APLICACIONES WEB 1

ESTUDIANTE: SANCHEZ FERNANDEZ EULISES SANTIAGO

NIVEL:5TO

FECHA:16/07/2025

DOCENTE: ING. PATRICIA QUIROZ

TEMA: PROYECTO SEFUNDO PARCIAL

2024-1

Uleam
Facultad de Ciencias de la Vida y Tecnologías

Carrera: Ingeniería en Tecnología de la Información – 5 nivel

Descripción del Negocio o Tema Elegido:

Contexto o Dominio:

El proyecto de gestión de becas de la Universidad Laica Eloy Alfaro de Manabí (ULEAM) tiene como objetivo principal optimizar y facilitar el proceso de aplicación y administración de becas para estudiantes. La plataforma está diseñada para brindar una experiencia eficiente y accesible tanto para los estudiantes que desean postularse a becas como para el personal administrativo encargado de gestionar estas solicitudes.

El sistema permite a los estudiantes completar y enviar sus aplicaciones en línea, adjuntar documentos necesarios, y seguir el estado de sus solicitudes en tiempo real. Por otro lado, el personal administrativo puede revisar, aprobar o rechazar las solicitudes, y llevar un registro detallado de todas las aplicaciones de manera centralizada.

Características Clave:

1. **Interfaz de Usuario Amigable:** La plataforma cuenta con una interfaz intuitiva y fácil de usar, diseñada para que los estudiantes puedan navegar y completar sus solicitudes sin complicaciones.
2. **Gestión de Solicitudes:** Los administradores pueden gestionar eficientemente todas las solicitudes de becas, con herramientas para revisar, aprobar, rechazar y comunicarse con los postulantes.
3. **Notificaciones y Seguimiento:** Los estudiantes reciben notificaciones sobre el estado de sus solicitudes y pueden hacer seguimiento en tiempo real.
4. **Seguridad y Privacidad:** El sistema asegura que toda la información y documentos proporcionados por los estudiantes sean manejados de manera segura y confidencial.
5. **Soporte y Ayuda:** La plataforma incluye secciones de ayuda y soporte para asistir a los usuarios en cualquier etapa del proceso de aplicación.

Beneficios:

- **Para los Estudiantes:** Facilita el acceso a oportunidades de becas, ahorra tiempo y reduce la complejidad del proceso de aplicación.
- **Para la Administración:** Mejora la eficiencia en la gestión de becas, reduce el papeleo y permite un seguimiento detallado de cada solicitud.

Este proyecto no solo mejora la experiencia de los usuarios, sino que también contribuye al objetivo de la ULEAM de apoyar a los estudiantes talentosos y necesitados, asegurando que tengan acceso a los recursos necesarios para continuar su educación.

Propósito del Sistema Web:

El propósito principal del sistema web de gestión de becas de la Universidad Laica Eloy Alfaro de Manabí (ULEAM) es simplificar y automatizar el proceso de aplicación, revisión y administración de becas para estudiantes y personal administrativo. Este sistema tiene como objetivos específicos:

1. **Facilitar el Proceso de Aplicación:**
    - Proveer una plataforma accesible y fácil de usar donde los estudiantes puedan completar y enviar sus solicitudes de becas en línea, eliminando la necesidad de procesos manuales y papeleo.
2. **Optimizar la Gestión Administrativa:**
    - Permitir a los administradores de becas revisar, aprobar o rechazar solicitudes de manera eficiente, con herramientas que agilicen la gestión y seguimiento de cada aplicación.
3. **Mejorar la Comunicación:**

- Facilitar la comunicación entre los estudiantes y el personal administrativo mediante notificaciones automáticas sobre el estado de las solicitudes y canales de soporte para resolver dudas y problemas.

4. **Asegurar la Transparencia y el Seguimiento:**
    - Ofrecer a los estudiantes la posibilidad de hacer seguimiento en tiempo real del estado de sus solicitudes, asegurando transparencia en el proceso y generando confianza en el sistema.
5. **Garantizar la Seguridad y Privacidad:**
    - Proteger la información y documentos de los estudiantes con altos estándares de seguridad y privacidad, cumpliendo con las normativas y regulaciones aplicables.
6. **Promover la Equidad y Oportunidad:**
    - Asegurar que todos los estudiantes tengan igual acceso a las oportunidades de becas, apoyando a aquellos con talento y necesidad económica para continuar su educación sin barreras financieras.

En resumen, el sistema web está diseñado para ser una herramienta integral que facilita tanto la postulación a becas por parte de los estudiantes como la gestión eficiente y segura de estas solicitudes por parte del personal administrativo, contribuyendo así al desarrollo académico y personal de la comunidad estudiantil de ULEAM.

Requerimientos Funcionales:

1. **Registro y Autenticación:**
    - El sistema debe permitir a los estudiantes registrarse utilizando su correo institucional.
    - El sistema debe permitir a los estudiantes iniciar sesión para acceder a sus aplicaciones de becas.
    - El sistema debe permitir a los administradores iniciar sesión para gestionar las solicitudes de becas.
2. **Gestión de Solicitudes de Becas:**
    - El sistema debe permitir a los estudiantes completar y enviar formularios de solicitud de becas en línea.
    - El sistema debe permitir a los estudiantes adjuntar documentos necesarios (ej. comprobante de ingresos, cartas de recomendación).
    - El sistema debe permitir a los administradores revisar, aprobar o rechazar las solicitudes de becas.
    - El sistema debe permitir a los administradores agregar comentarios y observaciones a las solicitudes.
3. **Notificaciones y Seguimiento:**
    - El sistema debe enviar notificaciones automáticas a los estudiantes sobre el estado de sus solicitudes (recibido, en revisión, aprobado, rechazado).
    - El sistema debe permitir a los estudiantes hacer seguimiento en tiempo real del estado de sus solicitudes.
4. **Gestión de Usuarios:**
    - El sistema debe permitir a los administradores gestionar los perfiles de los usuarios (estudiantes y otros administradores).

- El sistema debe permitir la asignación de roles y permisos a los usuarios.

5. **Interfaz de Usuario:**
    - El sistema debe proporcionar una interfaz de usuario intuitiva y fácil de usar.
    - El sistema debe ser accesible desde dispositivos móviles y de escritorio.

Requerimientos No Funcionales:

1. **Seguridad:**
    - El sistema debe proteger la información personal y documentos de los estudiantes con altos estándares de seguridad.
    - El sistema debe implementar autenticación segura y encriptación de datos sensibles.
2. **Rendimiento:**
    - El sistema debe ser capaz de manejar múltiples solicitudes simultáneamente sin afectar el rendimiento.
    - El sistema debe ofrecer tiempos de respuesta rápidos para todas las operaciones de usuario.
3. **Escalabilidad:**
    - El sistema debe ser escalable para soportar un aumento en el número de usuarios y solicitudes de becas en el futuro.
4. **Disponibilidad:**
    - El sistema debe estar disponible y accesible para los usuarios al menos el 99.9% del tiempo.
5. **Mantenibilidad:**

- El código del sistema debe ser modular y bien documentado para facilitar su mantenimiento y actualización.
- El sistema debe permitir la fácil integración de nuevas funcionalidades y mejoras.

6. **Usabilidad:**
   - El sistema debe ser fácil de aprender y usar para todos los tipos de usuarios.
   - El diseño de la interfaz debe seguir principios de usabilidad y accesibilidad.
7. **Compatibilidad:**
   - El sistema debe ser compatible con los navegadores web más comunes (Chrome, Firefox, Safari, Edge).
   - El sistema debe funcionar en diferentes sistemas operativos (Windows, macOS, Linux).

