



## **TÍTULO PROYECTO**

Aplicativo Web de Sistema de Gestión de Inventarios para Cafetería – Bar Texas

Gustavo Adolfo Güiza Walteros  
C.C. 1.007.306.880  
2251159

## **DIRECTOR**

Harvey Nicolas Echavarría Ortiz



Tunja  
Noviembre de 2023

## TABLA DE CONTENIDO

<b>1. FICHA TÉCNICA DEL PROYECTO .....</b>	<b>4</b>
<b>2. PLANTEAMIENTO DEL PROBLEMA.....</b>	<b>5</b>
<b>3. JUSTIFICACIÓN .....</b>	<b>7</b>
<b>4. OBJETIVOS .....</b>	<b>8</b>
<b>5. MODELO DE DESARROLLO.....</b>	<b>9</b>
<b>6. DESARROLLO DEL PROYECTO .....</b>	<b>11</b>
6.1. FASE 1. DEFINICIÓN DE REQUISITOS .....	11
6.2. FASE 2. SELECCIÓN DE TECNOLOGÍA.....	20
6.3. FASE 3. DISEÑO DEL SISTEMA .....	22
6.3.1. MOCKUPS.....	22
➤ LANDING PAGE .....	22
➤ ACERCA DE.....	22
➤ EQUIPO DE DESARROLLO.....	23
➤ LOGIN.....	24
➤ DASHBOARD .....	25
➤ PRODUCTOS .....	25
➤ USUARIOS .....	26
<b>6.4. FASE 4. DESARROLLO DE LA APLICACIÓN.....</b>	<b>26</b>
6.4.1. CREACIÓN DE PROYECTO .....	26
6.4.2. MAPEO DE LAS TABLAS DE BASE DE DATOS. ....	27
6.4.3. CREACIÓN DE ARCHIVOS DAO .....	28
6.4.4. CREACIÓN DE SERVICIOS .....	30
6.4.5. IMPLEMENTACIÓN DE SERVICIOS.....	32
6.4.6. MAPEO DEL FRONT-END (HTML) CON CONTROLADORES .....	34
6.4.7. CREACIÓN DE INTERFACES .....	35
<b>6.5. FASE 5. PRUEBAS .....</b>	<b>44</b>
<b>7. CONCLUSIONES Y ANÁLISIS DE RESULTADOS .....</b>	<b>73</b>

<b>8. REFERENCIAS .....</b>	<b>74</b>
<b>9. ANEXOS.....</b>	<b>76</b>
<b>9.1. MANUAL DE USUARIO.....</b>	<b>76</b>
9.1.1. NAVEGACIÓN PÁGINA PRINCIPAL .....	76
9.1.2. INGRESO Y REGISTRO.....	78
9.1.3. APLICATIVO SISTEMA DE INVENTARIO. ....	81

## 1. FICHA TÉCNICA DEL PROYECTO

Título	Aplicativo Web de Sistema de Gestión de Inventarios para Cafetería – Bar Texas
Nombre Estudiante	Gustavo Adolfo Güiza Walteros
Documento estudiante	1.007.306.880
Correo electrónico estudiante	gustavo.guiza@usantoto.edu.co
Director	Harvey Nicolás Echavarría
Lugar de ejecución del proyecto	Universidad Santo Tomás Seccional - Tunja
Duración	5 meses
Palabras claves	Gestión de Inventarios, Procesamiento de datos, Tecnologías de la información, Gestión Contable.
	
Firma del autor Gustavo Adolfo Güiza Walteros	
	
Firma del director Harvey Nicolas Echavarría	

## 2. PLANTEAMIENTO DEL PROBLEMA

La información es un medio de divulgación que ha estado presente desde el inicio de los tiempos, es el ser humano quien ha adaptado los elementos para su conocimiento, administración y divulgación (Consuelo, s.f.). En un principio la comunicación oral era la forma utilizada para gestionar los datos recolectados, las historias pasadas de generación en generación se convertían en las bitácoras con las que se guardaban aquellos sucesos importantes para la sociedad (Ramón, 2020).

Con la revolución de la tecnología y el nacimiento de nuevas formas de hacer que los datos recolectados perduraran en el tiempo, se empezó a ser uso de diferentes elementos tales como la tinta sobre las pieles de los animales para luego utilizar el papel, el envío y recepción de cartas, la creación de la imprenta, y más adelante la implementación de las TICS para el desarrollo de los medios digitales (Manuel, s.f.), permitiendo que el proceso de gestión y administración de la información se hiciera cada vez más eficiente.

Las empresas al no ser ajenas a estos cambios se vieron obligadas a renovar los procesos tradicionales con los que gestionaban la información, puesto que las tareas de guardado o actualización de los informes contables se hacía a través de medios físicos que ocasionaban deterioro con el tiempo y podían tener complicaciones al revisarlos, ya que en ocasiones no eran legibles y era muy grande la pérdida de información y por ende deterioro económico.

Con la implementación de las TIC, la información fue digitalizada y con el tiempo nuevas herramientas nacieron para hacer el consumo de ésta de forma más rápida y eficiente ahorrando tiempo y dinero a las empresas. Con la creación de software contable y software para la gestión de información se proporcionó la calidad y facilidad de generar informes de gestión de inventarios que permiten procesos de autenticación de usuarios y un CRUD (*Create-Read-Update-Delete*) sobre la información que en él reside.

Los libros y planillas de un inventario presentan debilidades tales como la degradación a través del tiempo por factores tales como la humedad, las bacterias y daños ambientales;

atendiendo a lo anteriormente expuesto la “Cafetería - Bar Texas” encuentra la necesidad de hacer uso de un aplicativo web de sistema de gestión de inventario para mejorar la calidad, la eficiencia, y la facilidad en la gestión de información, para lograr el paso de hacer de procesos manuales a procesos digitales.

A continuación, se visualiza el diagrama causa y efecto, con el fin de explicar de manera gráfica el problema planteado:

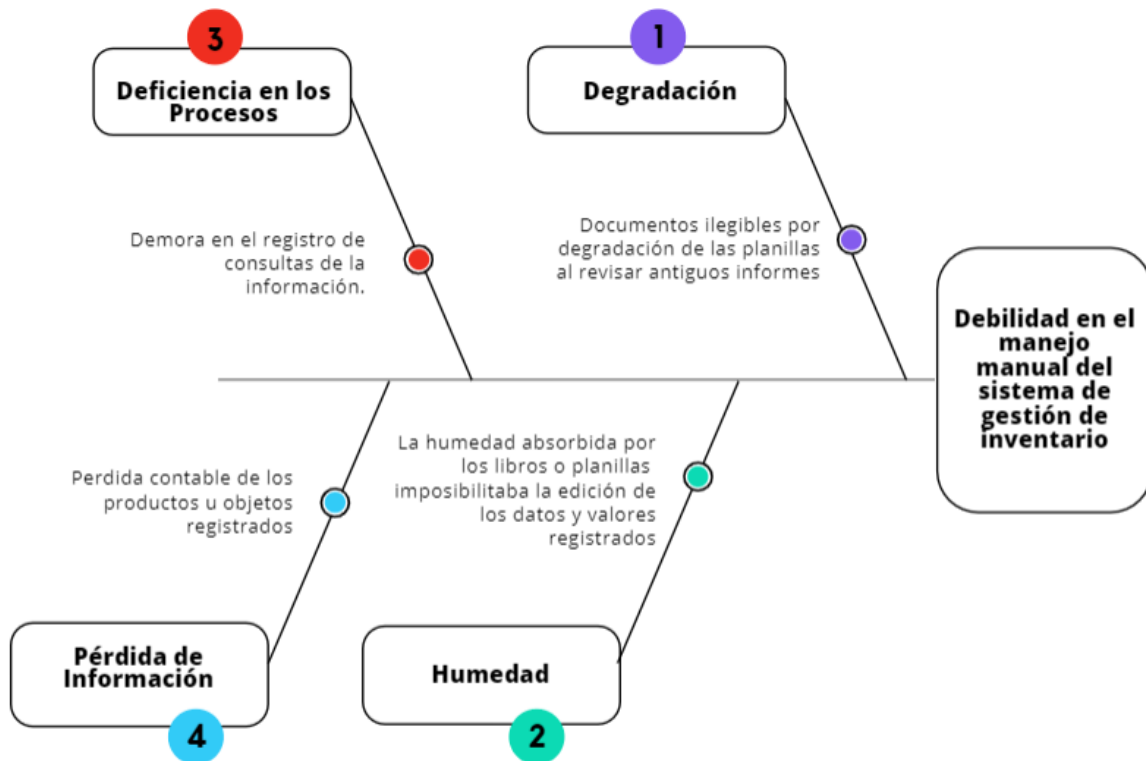


Figura 1: Diagrama Causa y Efecto  
Fuente: Autor

### **3. JUSTIFICACIÓN**

Desarrollar el aplicativo Web de Sistema de Gestión de Inventarios para Cafetería – Bar Texas, es relevante para brindar a la empresa herramientas tecnológicas que digitalicen el proceso de gestión y administración de la información, permitiendo así el paso de uso de estrategias tradicionales a la implementación de soluciones informáticas, con el fin de optimizar las tareas relacionadas con la creación, lectura, actualización y eliminación de los datos que se encuentran plasmados en los formatos físicos.

