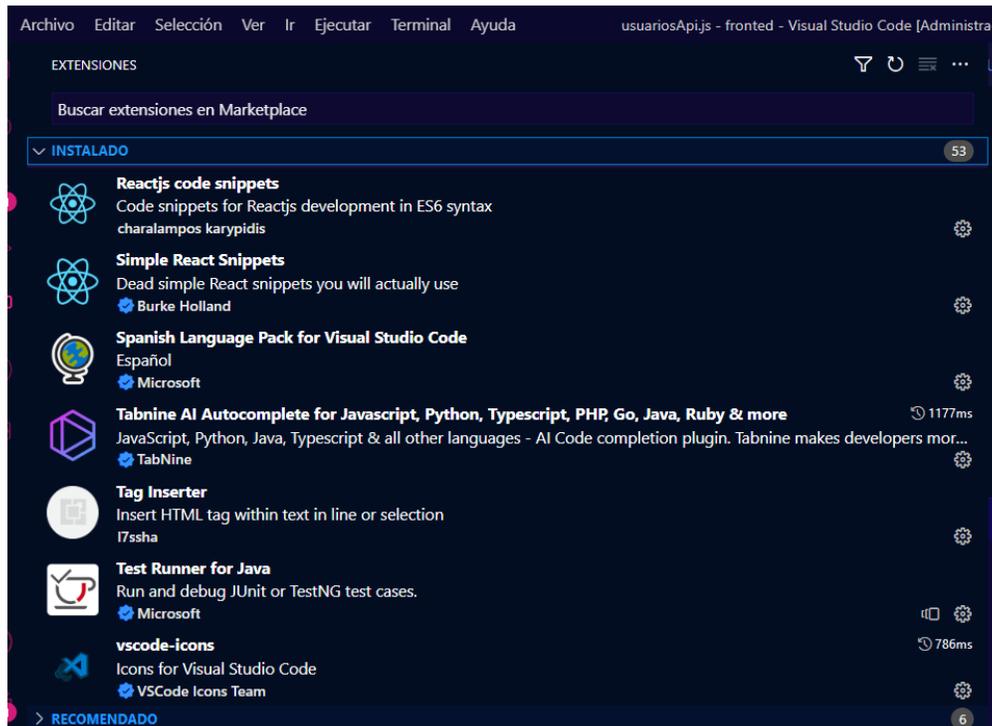


DOCUMENTACIÓN DEL PROCESO REALIZADO:

Descripción detallada del proceso realizado para el desarrollo del frontend, incluyendo las dificultades presentadas

1. Se instalan las siguientes dependencias en visual estudio Code, para el entorno de desarrollo

Nombre	Autor
Atom One Dark Theme	Mahmoud Ali
Auto Import	steoates
Bootstrap 5 & Font Awesome Snippets	HansUXdev
Bracket Pair Colorizer 2	CoenraadS
ES7+ React/Redux/React-Native snippets	dsznajder
ESLint	Microsoft
Simple React Snippets	Burke Holland
html to JSX	Riaz Laskar
vscode-icons	VSCoDe Icons Team
Prettier	Code formatter Prettier
React Native Tools	Microsoft



2. CREACION DE PROYECTO REACT LOCAL: Se crea repositorio local con el nombre del proyecto `fronted`, en minúsculas, en la librería de React js (se debe tener `nodejs` instalado previamente)

a) Se inicializa la terminal, y se ejecuta el siguiente comando:

```
npx create-react-app ./ o npx create-react-app nombrecarpet
```

- b) Se instalan todos los `packages` o dependencias del entorno `nodejs`, necesarias para el desarrollo, la producción, la depuración entre otros.
- c) Inicializamos nuestro proyecto con el comando `npm init`, para inicializar el servidor de `nodejs`
- d) Se despliega el `index` del proyecto en el navegador predeterminado según el puerto disponible en el servidor.
- e) Eliminamos las estructuras predeterminadas en los archivos de `index.css`, `index.js`, y de `App.js` dejamos solo la estructura de la función, ya que crearemos nuestro propio `templete`.
- f) Instalación de la dependencia `router` con el comando `npm install --save react-router-dom`, paquete que nos permite realizar la navegación entre componente o páginas del proyecto.