Herramientas y Tecnologías de Programación Utilizadas:

1. **Framework y Librerías Frontend:**
   - **React:** Para construir la interfaz de usuario y manejar la vista de los componentes.
   - **React Router:** Para la navegación entre los diferentes componentes de la aplicación (como `Bienvenida`, `Postular`, `Consultabeca`, y `GestionAdmin`).
   - **Axios:** Para realizar solicitudes HTTP al backend.
2. **Herramientas de Desarrollo:**

- **Visual Studio Code (VS Code):** Utilizado como el entorno de desarrollo integrado (IDE) para escribir y depurar el código del proyecto.
- **npm (Node Package Manager):** Para la gestión de dependencias y paquetes necesarios para el proyecto, como React y otras librerías.

3. **Estilización:**
   - **CSS:** Para la estilización de los componentes de la interfaz de usuario.
   - **Bootstrap:** Posiblemente para el diseño responsivo y componentes estilizados (si fue mencionado en el proyecto).

Almacenamiento de Datos:

1. **Almacenamiento:**

   Almacenamiento del Token: Cuando el usuario inicia sesión, el token se guarda en localStorage.

   Uso del Token: En el componente Consultabeca, el token se recupera de localStorage y se usa para autenticar la solicitud al backend.

   Persistencia de Datos: Los datos del usuario y el estado de las solicitudes se pueden almacenar en localStorage para su uso en toda la aplicación.

2. **Autenticación y Seguridad (si implementado):**
   - **JWT (JSON Web Tokens):** Para la autenticación segura de usuarios.
   - **bcrypt:** Para el hash de contraseñas y asegurar la autenticación de los usuarios.

Código fuente completo.

Github: https://github.com/EulisesSanchez17/Proyecro2P.git

Interfaz1…. App.js

```
import React, { useState, useEffect } from 'react';
import { BrowserRouter as Router, Route, Routes, useNavigate } from
'react-router-dom';
import Bienvenida from '../Interfa2/Bienvenida';
import Postular from '../Interfaz3/Postular';
import Consultabeca from '../Interfaz4/Consultabeca'; // Importa el
componente Consultabeca
import Formulario from '../Interfaz5/Formulario'; // Importa el
componente Formulario
import Admin from '../InterfazAdmin/admin'; // Importa el componente
Admin
import GestionAdmin from '../InterfazAdmin/GestionAdmin'; // Importa el
componente GestionAdmin

const styles = {
  container: {
    fontFamily: 'Arial, sans-serif',
    backgroundColor: '#f4f4f4',
    display: 'flex',
    justifyContent: 'center',
    alignItems: 'center',
    height: '100vh',
    backgroundImage: `url(${process.env.PUBLIC_URL}/LogoUleam.png)`,
    backgroundSize: 'cover',
    backgroundRepeat: 'no-repeat',
    backgroundPosition: 'center'
  },
  formContainer: {
    backgroundColor: 'transparent',
    padding: '20px',
    borderRadius: '0px',
    border: '3px solid black',
    boxShadow: '0 0 10px rgb(1, 1, 1)',
    maxWidth: '400px',
    width: '100%',
    color: 'black',
    marginTop: '20px'
  },
  title: {
    textAlign: 'center',
    marginBottom: '20px',
    color: 'black'
  },
```

```
  input: {
    width: '100%',
    padding: '10px',
    marginTop: '5px',
    border: '1px solid #ccc',
    borderRadius: '5px'
  },
  select: {
    width: '100%',
    padding: '10px',
    marginTop: '5px',
    border: '1px solid #ccc',
    borderRadius: '5px'
  },
  button: {
    width: '100%',
    padding: '10px',
    backgroundColor: '#241ba4',
    border: 'none',
    borderRadius: '5px',
    color: 'white',
    fontSize: '16px',
    marginTop: '20px',
    cursor: 'pointer'
  },
  message: {
    marginTop: '10px',
    textAlign: 'center',
    color: 'red'
  }
};

function App() {
  const [email, setEmail] = useState('');
  const [password, setPassword] = useState('');
  const [userType, setUserType] = useState('');
  const [message, setMessage] = useState('');
  const [isAuthenticated, setIsAuthenticated] = useState(false);
  const [userRole, setUserRole] = useState('');
  const navigate = useNavigate();

  useEffect(() => {
    if (isAuthenticated) {
      if (userRole === 'admin') {
        navigate('/admin');
      } else {
        navigate('/bienvenida');
      }
    }
```

```
  }, [isAuthenticated, userRole, navigate]);

  const handleSubmit = (event) => {
    event.preventDefault();
    const credentials = [
      { userType: 'student', email: 'e1312961434@uleam.live.edu.ec',
password: 'mary2000xD' },
      { userType: 'admin', email: 'administrador@gmail.com', password:
'mary2000xd' }
    ];

    const storedCredentials = credentials.find(cred => cred.userType ===
userType);
    if (!storedCredentials) {
      setMessage('Por favor, seleccione un tipo de usuario.');
      return;
    }

    if (email === storedCredentials.email && password ===
storedCredentials.password) {
      setIsAuthenticated(true);
      setUserRole(userType);
    } else {
      setMessage('Credenciales incorrectas. Por favor, intente de
nuevo.');
    }
  };

  return (
    <div style={styles.container}>
      <div style={styles.formContainer}>
        <h2 style={styles.title}>Becas Uleam</h2>
        <h1 style={styles.title}>Acceda con su correo institucional</h1>
        <form id="loginForm" onSubmit={handleSubmit}>
          <label htmlFor="email" style={{ ...styles.title
}}>Usuario:</label>
          <input type="email" id="email" name="email"
placeholder="usuario@uleam.edu.ec" required value={email} onChange={e =>
setEmail(e.target.value)} style={styles.input} />

          <label htmlFor="password" style={{ ...styles.title
}}>Contraseña:</label>
          <input type="password" id="password" name="password"
minLength="8" required value={password} onChange={e =>
setPassword(e.target.value)} style={styles.input} />

          <label htmlFor="userType" style={{ ...styles.title }}>Tipo de
usuario:</label>
```

```
          <select id="userType" name="userType" required value={userType}
onChange={e => setUserType(e.target.value)} style={styles.select}>
            <option value="">Seleccione una opción</option>
            <option value="student">Estudiante</option>
            <option value="admin">Administrador</option>
          </select>

          <button type="submit" style={styles.button}>Ingresar</button>
        </form>
        <div id="message" style={styles.message}>{message}</div>
      </div>
    </div>
  );
}

function AppWrapper() {
  return (
    <Router>
      <Routes>
        <Route path="/" element={<App />} />
        <Route path="/bienvenida" element={<Bienvenida />} />
        <Route path="/postular" element={<Postular />} />
        <Route path="/consultabeca" element={<Consultabeca />} />
        <Route path="/formulario" element={<Formulario />} />
        <Route path="/admin" element={<Admin />} />
        <Route path="/gestionadmin" element={<GestionAdmin />} /> {/*
Nueva ruta hacia GestionAdmin */}
      </Routes>
    </Router>
  );
}

export default AppWrapper;
```