Es por eso que la implementación del software de gestión busca proporcionar una solución tecnológica a la empresa Cafetería – Bar Texas que necesita de un software contable o de un sistema de inventario que contenga entre sus funcionalidades la autenticación de usuarios y un CRUD sobre la información, puesto que en la era digital las empresas necesitan adaptarse a los cambios que la tecnología propone para que factores como el tiempo, la calidad y la eficiencia sean los pilares de los procesos que aunque se utilizan desde tiempos antiguos necesitan una renovación que esté acorde con los avances tecnológicos.

Este aplicativo mejora la calidad y eficiencia de los procesos pues reemplaza los trabajos manuales a través del sistema de digitalización para un almacenamiento más seguro sin errores de contabilidad o mal conteo de objetos, pues los trabajos manuales dependen de errores humanos, que si se comenten con o sin intención entorpecen y retrasan la labor empresarial de todos los trabajadores provocando pérdidas de tiempo, seguridad y económicas.

En conclusión, la tesis propuesta responde a la necesidad creciente de la era tecnológica para la digitalización de los procesos empresariales, puesto que este proyecto provee un sistema informático que es capaz de ofrecer robustez, seguridad y fiabilidad al momento de gestionar información de vital importancia para el excelente funcionamiento de la empresa Cafetería – Bar Texas.

## 4. OBJETIVOS

### 4.1. Objetivo General

Desarrollar un aplicativo web para la gestión de inventarios de la empresa Cafetería – Bar Texas, teniendo como paradigma la programación orientada a objetos.

### 4.2. Objetivos específicos

Tabla 1. Objetivos específicos

Nro.	Objetivos Específicos
1	Construir la especificación de requisitos del software propuesto por medio del Lenguaje Unificado de Modelado - UML y utilizando la metodología de desarrollo ágil Extreme Programming XP, para definir las funcionalidades y características que este tendrá.
2	Implementar el sistema de gestión de inventario utilizando el framework Spring, arquitectura MVC, lenguaje HTML y la biblioteca Bootstrap, con el fin de hacer la entrega del desarrollo propuesto.
3	Ejecutar pruebas funcionales, integrales y rendimiento al sistema con el fin de validar su funcionamiento, calidad, eficiencia y el cumplimiento de los estándares utilizando el software SonarQube.

Fuente: Autor

## 5. MODELO DE DESARROLLO

“La Programación Extrema o Extreme Programming, es un enfoque de la ingeniería de software formulado por Kent Beck, se considera el más destacado de los procesos ágiles de desarrollo de software. Al igual que estos, la programación extrema se diferencia de los métodos tradicionales principalmente en que presenta más énfasis en la adaptabilidad que en la previsibilidad” (SINTYA MILENA MELÉNDEZ VALLADAREZ, 2016).

Esta metodología se basa en ciclos cortos que se enfocan en desarrollar características del software, teniendo presente que la prueba y error son factores claves para validar si se está cumpliendo con el objetivo de la funcionalidad a entregar al cliente. Por tal motivo, tiene como bases la satisfacción del cliente, potencialización del trabajo en grupo y la minimización de riesgos en cuanto a costo, tiempo, calidad y alcance del proyecto.

Como características de esta metodología se tiene:

- Metodología basada en prueba y error para obtener un software que funcione realmente.
- Fundamentada en principios.
- Está orientada hacia quien produce y usa software (el cliente participa muy activamente).
- Reduce el coste del cambio en todas las etapas del ciclo de vida del sistema.
- Combina las que han demostrado ser las mejores prácticas para desarrollar software, y las lleva al extremo.
- Cliente bien definido. (SINTYA MILENA MELÉNDEZ VALLADAREZ, 2016)
- Los requisitos pueden cambiar.
- Grupo pequeño y muy integrado (2-12 personas).
- Equipo con formación elevada y capacidad de aprender (SINTYA MILENA MELÉNDEZ VALLADAREZ, 2016)

Este proyecto al abarcar la metodología Extreme Programming XP usó cuatro fases que le permitieron en conjunto definir las tareas que se realizaron para la entrega del software de gestión, teniendo en cuenta que cada una de estas fases permitió hacer seguimiento a los

objetivos y cada una de las actividades propuestas, facilitaron el desarrollo de los módulos del proyecto.

Estas fases fueron:

### **Fase 1. Ingeniería de Requisitos**

Esta especificación se basó en las directrices dadas por el estándar IEEE Práctica Recomendada para Especificaciones de Requisitos Software ANSI/IEEE 830, 1998. Así mismo, se planteó el diagrama de casos de uso y el diagrama Entidad – Relación que define cada una de las tablas y sus cardinalidades para la base de datos.

También se tuvo en cuenta la metodología ágil Extreme Programming y el desarrollo de la documentación referente a las historias de usuario y tareas de ingeniería.

### **Fase 2. Propuesta de Diseño**

Se diseñó la base de datos mediante un Diagrama entidad relación (EDR), incluyendo los datos que se tuvieron en cuenta en cada requisito que se registró. Además, se construyeron los mockups respectivos del aplicativo para ser validados por el cliente.

### **Fase 3. Desarrollo**

En esta fase se estructuraron y ejecutaron las tareas relacionadas con la codificación del BackEnd y FrontEnd del proyecto, para el sistema de gestión de inventarios de la empresa Cafetería – Bar Texas, utilizando MVC como patrón de arquitectura de software y como paradigma, la programación orientada a objetos. Así mismo se hizo uso del framework Spring, lenguaje HTML y la biblioteca Bootstrap.

### **Fase 4. Pruebas**

En esta fase se ejecutaron las pruebas al sistema, incluyendo pruebas funcionales, integrales y de rendimiento, con el fin de validar su correcto funcionamiento y eficiencia, para dar cumplimiento a los estándares de calidad mediante el uso del software SonarQube.

## 6. DESARROLLO DEL PROYECTO

### 6.1. Fase 1. Definición de requisitos

En esta sección se dan a conocer las historias de usuario, los requisitos funcionales y no funcionales, diagrama de casos de uso y diagrama entidad – relación:

#### ➤ Historias de Usuario

#### Roles

1. Administrador
2. Empleado

#### Puntos Estimados (Puntos de Historia)

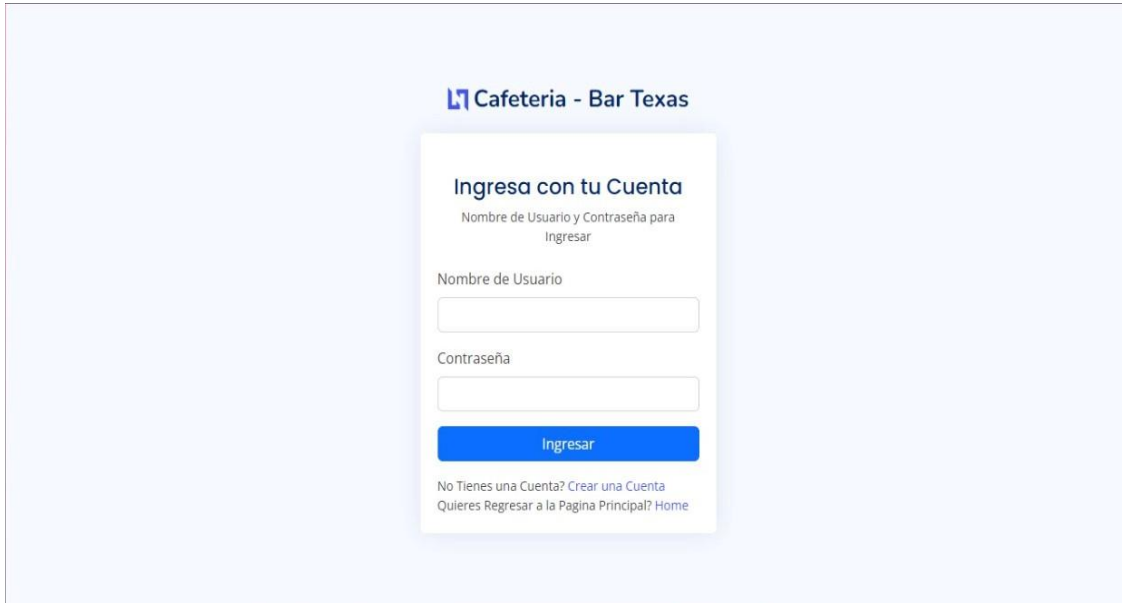
- 1 → Sencillo (Toma un par de horas)
- 2 → Regular (Un día)
- 3 → Complejo (Varios días)
- 5 → Muy Complejo (Casi una semana)
- 8 → Casi Imposible (Varias semanas)