3. CREACION DE ESTRUCTURA DEL PROYECTO:



- a) Creamos 2 carpetas en la matriz del proyecto, llamadas components (bloques de código que integran la interfaz gráfica de la app web y que se pueden reutilizar en todo el proyecto); dentro de cada una respectivamente se crearon las subcarpetas de home, proveedores y usuarios con la extensión js, además de usuariosapis.js
 - b) Se crea la carpeta para consumir servicios con el nombre apis (tiene los componentes que integran el CRUD prefabricado del proyecto, que facilitan el desarrollo de los componentes, además de integrar la ruta del backend creado previamente en el Sprint 2); en el archivo configrs exportamos el módulo de la url del backend, lo que asegura que al consumir los servicios del API como el GET (traer consulta), POST (crear registro), PUT (actualizar registro) y el DELETE (eliminar registro), métodos requeridos para realizar las peticiones al servidor desde una conexión sincrónica, y así poder actualizar la información en el entorno Frontend automáticamente.
4. CREACION DE COMPONENTE APP: El componente de arranque o de inicio de la aplicación es el archivo App.js ya creado en la instalación, el cual permite al renderizar la página, cargar las rutas que integran el proyecto y se relacionan en este archivo, es decir que en el mismo se importan las direcciones de las páginas que componen el proyecto.
 5. CREACION DE CLASE VISTA para consumir los servicios comunes de un API prefabricados en el CRUD del servidor, creamos un archivo con el nombre usuariosApi.js, que integra la importación de la ruta del backend definida en el archivo configrs.js, las cabeceras del API, y los métodos requeridos para la funcionalidad del CRUD que se podrán importar en otros componentes para dotarlos de funcionalidad y no tener que crearlos desde 0 cuando consumamos un servicio del backend; este archivo integra además los métodos asíncronos, como queryParams, y todos los necesarios para consumir el servicio.
 6. CREACION DE ESTRUCTURA DE COMPONENTES:
 - a) Importamos los componentes del ruteo y de todos los paquetes necesarios como Import React, en cada archivo js según las necesidades de desarrollo.
 - b) Definimos la función de flecha (sfc + enter) que contiene un return que no debe estar vacío para que no genere un error y debe contener un código JSX, que es el lenguaje propio de React para maquetar en esta tecnología los elementos que se desea que el usuario visualice en la interfaz gráfica; este JSX integra lenguaje JS y HTML y deberá contener un componente

primario que contenga todo el código, por ejemplo una etiqueta div que contengan la dependencia Routes.

```
import React, {Fragment} from 'react';
import { BrowserRouter as Router, Route, Routes } from 'react-router-dom';

function App() {
  return (
    <Fragment>
      <Router>
        <Routes>
          <Route/>
        </Routes>
      </Router>
    </Fragment>
  );
}

export default App;
```

- c) Se exporta el proyecto al final para poder ser reutilizado en otros componentes
- d) Se ejecuta nuevamente el proyecto en terminal con el comando npm start para visualizar en el navegador la interfaz gráfica del componente creado y así validar la funcionalidad del ruteo definido.