Interfaz2…Bienvenida.js

```
import React from 'react';
import { useNavigate } from 'react-router-dom';

const styles = {
  body: {
    fontFamily: 'Segoe UI, Tahoma, Geneva, Verdana, sans-serif',
    backgroundColor: '#f0f2f5',
    margin: 0,
    display: 'flex',
```

```
    flexDirection: 'column',
    alignItems: 'center',
    height: '100vh',
    backgroundImage: `url(${process.env.PUBLIC_URL}/Fondo.png)`,
    backgroundSize: 'cover',
    backgroundRepeat: 'no-repeat',
  },
  header: {
    width: '100%',
    backgroundColor: '#313132',
    color: 'white',
    display: 'flex',
    justifyContent: 'flex-start',
    padding: '10px 20px',
    boxShadow: '0 2px 4px rgba(0, 0, 0, 0.1)',
  },
  headerLeft: {
    display: 'flex',
    gap: '10px',
  },
  button: {
    backgroundColor: '#b7bfc2',
    border: 'none',
    color: 'rgb(0, 0, 0)',
    padding: '10px 20px',
    fontSize: '14px',
    cursor: 'pointer',
    borderRadius: '4px',
    transition: 'background 0.3s ease',
  },
  buttonHover: {
    backgroundColor: '#21a8d5',
  },
  main: {
    flex: 1,
    display: 'flex',
    flexDirection: 'column',
    justifyContent: 'center',
    alignItems: 'center',
    textAlign: 'center',
    padding: '20px',
  },
  h1: {
    fontSize: '24px',
    color: '#333',
  },
  p: {
    fontSize: '18px',
    color: '#666',
```

```
  },
};

function Bienvenida() {
  const navigate = useNavigate();

  const handlePostularClick = () => {
    navigate('/postular'); // Navega a la ruta '/postular' al hacer clic
en el botón Postular
  };

  const handleConsultaBecasClick = () => {
    navigate('/consultabeca'); // Navega a la ruta '/consultabeca' al
hacer clic en el botón Consulta de Becas
  };

  const handleSalirClick = () => {
    navigate('/'); // Navega a la ruta '/' al hacer clic en el botón
Salir
  };

  return (
    <div style={styles.body}>
      <header style={styles.header}>
        <div style={styles.headerLeft}>
          <button style={styles.button}
onClick={handlePostularClick}>Postular</button>
          <button style={styles.button}
onClick={handleConsultaBecasClick}>Consulta de Becas</button>
          <button style={styles.button}
onClick={handleSalirClick}>Salir</button>
        </div>
      </header>

      <main style={styles.main}>
        <h1 style={styles.h1}>Bienvenido a la Plataforma de Solicitud de
Becas</h1>
        <p style={styles.p}>Utilice el botón "Postular" para comenzar su
solicitud de beca.</p>
      </main>
    </div>
  );
}

export default Bienvenida;
```

Interfaz3…..Postular.js

```
import React, { useState } from 'react';
import { useNavigate } from 'react-router-dom';

const styles = {
  body: {
    fontFamily: 'Arial, sans-serif',
    margin: 0,
    padding: 0,
    backgroundColor: '#f0f0f0',
    backgroundImage: `url('LogoUleam.png')`,
    backgroundSize: 'cover',
    backgroundRepeat: 'no-repeat',
    height: '100vh',
    display: 'flex',
    flexDirection: 'column',
    alignItems: 'center',
    justifyContent: 'center'
  },
  header: {
    display: 'flex',
    justifyContent: 'space-between',
    alignItems: 'center',
    backgroundColor: '#313132',
    padding: '10px 20px',
    color: 'white',
    width: '100%'
  },
  headerLeft: {
    display: 'flex',
    gap: '10px'
  },
  button: {
    backgroundColor: '#b7bfc2',
    border: 'none',
    color: '#000',
    padding: '10px 20px',
    fontSize: '14px',
    cursor: 'pointer',
    borderRadius: '4px',
    transition: 'background 0.3s ease'
  },
  buttonHover: {
```

```
    backgroundColor: '#21a8d5'
  },
  main: {
    width: '100%',
    maxWidth: '800px',
    margin: '40px auto',
    padding: '20px',
    backgroundColor: 'transparent',
    borderRadius: '10px',
    boxShadow: '0 0 10px rgba(0, 0, 0, 0.1)'
  },
  section: {
    marginBottom: '20px'
  },
  title: {
    color: '#333',
    marginBottom: '10px'
  },
  formGroup: {
    marginBottom: '15px'
  },
  label: {
    display: 'block',
    marginBottom: '5px',
    color: '#555'
  },
  input: {
    width: '100%',
    padding: '10px',
    border: '1px solid #ccc',
    borderRadius: '5px'
  },
  submitButton: {
    backgroundColor: '#3498db',
    color: 'white',
    padding: '12px 20px',
    border: 'none',
    borderRadius: '5px',
    cursor: 'pointer',
    transition: 'background-color 0.3s'
  },
  submitButtonHover: {
    backgroundColor: '#2980b9'
  }
};

const Postular = () => {
  const navigate = useNavigate();
  const [solicitud, setSolicitud] = useState({
```

```
    tipoBeca: '',
    periodoAcademico: '',
    fechaElaboracion: ''
  });

  const handleVolver = () => {
    navigate('/bienvenida');
  };

  const handleSubmit = (event) => {
    event.preventDefault();
    // Verificar si ya existe una solicitud de beca
    const storedSolicitud = localStorage.getItem('solicitudBeca');
    if (storedSolicitud) {
      alert('Ya existe una solicitud de beca pendiente. Si desea
cambiarla, elimine la solicitud actual.');
    } else {
      // Guardar datos en localStorage
      localStorage.setItem('solicitudBeca', JSON.stringify(solicitud));
      alert('Beca enviada correctamente.');
      // Limpiar formulario después del envío
      setSolicitud({
        tipoBeca: '',
        periodoAcademico: '',
        fechaElaboracion: ''
      });
    }
  };

  const handleChange = (event) => {
    const { name, value } = event.target;
    setSolicitud(prevState => ({
      ...prevState,
      [name]: value
    }));
  };

  return (
    <div style={styles.body}>
      <header style={styles.header}>
        <div style={styles.headerLeft}>
          <button style={styles.button}
onClick={handleVolver}>Volver</button>
        </div>
      </header>

      <main style={styles.main}>
        <section style={styles.section}>
          <h2 style={styles.title}>Formulario de Solicitud de Beca</h2>
```

```
          <form id="scholarship-form" onSubmit={handleSubmit}>
            <div style={styles.formGroup}>
              <label htmlFor="tipo_beca" style={styles.label}>Tipo de
Beca o Ayuda Económica:</label>
              <select id="tipo_beca" name="tipoBeca" required
style={styles.input} onChange={handleChange} value={solicitud.tipoBeca}>
                <option value="">Seleccionar...</option>
                <option value="merito">Beca por Mérito</option>
                <option value="necesidad">Beca por Necesidad
Económica</option>
                <option value="deportiva">Beca Deportiva</option>
                <option value="cultural">Beca Cultural</option>
              </select>
            </div>
            <div style={styles.formGroup}>
              <label htmlFor="periodo_academico"
style={styles.label}>Período Académico:</label>
              <select id="periodo_academico" name="periodoAcademico"
required style={styles.input} onChange={handleChange}
value={solicitud.periodoAcademico}>
                <option value="">Seleccionar...</option>
                <option value="2024A">2024 - Primer Semestre</option>
                <option value="2024B">2024 - Segundo Semestre</option>
                <option value="2025A">2025 - Primer Semestre</option>
                <option value="2025B">2025 - Segundo Semestre</option>
              </select>
            </div>
            <div style={styles.formGroup}>
              <label htmlFor="fecha_elaboracion"
style={styles.label}>Fecha de Elaboración:</label>
              <input type="date" id="fecha_elaboracion"
name="fechaElaboracion" required style={styles.input}
onChange={handleChange} value={solicitud.fechaElaboracion} />
            </div>
            <button type="submit" style={styles.submitButton}>Enviar
Solicitud</button>
          </form>
        </section>
      </main>
    </div>
  );
};

export default Postular;
```