#### 1. Login

- El usuario que ingreso al sistema debe tener la capacidad y los permisos para administrar el inventario y los recursos humanos de la empresa, para ello debe digitar en el aplicativo su nombre de usuario y contraseña.

<b>Historia de Usuario</b>	
<b>Número:</b> UH-001	<b>Usuario:</b> Persona que ingresa al sistema
<b>Nombre Historia:</b> Yo como usuario quiero acceder al sistema por medio de mi nombre de usuario y contraseña para administrar el inventario y los recursos humanos de la empresa.	
<b>Prioridad en Negocio:</b> Alto	<b>Riesgo en Desarrollo:</b> Medio
<b>Puntos Estimados:</b> 3	
<b>Programador Responsable:</b> Gustavo	
<b>Descripción:</b> Proceso de login por medio del nombre de usuario y la contraseña.	
<b>Observaciones:</b> La contraseña será el número de cédula.	

**Mockup:**



*Figura 2: Mockup Login*

*Fuente: Autor*

*Tabla 1: Historia de Usuario 1*

*Fuente: Autor*

**2. Registro**

- El usuario debe registrar en el formulario de la página el número de cedula, el nombre y la contraseña, esta última quedara cifrada en la base de datos.

<b>Historia de Usuario</b>	
<b>Número:</b> UH-002	<b>Usuario:</b> Persona que se registra al sistema
<b>Nombre Historia:</b> Yo como usuario quiero registrarme en el sistema por medio de un formulario que solicite el número de cedula, el nombre y la contraseña para acceder al aplicativo.	
<b>Prioridad en Negocio:</b> Alto	<b>Riesgo en Desarrollo:</b> Medio
<b>Puntos Estimados:</b> 3	
<b>Programador Responsable:</b> Gustavo	
<b>Descripción:</b> Proceso de registro por medio de la cedula, nombre de usuario y la contraseña.	
<b>Observaciones:</b> El nombre de Usuario será el primer nombre y apellido de la persona que se va a registrar, la contraseña estará cifrada en base de datos y será el número de cédula.	

**Mockup:**

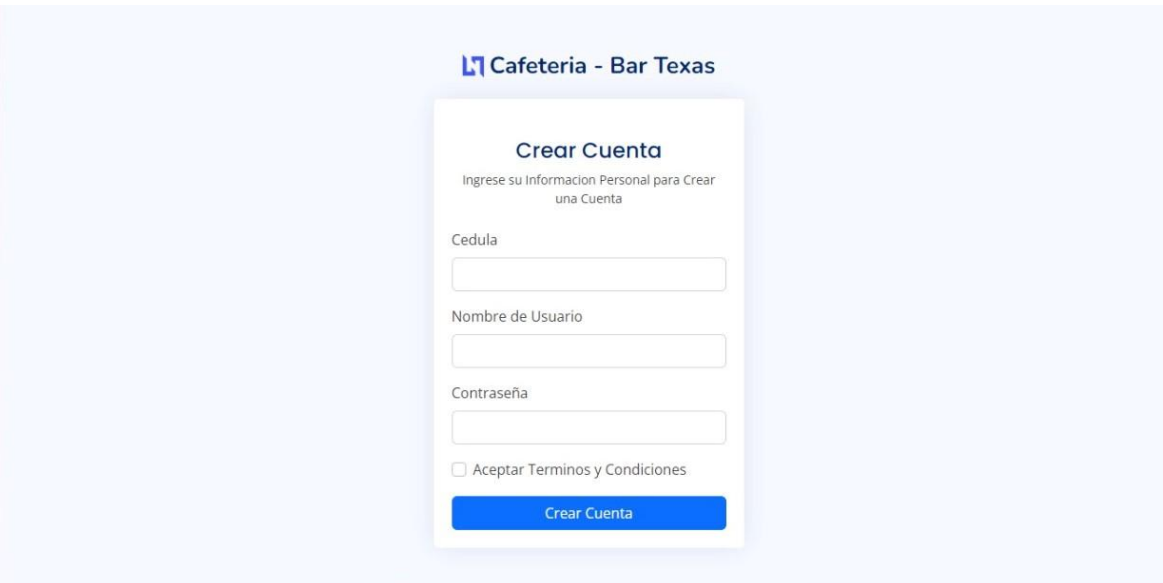


Figura 3: Mockup Registro  
 Fuente: Autor

**Cifrado:**

	id_usuario [PK] bigint	cedula character varying (255)	contrasena character varying (255)	estado boolean	nombre_usuario character varying (255)
1	1	1007306880	\$2a\$10\$eqfj7MeIGwFf...	true	Gustavo Adolfo Güiza ...
2	2	24041452	\$2a\$10\$eqfj7MeIGwFf...	true	Emilcen Walteros Caro
3	3	24856896	\$2a\$10\$eqfj7MeIGwFf...	true	Maria Fatima Murillo
4	4	24041452	\$2a\$10\$eqfj7MeIGwFf...	true	Rubiela Walteros Caro

Figura 4: Cifrado de Contraseña en Base de Datos  
 Fuente: Autor

Tabla 2: Historia de Usuario 2  
 Fuente: Autor

**3. Rol Administrador**

- Se configuró por defecto un usuario con el rol administrador en bases de datos, para que, los usuarios creados por medio del formulario del sitio web ingresen con rol de empleado y así estén limitadas las acciones que puedan ejecutar.

<b>Historia de Usuario</b>	
<b>Número:</b> UH-003	<b>Usuario:</b> jefe
<b>Nombre Historia:</b> Yo como jefe quiero buscar en la pestaña de productos los distribuidores por nombres.	
<b>Prioridad en Negocio:</b> Alto	<b>Riesgo en Desarrollo:</b> Medio
<b>Puntos Estimados:</b> 3	
<b>Programador Responsable:</b> Gustavo	
<b>Descripción:</b> Proceso de filtrado por distribuidor dentro de la pestaña de productos.	
<b>Observaciones:</b> El jefe podrá filtrar o buscar por nombre del distribuidor en la pestaña de productos para una fácil navegación.	

Tabla 3: Historia de Usuario 3

Fuente: Autor

- El administrador o jefe tendrán los permisos para la administración del inventario o de los productos registrados dentro de la entidad.

Historia de Usuario	
<b>Número:</b> UH-004	<b>Usuario:</b> jefe
<b>Nombre Historia:</b> Yo como jefe quiero administrar los productos que están en el local por medio del software.	
<b>Prioridad en Negocio:</b> Alto	<b>Riesgo en Desarrollo:</b> Medio
<b>Puntos Estimados:</b> 3	
<b>Programador Responsable:</b> Gustavo	
<b>Descripción:</b> Proceso de administración de productos dentro del local por medio del software.	
<b>Observaciones:</b> El jefe podrá crear, modificar y eliminar los productos que están y estarán pronto a registrar en el software pertenecientes al local.	
<b>Mockup:</b>	
	

Figura 5: Mockup Productos

Fuente: Autor

Tabla 4: Historia de Usuario 4

Fuente: Autor

- El jefe podrá administrar los datos ingresados en el aplicativo para que en una vista a manera de tabla se muestre el nombre, la cedula y el estado de los usuarios.