7. CREACION DE COMPONENTES USUARIO, PROVEEDOR Y HOME

Se define la estructura básica de componente indicada en líneas anteriores, y de acuerdo a la estructura de código html del elemento que se desea integrar en cada componente, se agrega después del return de la función flecha teniendo en cuenta que en el lenguaje JSX la etiqueta CLASS del lenguaje HTML y JS deberá definirse con la palabra reservada className (camelCase); para esto la extensión instalada inicialmente, llamada html to JSX, permite convertir automáticamente todo el código integrado, en el lenguaje propio de React, solo con señalar todo el código HTML y dar clic derecho, seguido de convert HTML to JSX; además de esto se debe tener presente que las etiquetas a y elementos href, deben ser sustituidos por el elemento link y to ={} respectivamente, los cuales deben ser importados en la parte superior del componente para su llamado, ya que hacen parte del package instalado react-router-dom; posteriormente se deben reemplazar estas etiquetas (a y href) manualmente en cada línea del código en la que se referencien:

```

paginas > auth > JS Login.js > ...
import React from 'react';
import { Link } from 'react-router-dom';

<div className="social-auth-links text-center mb-3">
  <p>- OR -</p>
  <Link to="#" className="btn btn-block btn-primary">
    <i className="fab fa-facebook mr-2" /> Sign in using Facebook
  </Link>
  <a href="#" className="btn btn-block btn-danger">

```

8. VALIDACION DE PRUEBAS EN CONSOLA Y DOM: en el localhost predeterminado por React, con f12 validamos en consola que se esté ejecutando sin errores la interfaz, en caso de presentarse se deberá verificar el error según la línea definida en pantalla, en los archivos del proyecto; esta acción de revisión tanto del DOM como de la consola se deberá realizar constantemente y alternamente, en la creación de nuevo código para evitar inconvenientes al renderizar el proyecto.
9. FUNCIONALIDAD DE LOS COMPONENTES: para consumir los servicios de los componentes debemos integrar los hooks useState y useEffect, que permiten utilizar el estado inicial de las variables y otras características de React al renderizar la página automáticamente, a través de funciones tipo flecha o función clásica declaradas en cada componente; antes del return (lenguaje JSX) de la función de un componente se debe integrar el código js que dota de funcionalidad a la página con las variables necesarias y los eventos requeridos para su dinamismo.
 - a) El useState

```

import React, {useEffect, useState} from 'react';

function FormUsuarios(props){
  const {onSave} = props;
  const [usuario, setUsuario] = useState({
    _id: null,
    usuario: "",
    clave: "",
    rol: "",
    nombres: "",
    apellidos: "",
    tipDoc: "",
    numDoc: "",
    direccion: "",
    telefono: "",
    email: "",
    estado: ""
  });

```

- b) El `useEffect` se crea con la siguiente sintaxis básica, y nos permite ejecutar todo el código que se integre dentro de su función flecha al renderizar la página por vez primera, dándole un foco al usuario a la hora de navegar en nuestro aplicativo.

```
useEffect(() =>{
  }, [])
```

```
useEffect(() => {
  if(dataUsuario)
    setUsuario(dataUsuario)
  else
    setUsuario(
      {
        _id: "null",
        usuario: "",
        clave: "",
        rol: "",
        nombres: "",
        apellidos: "",
        tipDoc: "",
        numDoc: "",
        direccion: "",
        telefono: "",
        email: "",
        estado: ""
      }
    )
  }, [dataUsuario])
const handleChange = (e) => { // funcion para cada vez que cambie Los
  valores del usuario, al final agregar la propiedad nombrey valor
  setUsuario({
    ...usuario,
    [e.target.name]:e.target.value
  });
}
```

- c) Creación de formularios, para recolección de información del usuario dentro de las cajas y botones creadas; se utilizan los eventos `onSubmit`, `onSave`, `dataUsuario`, limpiar dentro de las etiquetas del formulario, en las

cuales se desea captar o modificar información; estos eventos deberán llamar dentro de su valor la función tipo flecha definida antes del return con la siguiente sintaxis,

```
const onSubmit= (e)=> { //se crea funcion para guardar informacion del
formulario
    e.preventDefault(); //metodo que permite controlar el
comportamiento por defecto del formulario de redericcionar a un
servidor y evitar errores
    onSave(usuario);
    limpiar();
};

return (
    <form onSubmit={onSubmit}>
        <div>
            Usuario:
            <input type="text" name="usuario"
value={usuario.usuario} onChange={(e) => {handleChange(e)}}/>
        </div>
    </form>
);
```