Interfaz4….. Consultabeca.js

```
import React, { useState, useEffect } from 'react';
import { Link, useNavigate } from 'react-router-dom';

const styles = {
  body: {
    fontFamily: 'Arial, sans-serif',
    margin: 0,
    padding: 0,
    backgroundColor: 'transparent',
    backgroundImage: `url(${process.env.PUBLIC_URL}/LogoUleam.png)`,
    backgroundSize: 'cover',
    backgroundRepeat: 'no-repeat',
    height: '100vh',
    display: 'flex',
    flexDirection: 'column',
    alignItems: 'center',
    justifyContent: 'center'
  },
  header: {
    width: '100%',
    backgroundColor: '#313132',
    color: 'white',
    display: 'flex',
    justifyContent: 'space-between',
    padding: '10px 20px',
    boxShadow: '0 2px 4px rgba(0, 0, 0, 0.1)',
    position: 'fixed',
    top: 0,
    zIndex: 1000
  },
  headerLeft: {
    display: 'flex',
    alignItems: 'center'
  },
  button: {
    backgroundColor: '#b7bfc2',
    border: 'none',
    color: 'rgb(0, 0, 0)',
    padding: '10px 20px',
    fontSize: '14px',
    cursor: 'pointer',
    borderRadius: '4px',
    transition: 'background 0.3s ease',
```

```
    textDecoration: 'none' // Añadido para evitar subrayado del link
  },
  main: {
    width: '100%',
    maxWidth: '1000px',
    margin: '80px auto 40px auto', // Ajustamos el margen superior para
que el contenido no se oculte detrás de la barra de tarea
    padding: '20px',
    backgroundColor: 'transparent',
    borderRadius: '10px',
    boxShadow: '0 0 10px rgba(0, 0, 0, 0.1)'
  },
  section: {
    marginBottom: '20px'
  },
  title: {
    color: '#333',
    marginBottom: '10px',
    textAlign: 'center'
  },
  consultaTable: {
    marginBottom: '20px'
  },
  table: {
    width: '100%',
    borderCollapse: 'collapse',
    marginBottom: '20px'
  },
  thead: {
    backgroundColor: '#3498db',
    color: 'white'
  },
  th: {
    padding: '10px',
    border: '1px solid #ccc',
    textAlign: 'left'
  },
  td: {
    padding: '10px',
    border: '1px solid #ccc',
    textAlign: 'left'
  },
  fileInput: {
    display: 'none'
  },
  uploadButton: {
    backgroundColor: '#3498db',
    color: 'white',
    padding: '5px 10px',
```

```
    border: 'none',
    borderRadius: '5px',
    cursor: 'pointer',
    transition: 'background-color 0.3s'
  },
  downloadButton: {
    backgroundColor: '#3498db',
    color: 'white',
    padding: '5px 10px',
    border: 'none',
    borderRadius: '5px',
    cursor: 'pointer',
    transition: 'background-color 0.3s'
  },
  eliminarButton: {
    backgroundColor: '#e74c3c',
    color: 'white',
    padding: '5px 10px',
    border: 'none',
    borderRadius: '5px',
    cursor: 'pointer',
    transition: 'background-color 0.3s'
  }
};

const Consultabeca = () => {
  const navigate = useNavigate();
  const [solicitud, setSolicitud] = useState(null);
  const [enviado, setEnviado] = useState(false);

  useEffect(() => {
    // Cargar solicitud de beca guardada en localStorage al cargar la
página
    const storedSolicitud = localStorage.getItem('solicitudBeca');
    if (storedSolicitud) {
      setSolicitud(JSON.parse(storedSolicitud));
    }
  }, []);

  const handleNavigateToForm = () => {
    navigate('/formulario');
  };

  const handleEliminarSolicitud = () => {
    localStorage.removeItem('solicitudBeca');
    setSolicitud(null);
    alert('Solicitud de beca eliminada correctamente.');
  };
```

```
  const handleFileUpload = () => {
    alert('Subiendo documento...');
    // Lógica para subir documento
  };

  const handleFileDownload = () => {
    alert('Descargando documento...');
    // Lógica para descargar documento
  };

  const handleEnviarSolicitud = () => {
    // Enviar solicitud a GestionAdmin
    if (solicitud) {
      // Simulación de envío de datos a GestionAdmin
      localStorage.setItem('gestionAdminData',
JSON.stringify(solicitud));
      setEnviado(true);
    }
  };

  return (
    <div style={styles.body}>
      <header style={styles.header}>
        <div style={styles.headerLeft}>
          <Link to="/bienvenida" style={styles.button}>Volver</Link>
        </div>
      </header>

      <main style={styles.main}>
        <section style={styles.section}>
          <h2 style={styles.title}>Consulta de Becas</h2>
          <div className="consulta-table" style={styles.consultaTable}>
            <h3 style={styles.title}>Estado de la Solicitud</h3>
            {enviado && <p style={{ color: 'green', textAlign: 'center'
}}>Solicitud enviada correctamente a GestionAdmin.</p>}
            {solicitud ? (
              <table id="solicitudes-table" style={styles.table}>
                <thead style={styles.thead}>
                  <tr>
                    <th style={styles.th}>Formulario</th>
                    <th style={styles.th}>ID Solicitud</th>
                    <th style={styles.th}>Tipo de Beca</th>
                    <th style={styles.th}>Período Académico</th>
                    <th style={styles.th}>Estado</th>
                    <th style={styles.th}>Descargar Documentos</th>
                    <th style={styles.th}>Cargar Documento</th>
                    <th style={styles.th}>Eliminar Solicitud</th>
                  </tr>
                </thead>
```

```
                <tbody>
                  <tr>
                    <td>
                      <button onClick={handleNavigateToForm}
className="ingresar-button" style={styles.button}>Ingresar</button>
                    </td>
                    <td>001</td>
                    <td>{solicitud.tipoBeca}</td>
                    <td>{solicitud.periodoAcademico}</td>
                    <td>Pendiente</td>
                    <td>
                      <button className="download-button"
onClick={handleFileDownload}
style={styles.downloadButton}>Descargar</button>
                    </td>
                    <td>
                      <label htmlFor="file-upload" className="upload-
label" style={styles.uploadButton}>Subir</label>
                      <input type="file" id="file-upload"
name="fileUpload" style={styles.fileInput} onChange={handleFileUpload} />
                    </td>
                    <td>
                      <button onClick={handleEliminarSolicitud}
className="eliminar-button"
style={styles.eliminarButton}>Eliminar</button>
                    </td>
                  </tr>
                </tbody>
              </table>
            ) : (
              <p>No hay solicitud de beca enviada.</p>
            )}
            <div style={{ textAlign: 'center', marginTop: '20px' }}>
              <button onClick={handleEnviarSolicitud}
style={styles.button}>Enviar Solicitud a Gestión Admin</button>
            </div>
          </div>
        </section>
      </main>
    </div>
  );
};

export default Consultabeca;
```

Interfaz5…...Formulario.js

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

const Formulario = () => {
  const [formData, setFormData] = useState({
    'nombre-completo': '',
    'edad': '',
    'email': '',
    'telefono': '',
    'universidad': '',
    'carrera': '',
    'semestre': '',
    'motivos': '',
    'referencia1-nombre': '',
    'referencia1-contacto': '',
    'referencia2-nombre': '',
    'referencia2-contacto': '',
    'experiencia-previa': '',
    'expectativas': ''
  });

  const [errors, setErrors] = useState({});

  const handleChange = (e) => {
    const { name, value } = e.target;
    setFormData({
      ...formData,
      [name]: value
    });
    // Limpiar el error al modificar el campo
    if (errors[name]) {
      setErrors({
        ...errors,
        [name]: ''
      });
    }
  };

  const handleSubmit = (e) => {
    e.preventDefault();
    const validationErrors = validate(formData);
    if (Object.keys(validationErrors).length === 0) {
```