Historia de Usuario	
<b>Número:</b> UH-005	<b>Usuario:</b> jefe
<b>Nombre Historia:</b> Yo como jefe quiero administrar la información de mis trabajadores.	
<b>Prioridad en Negocio:</b> Alto	<b>Riesgo en Desarrollo:</b> Medio
<b>Puntos Estimados:</b> 3	
<b>Programador Responsable:</b> Gustavo	
<b>Descripción:</b> Proceso de administrar la información de los trabajadores actuales.	
<b>Observaciones:</b> El jefe podrá administrar la información de los trabajadores actuales en pantalla donde se visualizará nombre, cedula y el estado del empleado.	

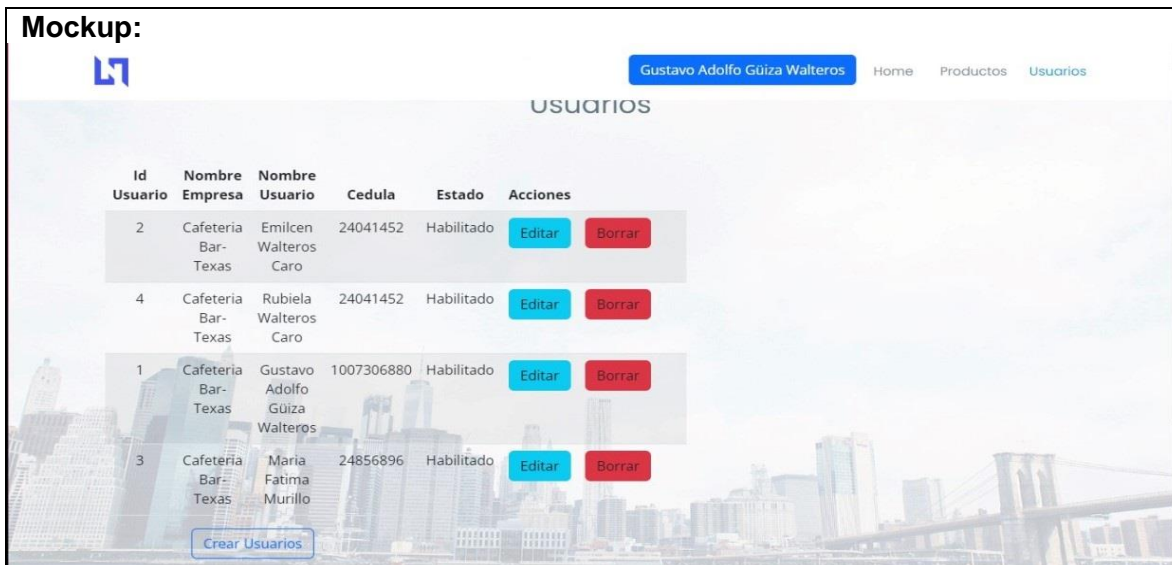


Figura 6: Mockup Recursos Humanos

Fuente: Autor

Tabla 5: Historia de Usuario 5

Fuente: Autor

- Los empleados solo visualizarán la información de los productos registrados en el aplicativo, y no tendrán acceso a las acciones que sólo tiene el administrador.

Historia de Usuario	
<b>Número:</b> UH-006	<b>Usuario:</b> jefe
<b>Nombre Historia:</b> Yo como jefe quiero que mis empleados sólo visualicen la información de los productos existentes en el local.	
<b>Prioridad en Negocio:</b> Alto	<b>Riesgo en Desarrollo:</b> Medio
<b>Puntos Estimados:</b> 3	
<b>Programador Responsable:</b> Gustavo	
<b>Descripción:</b> Proceso de visualizar la información de los productos a los trabajadores actuales.	
<b>Observaciones:</b> los trabajadores podrán visualizar la información de los productos existentes.	

Tabla 6: Historia de Usuario 6

Fuente: Autor

#### 4. Rol Empleado

- Los empleados sólo visualizaran la información de los productos dentro del aplicativo, sin tener presente las acciones que puede hacer el administrador.

Historia de Usuario	
<b>Número:</b> UH-007	<b>Usuario:</b> Empleado
<b>Nombre Historia:</b> Yo como empleado quiero visualizar la información de los productos existentes en el local.	
<b>Prioridad en Negocio:</b> Alto	<b>Riesgo en Desarrollo:</b> Medio
<b>Puntos Estimados:</b> 3	
<b>Programador Responsable:</b> Gustavo	
<b>Descripción:</b> Proceso de visualizar la información de los productos actuales.	

**Observaciones:** Se podrá visualizar la información de los productos existentes.

**Mockup:**

Id Distribuidor	Distribuidor	Productos	Cantidad Unitaria	Precio Unitario	Empaquetado	Cantidad Al Mayor	Precio Al Mayor
1	Bavaria	Poker	64	3000	Canasta	2	92000
12	Bavaria 23	Poker 23	23	23000	Caja	23	2330

Tabla 7: Historia de Usuario 7

Fuente: Autor

### TAREAS DE INGENIERÍAS (TASK CARD)

Tarea de Ingeniería	
<b>Número de Tarea:</b> IT-001	<b>Número de Historia:</b> UH-001 y UH-002.
<b>Nombre de Tarea:</b> Maquetar la vista de usuario de login y registro con autenticación de credenciales.	
<b>Tipo de Tarea:</b> Front-End y Back-End	
<b>Programador Responsable:</b> Gustavo	
<b>Descripción:</b> Construir la interfaz de usuario con Python, Spring, HTML, Hibernate, CSS y Thymeleaf.	

Tabla 8: Tarea de Ingeniería 1

Fuente: Autor

Tarea de Ingeniería	
<b>Número de Tarea:</b> IT-002	<b>Número de Historia:</b> UH-003
<b>Nombre de Tarea:</b> Maquetar la vista de recursos humanos.	
<b>Tipo de Tarea:</b> Front-End y Back-End	
<b>Programador Responsable:</b> Gustavo	
<b>Descripción:</b> Construir la interfaz de usuario con Python, Spring, HTML, Hibernate, CSS y Thymeleaf.	

Tabla 9: Tarea de Ingeniería 2

Fuente: Autor

Tarea de Ingeniería	
<b>Número de Tarea:</b> IT-003	<b>Número de Historia:</b> UH-004

<b>Nombre de Tarea:</b> Maquetar la vista de usuario de productos para jefe.
<b>Tipo de Tarea:</b> Front-End y Back-End
<b>Programador Responsable:</b> Gustavo
<b>Descripción:</b> Construir la interfaz de usuario con Python, Spring, HTML, Hibernate, CSS y Thymeleaf.

Tabla 10: Tarea de Ingeniería 3

Fuente: Autor

Tarea de Ingeniería	
<b>Número de Tarea:</b> IT-004	<b>Número de Historia:</b> UH-006
<b>Nombre de Tarea:</b> Maquetar la vista de usuario de recursos humanos.	
<b>Tipo de Tarea:</b> Front-End	
<b>Programador Responsable:</b> Gustavo	
<b>Descripción:</b> Construir la interfaz de usuario con Python, Spring, HTML, Hibernate, CSS y Thymeleaf.	

Tabla 11: Tarea de Ingeniería 4

Fuente: Autor

Tarea de Ingeniería	
<b>Número de Tarea:</b> IT-005	<b>Número de Historia:</b> UH-008
<b>Nombre de Tarea:</b> Maquetar la vista de usuario de productos para empleados.	
<b>Tipo de Tarea:</b> Front-End y Back-End	
<b>Programador Responsable:</b> Gustavo	
<b>Descripción:</b> Construir la interfaz de usuario con Python, Spring, HTML, CSS y Thymeleaf.	