10. MAQUETACIÓN DE INTERFAZ GRÁFICA

- Se crean los mockups de los componentes que integran el DOM del proyecto de acuerdo a los requerimientos funcionales y no funcionales definidos en el Sprint 1.
- Se implementa desde la librería de Bootstrap los bloques de códigos necesarios para dotar de funcionalidad gráfica al proyecto, relacionando el link de esta tecnología en la página principal del index.html, que contiene la carpeta public, como recurso en línea, sin necesidad de descargar un paquete adicional.
- De acuerdo a las plantillas definidas en la página de Bootstrap para los diversos elementos que van a componer nuestro proyecto, copiamos y pegamos el bloque de código predefinido dentro de la función de cada archivo js despues del return del componente a desarrollar.

11. PRUEBAS DE LOS ENPOINTS DE LA API: se realizan las pruebas del CRUD en el aplicativo Postman para valida que los métodos GET,

EVIDENCIAS DE USO DE JIRA:

Capturas de pantalla del uso de Jira y de historias de usuario

Jira Software | Tu trabajo | **Proyectos** | Filtros | Paneles | Personas | Aplicaciones | **Crear** | Q. Buscar

Proyectos / U29_MiDulceOnline_10

Backlog

Tablero Sprint 3 # Añadir fechas (15 incidencias) 0 0 0 **Iniciar sprint**

U2910-11	Como administrador del sistema quiero ver los productos, sus precios y sus cantidades en cualquier...	EN CURSO	J	
U2910-14	Como administrador del sistema quiero visualizar la hora y fecha de cada transacción en el inventario.	EN CURSO	LC	
U2910-35	Instalación de Dependencias en entorno de trabajo	DESARROLLO FRONTED Y BACKEND	EN CURSO	HS
U2910-36	Creacion de proyecto react local	DESARROLLO FRONTED Y BACKEND	EN CURSO	OC
U2910-37	Creacion de estructura del proyecto	DESARROLLO FRONTED Y BACKEND	EN CURSO	LC
U2910-40	Creacion de componente app	DESARROLLO FRONTED Y BACKEND	EN CURSO	SE
U2910-41	Creacion de clase vista	DESARROLLO FRONTED Y BACKEND	EN CURSO	J
U2910-42	Creacion de estructura de componentes	DESARROLLO FRONTED Y BACKEND	EN CURSO	HS
U2910-43	Creacion de componentes usuario, proveedor y home	DESARROLLO FRONTED Y BACKEND	EN CURSO	SE

Jira Software | Tu trabajo | **Proyectos** | Filtros | Paneles | Personas | Aplicaciones | **Crear** | Q. Buscar

Proyectos / U29_MiDulceOnline_10

Backlog

U2910-41 Creacion de clase vista DESARROLLO FRONTED Y BACKEND EN CURSO J

U2910-42 Creacion de estructura de componentes DESARROLLO FRONTED Y BACKEND EN CURSO HS

U2910-43 Creacion de componentes usuario, proveedor y home DESARROLLO FRONTED Y BACKEND EN CURSO SE

U2910-44 Validacion de pruebas en consola y DOM DESARROLLO FRONTED Y BACKEND EN CURSO J

U2910-45 Funcionalidad de los componentes DESARROLLO FRONTED Y BACKEND EN CURSO OC

U2910-101 Pruebas de los endpoints de la API DESARROLLO FRONTED Y BACKEND EN CURSO LC

U2910-100 Maquetación de interfaz gráfica DESARROLLO FRONTED Y BACKEND EN CURSO LC

U2910-34 Pruebas de Funcionamiento PLAN DE PRUEBAS, DESPLIEGUE Y ... EN CURSO SE

U2910-99 Evidencia Dailys EN CURSO LC

+ Crear incidencia