```
      // Aquí puedes agregar la lógica para enviar el formulario, como
enviar los datos a un backend
      console.log(formData);
      // También puedes resetear el formulario después del envío si es
necesario
      setFormData({
        'nombre-completo': '',
        'edad': '',
        'email': '',
        'telefono': '',
        'universidad': '',
        'carrera': '',
        'semestre': '',
        'motivos': '',
        'referencia1-nombre': '',
        'referencia1-contacto': '',
        'referencia2-nombre': '',
        'referencia2-contacto': '',
        'experiencia-previa': '',
        'expectativas': ''
      });
    } else {
      setErrors(validationErrors);
    }
  };

  // Función para validar el formulario
  const validate = (formData) => {
    let errors = {};

    if (!formData['nombre-completo']) {
      errors['nombre-completo'] = 'El nombre completo es requerido';
    }

    if (!formData['edad']) {
      errors['edad'] = 'La edad es requerida';
    } else if (isNaN(formData['edad'])) {
      errors['edad'] = 'La edad debe ser un número';
    } else if (parseInt(formData['edad']) <= 0) {
      errors['edad'] = 'La edad debe ser mayor que cero';
    }

    if (!formData['email']) {
      errors['email'] = 'El email es requerido';
    } else if (!/\S+@\S+\.\S+/.test(formData['email'])) {
      errors['email'] = 'El email es inválido';
    }

    if (!formData['telefono']) {
```

```
      errors['telefono'] = 'El teléfono es requerido';
    } else if (isNaN(formData['telefono'])) {
      errors['telefono'] = 'El teléfono debe ser un número';
    }

    if (!formData['universidad']) {
      errors['universidad'] = 'La universidad es requerida';
    }

    if (!formData['carrera']) {
      errors['carrera'] = 'La carrera es requerida';
    }

    if (!formData['semestre']) {
      errors['semestre'] = 'El semestre es requerido';
    }

    if (!formData['motivos']) {
      errors['motivos'] = 'Los motivos son requeridos';
    }

    if (!formData['referencia1-nombre']) {
      errors['referencia1-nombre'] = 'El nombre de la primera referencia
es requerido';
    }

    if (!formData['referencia1-contacto']) {
      errors['referencia1-contacto'] = 'El contacto de la primera
referencia es requerido';
    }

    if (!formData['referencia2-nombre']) {
      errors['referencia2-nombre'] = 'El nombre de la segunda referencia
es requerido';
    }

    if (!formData['referencia2-contacto']) {
      errors['referencia2-contacto'] = 'El contacto de la segunda
referencia es requerido';
    }

    if (!formData['experiencia-previa']) {
      errors['experiencia-previa'] = 'La experiencia previa es
requerida';
    }

    if (!formData['expectativas']) {
      errors['expectativas'] = 'Las expectativas son requeridas';
    }
```

```
    return errors;
  };

  return (
    <div style={{
      fontFamily: 'Arial, sans-serif',
      margin: 0,
      padding: 0,
      backgroundColor: '#ffffff',
      color: '#000000',
      backgroundImage: "url('LogoUleam.png')",
      backgroundSize: 'cover',
      backgroundRepeat: 'no-repeat'
    }}>
      <header style={{
        width: '100%',
        backgroundColor: '#313132',
        color: 'white',
        display: 'flex',
        justifyContent: 'space-between',
        padding: '10px 20px',
        boxShadow: '0 2px 4px rgba(0, 0, 0, 0.1)'
      }}>
        <img src="LogoUleam.png" alt="Logo Uleam" className="logo"
style={{ height: '50px' }} />
        <button onClick={() => window.history.back()} style={{
          backgroundColor: '#b7bfc2',
          border: 'none',
          color: '#000000',
          padding: '10px 20px',
          fontSize: '14px',
          cursor: 'pointer',
          borderRadius: '4px',
          transition: 'background 0.3s ease'
        }}>Volver</button>
      </header>

      <h2 style={{ textAlign: 'center', marginBottom: '20px'
}}>Formulario de Solicitud de Beca</h2>

      <form onSubmit={handleSubmit} style={{
        maxWidth: '800px',
        margin: '40px auto',
        padding: '20px',
        backgroundColor: '#555555',
        borderRadius: '5px',
        boxShadow: '0 0 10px rgba(0, 0, 0, 0.1)'
      }}>
```

```
        <div className="form-section" style={{ display: 'flex', flexWrap:
'wrap' }}>
          <div className="form-group" style={{ flex: '1 1 50%',
marginBottom: '15px', padding: '0 10px' }}>
            <label htmlFor="nombre-completo" style={{ fontWeight: 'bold',
display: 'block' }}>Nombre Completo:</label>
            <input type="text" id="nombre-completo" name="nombre-
completo" value={formData['nombre-completo']} onChange={handleChange}
required style={{
              width: '100%',
              padding: '8px',
              borderRadius: '5px',
              border: 'none',
              backgroundColor: '#777777',
              color: '#ffffff',
              marginTop: '5px'
            }} />
            {errors['nombre-completo'] && <span style={{ color: 'red'
}}>{errors['nombre-completo']}</span>}
          </div>

          <div className="form-group" style={{ flex: '1 1 50%',
marginBottom: '15px', padding: '0 10px' }}>
            <label htmlFor="edad" style={{ fontWeight: 'bold', display:
'block' }}>Edad:</label>
            <input type="number" id="edad" name="edad"
value={formData['edad']} onChange={handleChange} required style={{
              width: '100%',
              padding: '8px',
              borderRadius: '5px',
              border: 'none',
              backgroundColor: '#777777',
              color: '#ffffff',
              marginTop: '5px'
            }} />
            {errors['edad'] && <span style={{ color: 'red'
}}>{errors['edad']}</span>}
          </div>

          <div className="form-group" style={{ flex: '1 1 100%',
marginBottom: '15px', padding: '0 10px' }}>
            <label htmlFor="email" style={{ fontWeight: 'bold', display:
'block' }}>Email:</label>
            <input type="email" id="email" name="email"
value={formData['email']} onChange={handleChange} required style={{
              width: '100%',
              padding: '8px',
              borderRadius: '5px',
              border: 'none',
```

```
              backgroundColor: '#777777',
              color: '#ffffff',
              marginTop: '5px'
            }} />
            {errors['email'] && <span style={{ color: 'red'
}}>{errors['email']}</span>}
          </div>

          <div className="form-group" style={{ flex: '1 1 50%',
marginBottom: '15px', padding: '0 10px' }}>
            <label htmlFor="telefono" style={{ fontWeight: 'bold',
display: 'block' }}>Teléfono:</label>
            <input type="tel" id="telefono" name="telefono"
value={formData['telefono']} onChange={handleChange} required style={{
              width: '100%',
              padding: '8px',
              borderRadius: '5px',
              border: 'none',
              backgroundColor: '#777777',
              color: '#ffffff',
              marginTop: '5px'
            }} />
            {errors['telefono'] && <span style={{ color: 'red'
}}>{errors['telefono']}</span>}
          </div>