Tabla 12: Tarea de Ingeniería 5

Fuente: Autor

- Requerimientos Funcionales y No Funcionales

- ✓ Requerimientos Funcionales

Número de requisito	RF-01
Nombre de requisito	Registro
Descripción del requisito	El usuario podrá registrarse con el nombre de usuario (que vendría siendo el nombre completo o el primer nombre y apellido), la cédula y su respectiva contraseña.
Características	Habrará un formulario donde el usuario debe diligenciar los datos que se solicita.
Prioridad del Requisito	Alta

Tabla 13: Requerimiento Funcional 1

Fuente: Autor

Número de requisito	RNF-02
Nombre de requisito	Administración de recursos humanos
Descripción del requisito	El usuario con el rol o cargo administrador puede modificar y eliminar los usuarios previamente registrados en el aplicativo.
Características	Se mostrará una pestaña de recursos humanos donde se podrán modificar, eliminar y cambiar el estado.
Prioridad del Requisito	Alta

Tabla 14: Requerimiento Funcional 2  
 Fuente: Autor

Número de requisito	RF-03
Nombre de requisito	Administración de un producto
Descripción del requisito	El usuario con el rol o cargo respectivo puede modificar y eliminar productos del inventario.
Características	Se mostrará una pestaña de productos donde se podrá modificar, eliminar y cambiar el estado.
Prioridad del Requisito	Alta

Tabla 15: Requerimiento Funcional 3  
 Fuente: Autor

Número de requisito	RF-04
Nombre de requisito	Administración de un distribuidor
Descripción del requisito	El usuario con el rol o cargo respectivo puede modificar y eliminar distribuidores del inventario.
Características	Se mostrará una pestaña de distribuidores donde se podrá modificar, eliminar y cambiar el estado.
Prioridad del Requisito	Alta

Tabla 16: Requerimiento Funcional 4  
 Fuente: Autor

✓ Requerimientos NO Funcionales

Número de requisito	RNF-01
Nombre de requisito	Tipo de usuario
Descripción del requisito	El usuario al momento de acceder al sistema de su respectiva tendrá un respectivo rol dependiendo de éste podrá visualizar, modificar, agregar o eliminar los datos del inventario.
Características	Cada usuario tendrá su respectivo rol que se le asigne y dependiendo de éste podrá hacer varias actividades dentro del sistema.
Prioridad del Requisito	Alta

Tabla 17: Requerimiento No Funcional 1  
 Fuente: Autor

Número de requisito	RNF-02
Nombre de requisito	Inicio de sesión
Descripción del requisito	El usuario podrá acceder al sistema con el nombre de usuario (siendo el nombre completo o el primer nombre y apellido), y su respectiva contraseña (siendo el número de cedula).
Características	Habrà un formulario donde el usuario debe diligenciar los datos que se solicita.
Prioridad del Requisito	Alta

Tabla 18: Requerimiento No Funcional 2  
 Fuente: Autor

➤ Diagramas Casos de Usos

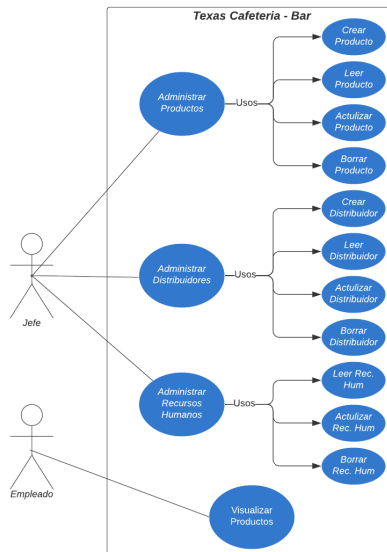


Figura 7: Diagrama Casos de Uso  
 Fuente: Autor

➤ Modelo Entidad – Relación (EDR)

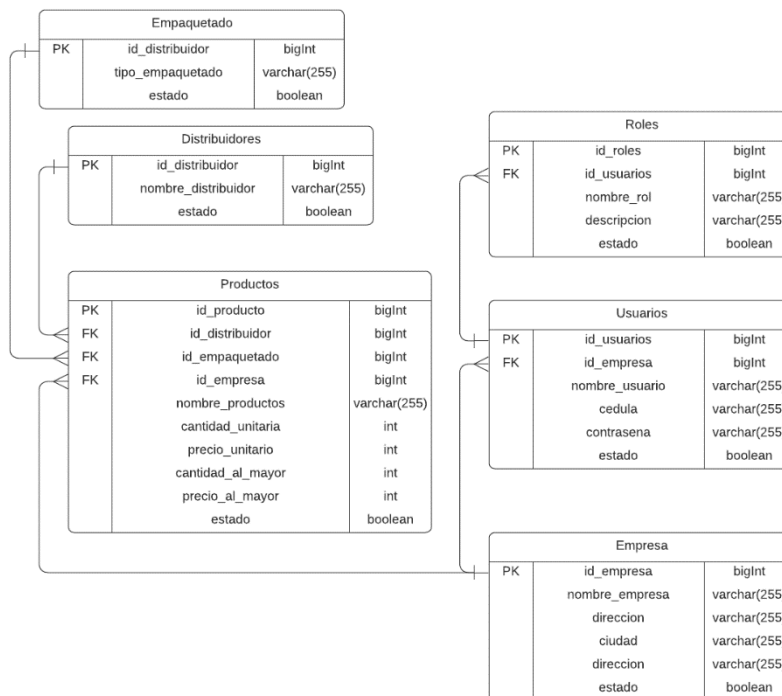


Figura 8: Diagrama Entidad - Relación  
 Fuente: Autor

## 6.2. Fase 2. Selección de tecnología

Herramienta utilizada	Para qué	Por qué
PostgreSQL	Motor de base de datos y almacenamiento de la información	Es un sistema de motor de bases de datos compatible con los diversos modelos de datos para la creación de aplicaciones orientada a objetos, potentes, escalables y la mayoría de las herramientas más populares del mercado. (OVHcloud, 2023)
Intellij Idea	Software para la codificación del Proyecto	Es un entorno de desarrollo para la creación de software informático, hoy en día es uno de los más utilizados junto con eclipse y NetBeans, ofreciendo posibilidades de ejecutar aplicaciones java, así mismo, como proyectos basados en Maven, Spring y muchos más. (DeveloPrograming, 2021)
Hibernate, Jpa, HTML y Spring Framework	Lenguajes de diseño y codificación del proyecto	<p>Hibernate. Es una capa de persistencia objeto/relación que permite diseñar objetos persistentes que incluyen polimorfismo, relaciones, colecciones y un gran número de tipos de datos, con gran estabilidad, eficiencia, de doble capa y utilidad para clúster. (Andalucia J. d., s.f.)</p> <p>JPA (Java Persistence API) Utiliza anotaciones de metadatos y/o archivos del descriptor XML para configurar el mapeo entre objetos de java en el ámbito de aplicación y tablas de base de datos relacional, siendo una solución ORM completa y soporta herencia y polimorfismo. (Andalucia J. , s.f.)</p> <p>HTML. Es parte fundamental de las normas web, en conjunto con otras tecnologías como CSS y JavaScript, suma importancia para desarrollo web. (Vadavo, 2023)</p> <p>Spring Framework. Se trata de uno de los entornos más populares y ayuda a desarrolladores Back-End a crear aplicaciones con un alto rendimiento empleando objetos de java sencillos, dando soporte para el desarrollo de aplicaciones y páginas web.</p>

Herramienta utilizada	Para qué	Por qué
		(School, 2023)
Bootstrap	Biblioteca para la toma de recursos para la creación del diseño de la interfaz de usuario	Es un framework que permite a desarrolladores web construir paginas responsives de una forma rápida y sencilla, proporcionando un conjunto de componentes y plantillas CSS, HTML y JavaScript, se pueden utilizar y modificar de forma gratuita. (Santander, 2023)