          <div className="form-group" style={{ flex: '1 1 50%',
marginBottom: '15px', padding: '0 10px' }}>
            <label htmlFor="universidad" style={{ fontWeight: 'bold',
display: 'block' }}>Universidad:</label>
            <input type="text" id="universidad" name="universidad"
value={formData['universidad']} onChange={handleChange} required style={{
              width: '100%',
              padding: '8px',
              borderRadius: '5px',
              border: 'none',
              backgroundColor: '#777777',
              color: '#ffffff',
              marginTop: '5px'
            }} />
            {errors['universidad'] && <span style={{ color: 'red'
}}>{errors['universidad']}</span>}
          </div>

          <div className="form-group" style={{ flex: '1 1 50%',
marginBottom: '15px', padding: '0 10px' }}>
            <label htmlFor="carrera" style={{ fontWeight: 'bold',
display: 'block' }}>Carrera:</label>
```

```
            <input type="text" id="carrera" name="carrera"
value={formData['carrera']} onChange={handleChange} required style={{
              width: '100%',
              padding: '8px',
              borderRadius: '5px',
              border: 'none',
              backgroundColor: '#777777',
              color: '#ffffff',
              marginTop: '5px'
            }} />
            {errors['carrera'] && <span style={{ color: 'red'
}}>{errors['carrera']}</span>}
          </div>

          <div className="form-group" style={{ flex: '1 1 50%',
marginBottom: '15px', padding: '0 10px' }}>
            <label htmlFor="semestre" style={{ fontWeight: 'bold',
display: 'block' }}>Semestre:</label>
            <input type="text" id="semestre" name="semestre"
value={formData['semestre']} onChange={handleChange} required style={{
              width: '100%',
              padding: '8px',
              borderRadius: '5px',
              border: 'none',
              backgroundColor: '#777777',
              color: '#ffffff',
              marginTop: '5px'
            }} />
            {errors['semestre'] && <span style={{ color: 'red'
}}>{errors['semestre']}</span>}
          </div>

          <div className="form-group" style={{ flex: '1 1 100%',
marginBottom: '15px', padding: '0 10px' }}>
            <label htmlFor="motivos" style={{ fontWeight: 'bold',
display: 'block' }}>Motivos para solicitar la beca:</label>
            <textarea id="motivos" name="motivos"
value={formData['motivos']} onChange={handleChange} required style={{
              width: '100%',
              padding: '8px',
              borderRadius: '5px',
              border: 'none',
              backgroundColor: '#777777',
              color: '#ffffff',
              minHeight: '100px',
              marginTop: '5px'
            }} />
            {errors['motivos'] && <span style={{ color: 'red'
}}>{errors['motivos']}</span>}
```

```
          </div>

          <div className="form-group" style={{ flex: '1 1 50%',
marginBottom: '15px', padding: '0 10px' }}>
            <label htmlFor="referencia1-nombre" style={{ fontWeight:
'bold', display: 'block' }}>Nombre de la primera referencia:</label>
            <input type="text" id="referencia1-nombre" name="referencia1-
nombre" value={formData['referencia1-nombre']} onChange={handleChange}
required style={{
              width: '100%',
              padding: '8px',
              borderRadius: '5px',
              border: 'none',
              backgroundColor: '#777777',
              color: '#ffffff',
              marginTop: '5px'
            }} />
            {errors['referencia1-nombre'] && <span style={{ color: 'red'
}}>{errors['referencia1-nombre']}</span>}
          </div>

          <div className="form-group" style={{ flex: '1 1 50%',
marginBottom: '15px', padding: '0 10px' }}>
            <label htmlFor="referencia1-contacto" style={{ fontWeight:
'bold', display: 'block' }}>Contacto de la primera referencia:</label>
            <input type="text" id="referencia1-contacto"
name="referencia1-contacto" value={formData['referencia1-contacto']}
onChange={handleChange} required style={{
              width: '100%',
              padding: '8px',
              borderRadius: '5px',
              border: 'none',
              backgroundColor: '#777777',
              color: '#ffffff',
              marginTop: '5px'
            }} />
            {errors['referencia1-contacto'] && <span style={{ color:
'red' }}>{errors['referencia1-contacto']}</span>}
          </div>

          <div className="form-group" style={{ flex: '1 1 50%',
marginBottom: '15px', padding: '0 10px' }}>
            <label htmlFor="referencia2-nombre" style={{ fontWeight:
'bold', display: 'block' }}>Nombre de la segunda referencia:</label>
            <input type="text" id="referencia2-nombre" name="referencia2-
nombre" value={formData['referencia2-nombre']} onChange={handleChange}
required style={{
              width: '100%',
              padding: '8px',
```

```
              borderRadius: '5px',
              border: 'none',
              backgroundColor: '#777777',
              color: '#ffffff',
              marginTop: '5px'
            }} />
            {errors['referencia2-nombre'] && <span style={{ color: 'red'
}}>{errors['referencia2-nombre']}</span>}
          </div>

          <div className="form-group" style={{ flex: '1 1 50%',
marginBottom: '15px', padding: '0 10px' }}>
            <label htmlFor="referencia2-contacto" style={{ fontWeight:
'bold', display: 'block' }}>Contacto de la segunda referencia:</label>
            <input type="text" id="referencia2-contacto"
name="referencia2-contacto" value={formData['referencia2-contacto']}
onChange={handleChange} required style={{
              width: '100%',
              padding: '8px',
              borderRadius: '5px',
              border: 'none',
              backgroundColor: '#777777',
              color: '#ffffff',
              marginTop: '5px'
            }} />
            {errors['referencia2-contacto'] && <span style={{ color:
'red' }}>{errors['referencia2-contacto']}</span>}
          </div>

          <div className="form-group" style={{ flex: '1 1 100%',
marginBottom: '15px', padding: '0 10px' }}>
            <label htmlFor="experiencia-previa" style={{ fontWeight:
'bold', display: 'block' }}>Experiencia previa (si aplica):</label>
            <textarea id="experiencia-previa" name="experiencia-previa"
value={formData['experiencia-previa']} onChange={handleChange} style={{
              width: '100%',
              padding: '8px',
              borderRadius: '5px',
              border: 'none',
              backgroundColor: '#777777',
              color: '#ffffff',
              minHeight: '100px',
              marginTop: '5px'
            }} />
            {errors['experiencia-previa'] && <span style={{ color: 'red'
}}>{errors['experiencia-previa']}</span>}
          </div>
```

```
          <div className="form-group" style={{ flex: '1 1 100%',
marginBottom: '15px', padding: '0 10px' }}>
            <label htmlFor="expectativas" style={{ fontWeight: 'bold',
display: 'block' }}>Expectativas al obtener la beca:</label>
            <textarea id="expectativas" name="expectativas"
value={formData['expectativas']} onChange={handleChange} required
style={{
              width: '100%',
              padding: '8px',
              borderRadius: '5px',
              border: 'none',
              backgroundColor: '#777777',
              color: '#ffffff',
              minHeight: '100px',
              marginTop: '5px'
            }} />
            {errors['expectativas'] && <span style={{ color: 'red'
}}>{errors['expectativas']}</span>}
          </div>

          <div className="form-group" style={{ flex: '1 1 100%',
textAlign: 'center' }}>
            <button type="submit" style={{
              backgroundColor: '#b7bfc2',
              border: 'none',
              color: '#000000',
              padding: '10px 20px',
              fontSize: '16px',
              cursor: 'pointer',
              borderRadius: '4px',
              transition: 'background 0.3s ease'
            }}>Enviar Solicitud</button>
          </div>
        </div>
      </form>