*Tabla 19: Selección de Tecnología*  
*Fuente: Autor*

## 6.3. Fase 3. Diseño del sistema

### 6.3.1. Mockups

#### ➤ Landing Page

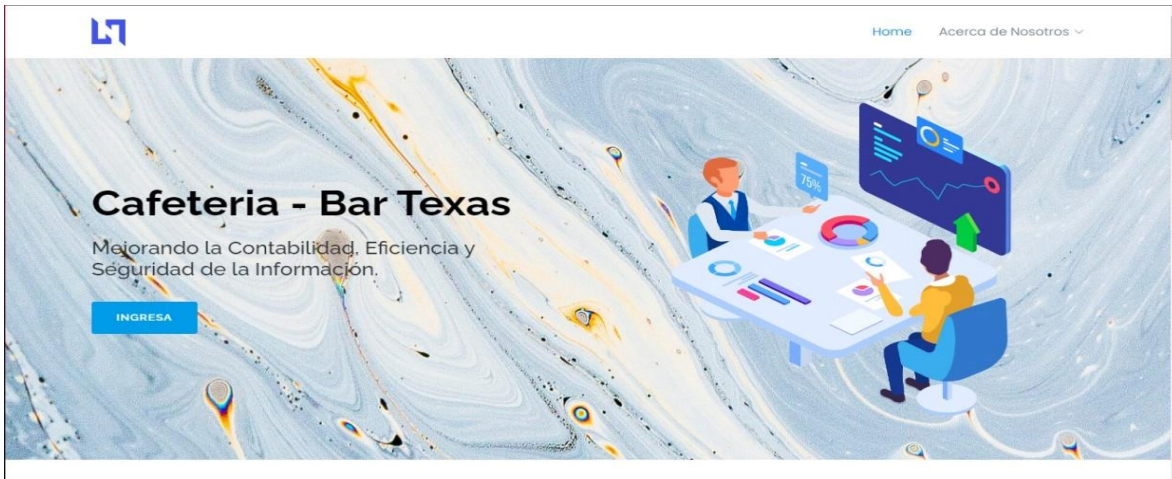


Figura 9: Mockup Home  
Fuente: Autor

#### ➤ Acerca de

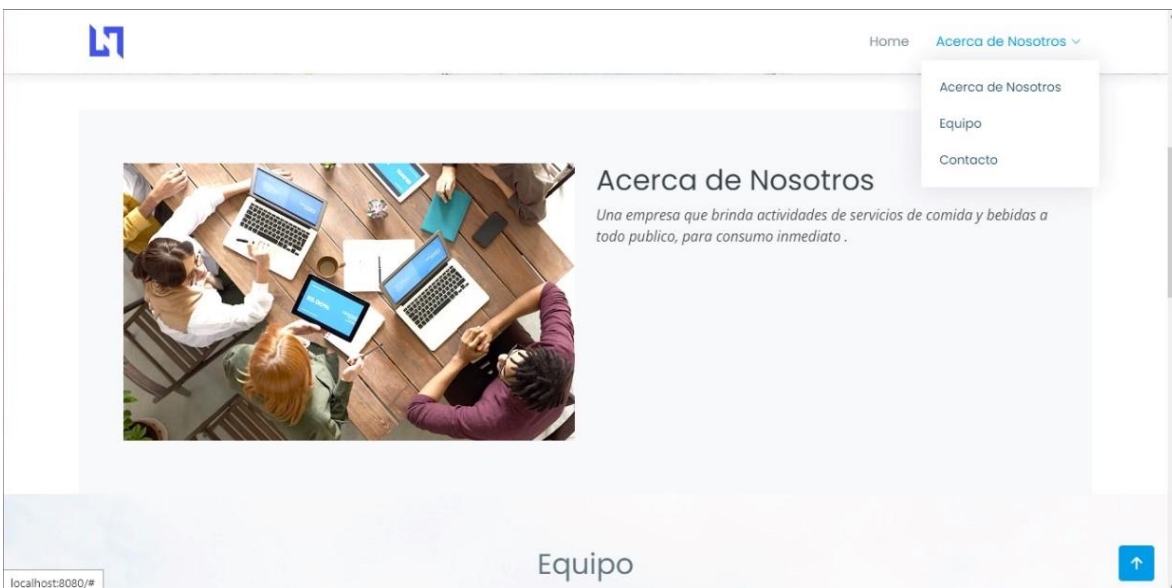


Figura 10: Mockup Acerca de Nosotros  
Fuente: Autor

➤ **Equipo de Desarrollo**

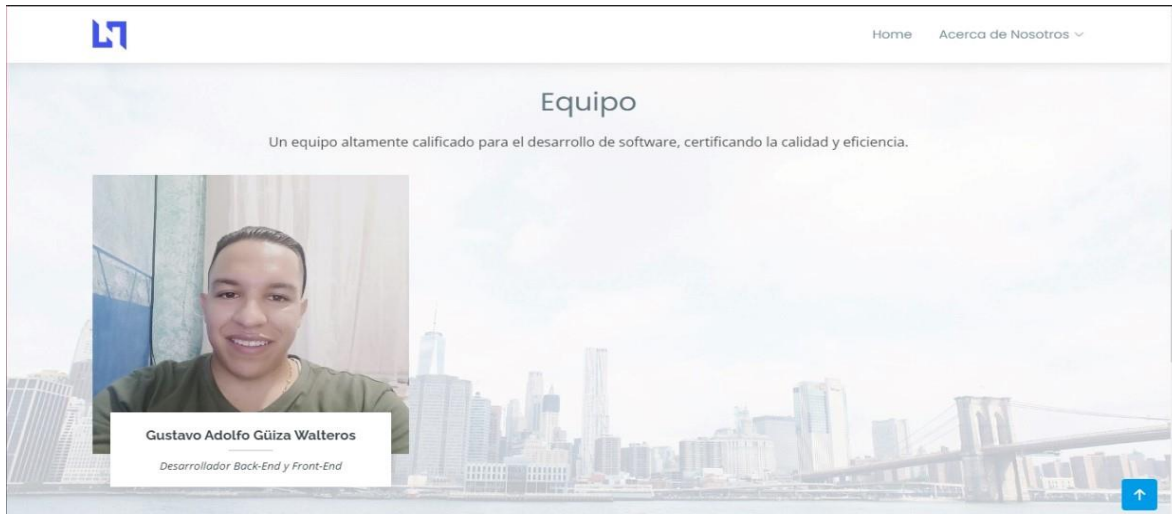


Figura 11: Mockup Equipo  
Fuente: Autor

➤ **Contacto**

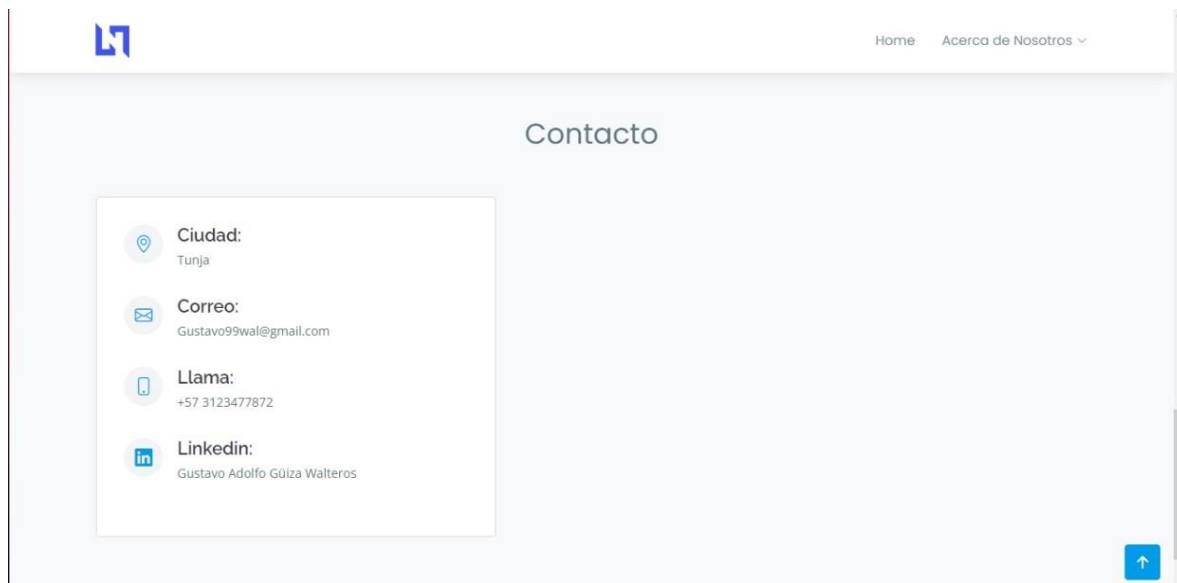
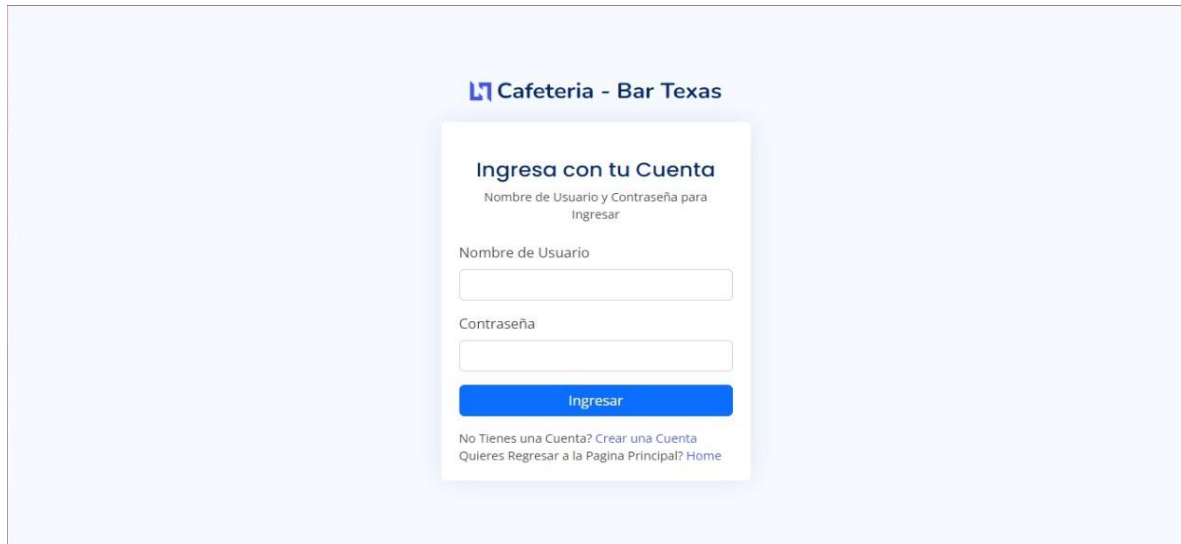


Figura 12: Mockup Contacto  
Fuente: Autor

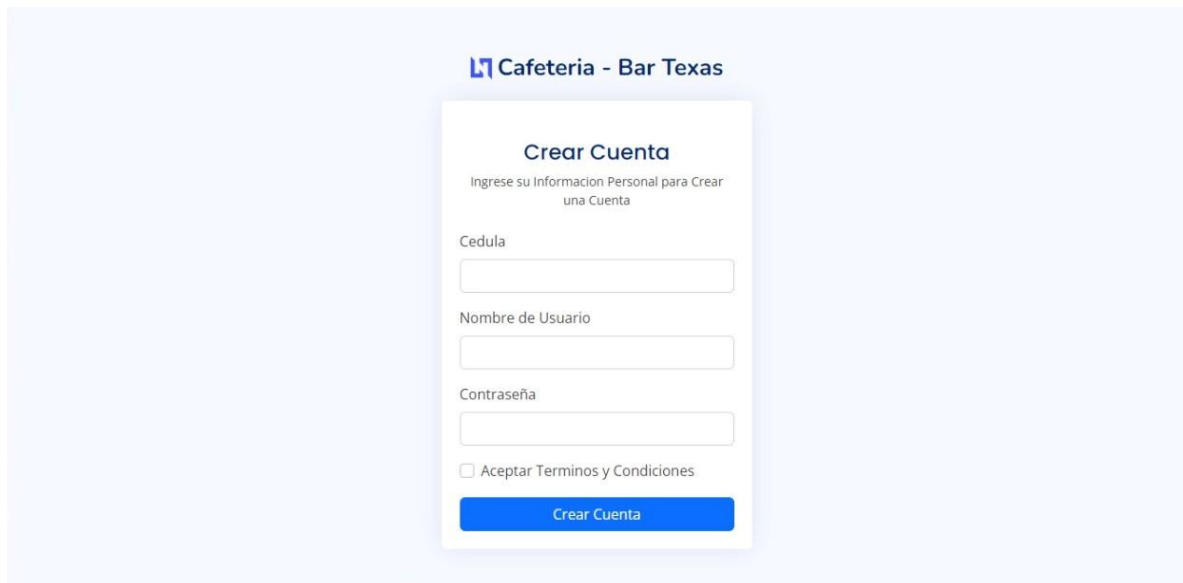
➤ **Login**



The mockup shows a login form titled "Cafeteria - Bar Texas" with the heading "Ingresa con tu Cuenta". Below the heading is the instruction "Nombre de Usuario y Contraseña para Ingresar". The form contains three input fields: "Nombre de Usuario", "Contraseña", and a blue "Ingresar" button. At the bottom, there are two links: "No Tienes una Cuenta? Crear una Cuenta" and "Quieres Regresar a la Pagina Principal? Home".

*Figura 13: Mockup Login*  
*Fuente: Autor*

➤ **Registro**



The mockup shows a registration form titled "Cafeteria - Bar Texas" with the heading "Crear Cuenta". Below the heading is the instruction "Ingrese su Informacion Personal para Crear una Cuenta". The form contains four input fields: "Cedula", "Nombre de Usuario", "Contraseña", and a checkbox for "Aceptar Terminos y Condiciones". A blue "Crear Cuenta" button is at the bottom.