      <footer style={{
        width: '100%',
        backgroundColor: '#313132',
        color: 'white',
        textAlign: 'center',
        padding: '10px 0',
        position: 'fixed',
        bottom: '0',
        left: '0'
      }}>
        <p style={{ margin: '0' }}>Universidad Laica Eloy Alfaro de
Manabí</p>
      </footer>
```

```
    </div>
  );
}

export default Formulario;
```

InterfazAdmin…..admin.js

```
import React from 'react';
import { useNavigate } from 'react-router-dom';

const Admin = () => {
  const navigate = useNavigate();

  const handleManageScholarships = () => {
    navigate('/GestionAdmin'); // Redirige a la página gestionS.html
  };

  const handleLogout = () => {
    navigate('/'); // Redirige a la página de inicio de sesión
  };

  const styles = {
    body: {
      fontFamily: 'Arial, sans-serif',
      margin: 0,
      padding: 0,
      backgroundColor: '#f0f0f0',
      backgroundImage: 'url("LogoUleam.png")',
      backgroundSize: 'cover',
      backgroundRepeat: 'no-repeat',
    },
    header: {
      display: 'flex',
      justifyContent: 'space-between',
      alignItems: 'center',
      backgroundColor: '#313132',
      padding: '10px 20px',
      color: 'white',
    },
    headerLeft: {
      display: 'flex',
      alignItems: 'center',
    },
    button: {
      marginLeft: '10px',
```

```
    padding: '10px 20px',
    border: 'none',
    borderRadius: '5px',
    backgroundColor: '#b7bfc2',
    color: 'white',
    cursor: 'pointer',
    transition: 'background-color 0.3s',
  },
  buttonHover: {
    backgroundColor: '#21a8d5',
  },
  main: {
    width: '100%',
    maxWidth: '1000px',
    margin: '40px auto',
    padding: '20px',
    backgroundColor: '#fff',
    borderRadius: '10px',
    boxShadow: '0 0 10px rgba(0, 0, 0, 0.1)',
  },
  hidden: {
    display: 'none',
  },
  h2: {
    color: '#333',
    marginBottom: '20px',
    textAlign: 'center',
  },
  table: {
    width: '100%',
    borderCollapse: 'collapse',
    marginBottom: '20px',
  },
  thead: {
    backgroundColor: '#3498db',
    color: 'white',
  },
  th: {
    padding: '10px',
    border: '1px solid #ccc',
    textAlign: 'left',
  },
  td: {
    padding: '10px',
    border: '1px solid #ccc',
    textAlign: 'left',
  },
  tbodyTrOdd: {
    backgroundColor: '#f9f9f9',
```

```
    },
    welcomeSection: {
      textAlign: 'center',
      margin: '40px 0',
      fontSize: '1.2em',
      color: '#333',
    },
    footer: {
      textAlign: 'left',
      padding: '10px 20px',
      backgroundColor: '#3498db',
      color: 'white',
      position: 'fixed',
      bottom: 0,
      left: 0,
      width: '100%',
    },
  };

  return (
    <div style={styles.body}>
      <header style={styles.header}>
        <div style={styles.headerLeft}>
          <button style={styles.button}
onClick={handleManageScholarships}>Gestionar Becas</button>
          <button style={styles.button}
onClick={handleLogout}>Salir</button>
        </div>
      </header>
      <main style={styles.main}>
        <section id="users-section" style={styles.hidden}>
          <h2 style={styles.h2}>Usuarios Registrados</h2>
          <table style={styles.table}>
            <thead style={styles.thead}>
              <tr>
                <th style={styles.th}>ID</th>
                <th style={styles.th}>Nombre</th>
                <th style={styles.th}>Correo Electrónico</th>
                <th style={styles.th}>Tipo de Usuario</th>
                <th style={styles.th}>Acciones</th>
              </tr>
            </thead>
            <tbody>
              {/* Aquí se insertarán los usuarios registrados */}
            </tbody>
          </table>
        </section>

        <section id="scholarships-section" style={styles.hidden}>
```

```
          <h2 style={styles.h2}>Solicitudes de Becas</h2>
          <table style={styles.table}>
            <thead style={styles.thead}>
              <tr>
                <th style={styles.th}>ID Solicitud</th>
                <th style={styles.th}>Nombre del Estudiante</th>
                <th style={styles.th}>Tipo de Beca</th>
                <th style={styles.th}>Período Académico</th>
                <th style={styles.th}>Estado</th>
                <th style={styles.th}>Acciones</th>
              </tr>
            </thead>
            <tbody>
              {/* Aquí se insertarán las solicitudes de becas */}
            </tbody>
          </table>
        </section>

        <section id="welcome-section" style={styles.welcomeSection}>
          <p>Bienvenido, Administrador. Aquí puede gestionar usuarios y
becas.</p>
        </section>
      </main>
      <footer style={styles.footer}>
        <p>Universidad Laica Eloy Alfaro de Manabí &copy; 2024</p>
      </footer>
    </div>
  );
};

export default Admin;
```

GestionAdmin.js

```
import React, { useState, useEffect } from 'react';
import { useNavigate } from 'react-router-dom';

const GestionAdmin = () => {
  const navigate = useNavigate();
  const [solicitudes, setSolicitudes] = useState([]);

  useEffect(() => {
    // Aquí deberías tener alguna lógica para cargar las solicitudes
desde algún servicio o almacenamiento
    // Ejemplo estático:
    const solicitudesEjemplo = [
      {
        id: 1,
        nombre: 'Nombre del Estudiante 1',
```

```
      correo: 'estudiante1@correo.com',
      tipoBeca: 'Beca de Mérito',
      estado: 'Pendiente',
      archivo: 'ruta/al/archivo_del_estudiante1.pdf',
    },
    {
      id: 2,
      nombre: 'Nombre del Estudiante 2',
      correo: 'estudiante2@correo.com',
      tipoBeca: 'Beca Deportiva',
      estado: 'Pendiente',
      archivo: 'ruta/al/archivo_del_estudiante2.pdf',
    },
  ];
  setSolicitudes(solicitudesEjemplo);
}, []);

const handleVolver = () => {
  navigate('/admin');
};

const styles = {
  body: {
    fontFamily: 'Arial, sans-serif',
    margin: 0,
    padding: 0,
    backgroundColor: '#f4f4f4',
    backgroundImage: `url(${process.env.PUBLIC_URL}/LogoUleam.png)`,
    backgroundSize: 'cover',
    backgroundRepeat: 'no-repeat',
  },
  header: {
    width: '100%',
    backgroundColor: '#313132',
    color: 'white',
    display: 'flex',
    justifyContent: 'flex-start',
    padding: '10px 20px',
    boxShadow: '0 2px 4px rgba(0, 0, 0, 0.1)',
  },
  buttons: {
    display: 'flex',
    alignItems: 'center',
  },
  button: {
    backgroundColor: '#b7bfc2',
    border: 'none',
    color: 'rgb(0, 0, 0)',
    padding: '10px 20px',
```

```
    fontSize: '14px',
    cursor: 'pointer',
    borderRadius: '4px',
    transition: 'background 0.3s ease',
    marginLeft: '10px',
  },
  buttonHover: {
    backgroundColor: '#21a8d5',
  },
  container: {
    maxWidth: '800px',
    margin: '40px auto',
    padding: '20px',
    backgroundColor: '#fff',
    borderRadius: '5px',
    boxShadow: '0 0 10px rgba(0, 0, 0, 0.1)',
  },
  h2: {
    textAlign: 'center',
    marginBottom: '20px',
    color: '#333',
  },
  solicitudesList: {
    marginTop: '20px',
  },
  solicitud: {
    border: '1px solid #ccc',
    borderRadius: '5px',
    padding: '15px',
    marginBottom: '15px',
    backgroundColor: '#f9f9f9',
  },
  solicitudInfo: {
    marginBottom: '10px',
  },
  solicitudH3: {
    marginTop: 0,
    color: '#007bff',
  },
  solicitudP: {
    margin: '5px 0',
  },
  acciones: {
    textAlign: 'right',
  },
  btnAprobar: {
    padding: '8px 16px',
    marginLeft: '10px',
    border: 'none',
```

```
      borderRadius: '3px',
      cursor: 'pointer',
      backgroundColor: '#28a745',
      color: '#fff',
    },
    btnRechazar: {
      padding: '8px 16px',
      marginLeft: '10px',
      border: 'none',
      borderRadius: '3px',
      cursor: 'pointer',
      backgroundColor: '#dc3545',
      color: '#fff',
    },
    btnHover: {
      opacity: 0.9,
    },
    descarga: {
      marginTop: '10px',
    },
    btnDescargar: {
      textDecoration: 'none',
      padding: '8px 16px',
      border: '1px solid #007bff',
      borderRadius: '3px',
      backgroundColor: '#007bff',
      color: '#fff',
      cursor: 'pointer',
    },
  };

  return (
    <div style={styles.body}>
      <header style={styles.header}>
        <div style={styles.buttons}>
          <button
            style={styles.button}
            onMouseOver={(e) => (e.target.style.backgroundColor =
styles.buttonHover.backgroundColor)}
            onMouseOut={(e) => (e.target.style.backgroundColor =
styles.button.backgroundColor)}
            onClick={handleVolver}
          >
            Volver
          </button>
        </div>
      </header>

      <div style={styles.container}>
```

```
        <h2 style={styles.h2}>Gestión de Solicitudes de Becas</h2>
        <div style={styles.solicitudesList}>
          {solicitudes.map((solicitud) => (
            <div key={solicitud.id} style={styles.solicitud}>
              <div style={styles.solicitudInfo}>
                <h3 style={styles.solicitudH3}>{solicitud.nombre}</h3>
                <p style={styles.solicitudP}>Correo Electrónico:
{solicitud.correo}</p>
                <p style={styles.solicitudP}>Tipo de Beca:
{solicitud.tipoBeca}</p>
                <p style={styles.solicitudP}>Estado:
{solicitud.estado}</p>
              </div>
              <div style={styles.acciones}>
                <button
                  style={styles.btnAprobar}
                  onMouseOver={(e) => (e.target.style.opacity =
styles.btnHover.opacity)}
                  onMouseOut={(e) => (e.target.style.opacity = 1)}
                >
                  Aprobar
                </button>
                <button
                  style={styles.btnRechazar}
                  onMouseOver={(e) => (e.target.style.opacity =
styles.btnHover.opacity)}
                  onMouseOut={(e) => (e.target.style.opacity = 1)}
                >
                  Rechazar
                </button>
              </div>
              <div style={styles.descarga}>
                <a href={solicitud.archivo} style={styles.btnDescargar}
download>
                  Descargar Archivo
                </a>
              </div>
            </div>
          ))}
        </div>
      </div>
    </div>
  );
};

export default GestionAdmin;
```

index.js

```
import React from 'react';
import ReactDOM from 'react-dom/client';
import App from './Interfaz1/App';


const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
  <React.StrictMode>
    <App />
  </React.StrictMode>
);
```

Manual de Usuario del Sistema Web de Gestión de Becas de ULEAM

Introducción

- Bienvenida y objetivos del manual.
- Descripción general del sistema y su propósito.

Acceso al Sistema

- **Registro de Usuarios**
    - Pasos para crear una cuenta de usuario.
    - Requisitos de registro (ej. uso de correo institucional).
- **Inicio de Sesión**
    - Cómo iniciar sesión en la plataforma.

- Manejo de credenciales seguras.

Página Principal

- **Interfaz de Usuario**
  - Descripción de los elementos de la página principal.
  - Navegación y accesibilidad.

Funcionalidades para Estudiantes

- **Postulación a Becas**
  - Cómo completar y enviar una solicitud de beca.
  - Adjuntar documentos requeridos (ej. comprobantes, cartas de recomendación).
- **Consultar Estado de Solicitudes**
  - Cómo verificar el estado de las solicitudes de becas enviadas.
  - Interpretación de los diferentes estados (pendiente, revisión, aprobado, rechazado).

Funcionalidades para Administradores

- **Gestión de Solicitudes**
  - Revisión y aprobación de solicitudes de becas.
  - Rechazo de solicitudes y comentarios adicionales.
- **Gestión de Usuarios**
  - Asignación de roles y permisos a otros administradores.
  - Actualización de datos de usuario y gestión de cuentas.

Seguridad y Privacidad

- **Políticas de Seguridad**
    - Manejo seguro de contraseñas y tokens de sesión.
    - Protección de datos personales y documentos adjuntos.

Ayuda y Soporte

- **Preguntas Frecuentes (FAQ)**
    - Respuestas a preguntas comunes sobre el uso del sistema.
- **Contacto de Soporte**
    - Información para contactar al equipo de soporte técnico en caso de problemas o dudas.

Mantenimiento y Actualización

- **Notificaciones de Actualización**
    - Procedimientos para informar a los usuarios sobre nuevas versiones o funcionalidades.
- **Mantenimiento Programado**
    - Políticas y procedimientos para mantenimientos programados del sistema.

Apéndices

- **Glosario de Términos**
    - Definiciones de términos técnicos y específicos utilizados en el manual.

Cómo Usar Este Manual

- Guía breve sobre cómo navegar y buscar información en el manual.

- Índice y estructura del contenido para referencia rápida.

Conclusión del Proyecto de Gestión de Becas de ULEAM

El desarrollo e implementación del sistema web de gestión de becas para la Universidad Laica Eloy Alfaro de Manabí (ULEAM) representa un paso significativo hacia la modernización y eficiencia en el proceso de solicitud, revisión y administración de becas. A lo largo de este proyecto, hemos logrado varios hitos importantes:

- **Automatización y Simplificación:** Hemos simplificado el proceso de aplicación a becas mediante una plataforma intuitiva y accesible, permitiendo a los estudiantes enviar sus solicitudes desde cualquier lugar y en cualquier momento.
- **Mejora en la Transparencia y Comunicación:** La implementación de notificaciones automáticas ha mejorado la comunicación entre los estudiantes y la administración, manteniéndolos informados sobre el estado de sus solicitudes en tiempo real.
- **Eficiencia Administrativa:** Para los administradores, el sistema ha optimizado la gestión de solicitudes, proporcionando herramientas para revisar y tomar decisiones informadas de manera más eficiente.
- **Seguridad y Protección de Datos:** Hemos priorizado la seguridad de los datos personales y documentos adjuntos de los estudiantes, implementando medidas robustas de seguridad y privacidad.

Este proyecto no solo ha facilitado el acceso a oportunidades educativas para los estudiantes de ULEAM, sino que también ha sentado las bases para futuras mejoras y expansiones del sistema. Continuaremos trabajando para mantener y mejorar el sistema,

asegurando que cumpla con las necesidades cambiantes de la comunidad estudiantil y administrativa de la universidad.

En resumen, el sistema web de gestión de becas de ULEAM no solo representa una herramienta tecnológica avanzada, sino también un compromiso con la equidad, la eficiencia y la excelencia en la administración de recursos educativos.

Bibliografía:

Documentación de React: https://reactjs.org/docs/getting-started.html

Documentación de MongoDB: https://docs.mongodb.com/

Documentación de Node.js: https://nodejs.org/en/docs/