*Figura 14: Mockup Registro*  
*Fuente: Autor*

➤ Dashboard

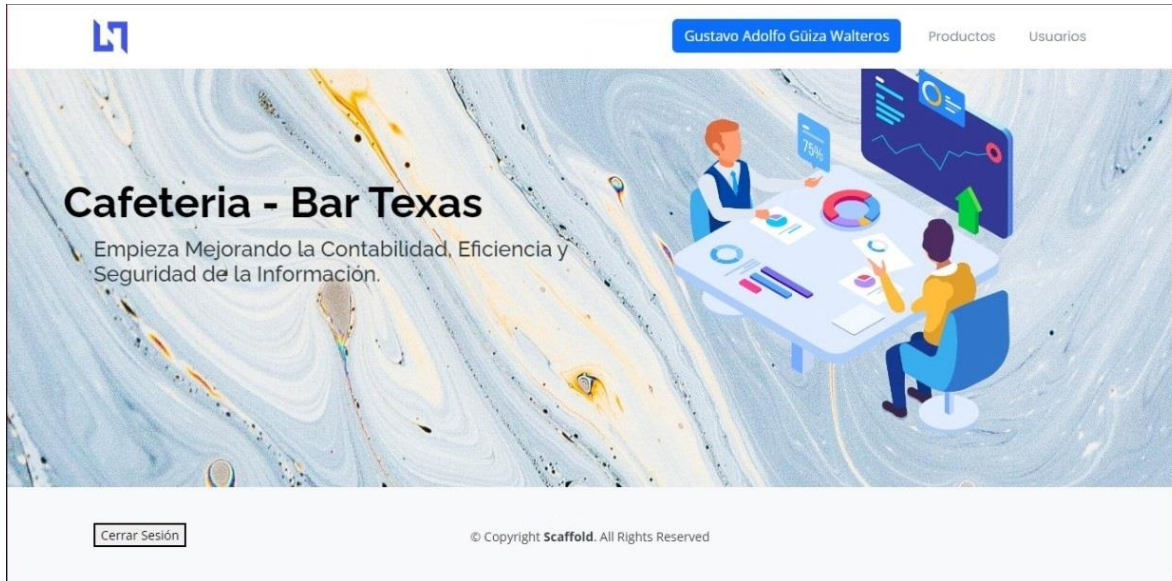


Figura 15: Mockup Dashboard  
Fuente: Autor

➤ Productos

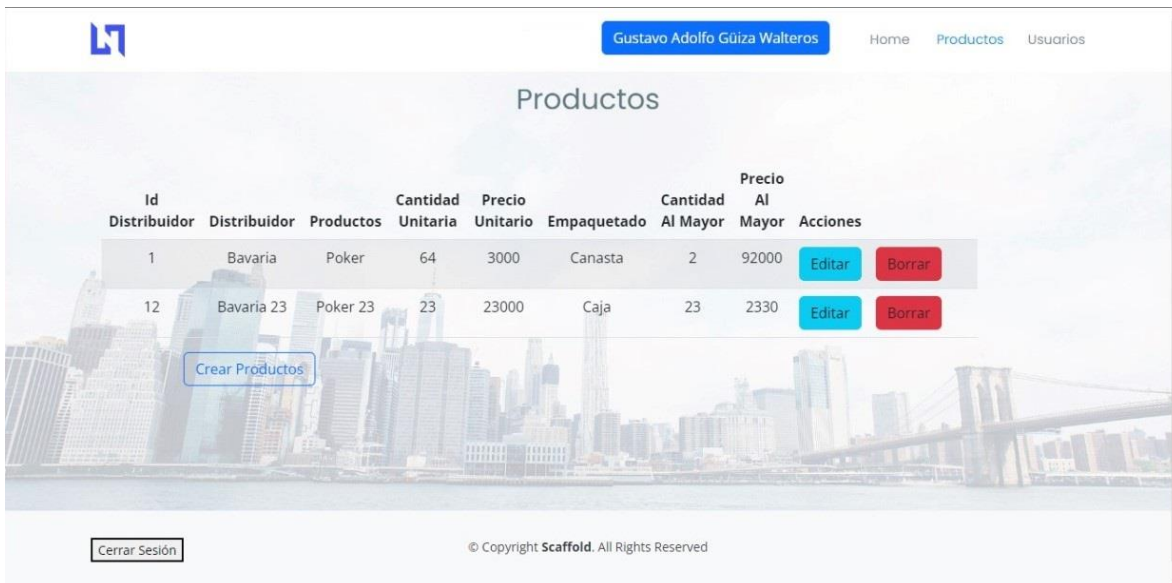


Figura 16: Mockup Productos  
Fuente: Autor

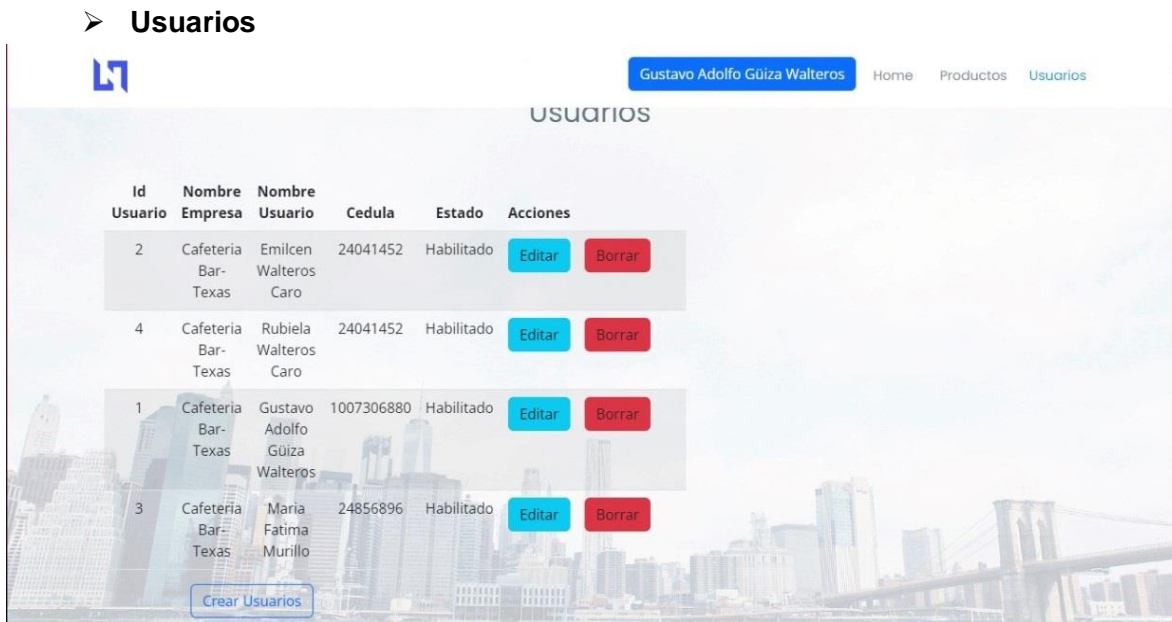


Figura 17: Mockup Recursos Humanos

Fuente: Autor

#### 6.4. Fase 4. Desarrollo de la aplicación

El desarrollo de la propuesta se hizo con el uso de las herramientas spring framework, JPA e Hibernate para la persistencia de base de datos, así mismo se utilizó HTML, Thymeleaf para el maquetado y configuración de estilos de los mockups. A continuación, se da a conocer cada una de las etapas que se tuvieron en cuenta:

##### 6.4.1. Creación de Proyecto

Se utiliza el IDE IntelliJ IDEA, donde se consume una página que le llama star.spring.io pasando los parámetros correspondientes como el nombre del aplicativo, lenguaje java, el gestor de paquetes que utilizará el sistema, grupo de trabajo al ser para un proyecto de trabajo de grado para la universidad queda com.usta, nombre del artefacto, la versión de java y JDK que se va a trabajar.

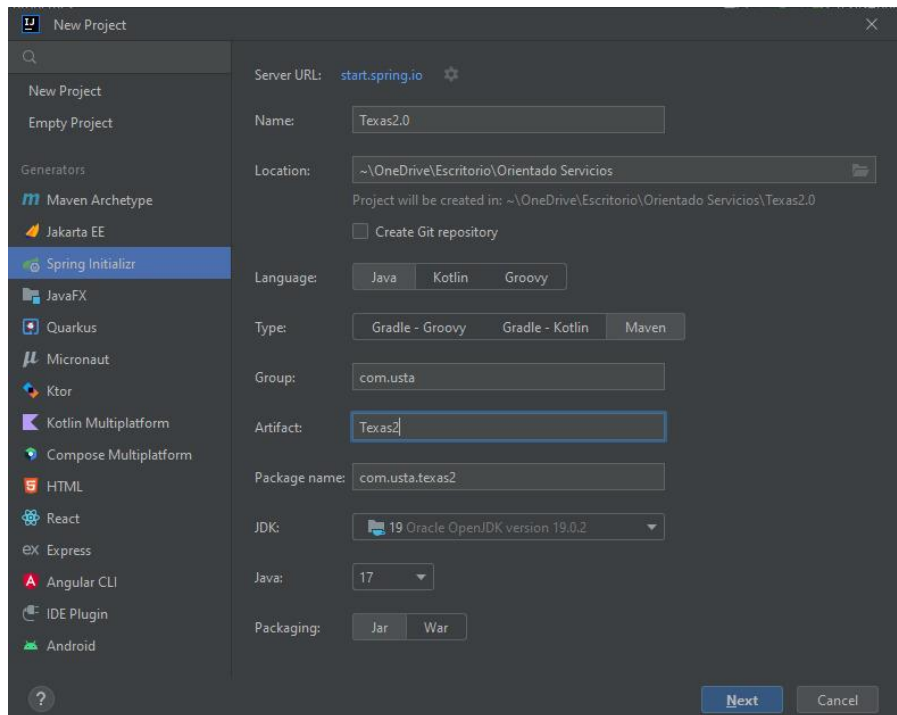
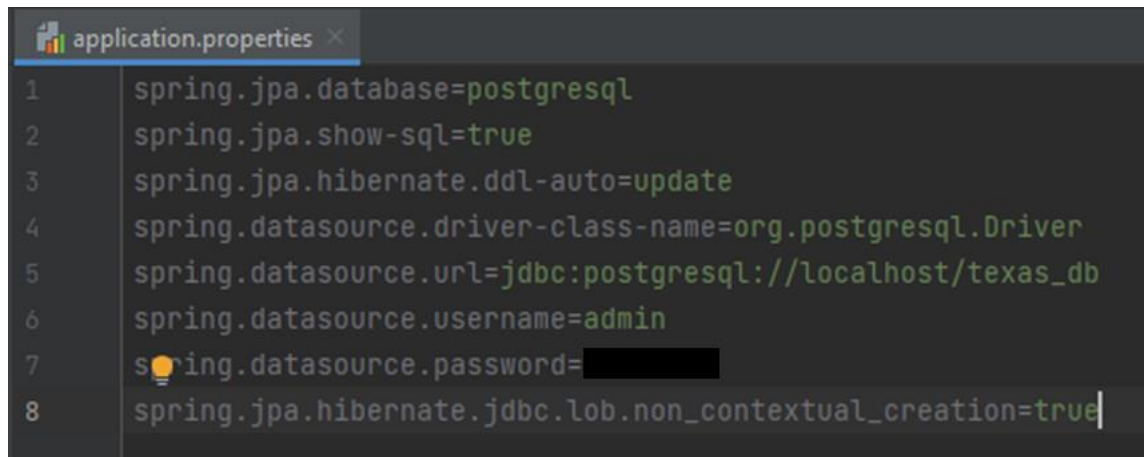


Figura 18: IntelliJ IDEA Creación del Proyecto  
Fuente: Autor

#### 6.4.2. Mapeo de las tablas de base de datos.

Al utilizar el patrón de arquitectura MVC, fue necesario mapear cada una de las tablas propuestas en el diagrama entidad - relación en una serie de archivos java que permitieran comunicar la lógica de negocios con cada una de las operaciones realizadas a nivel de la vista. Para ello una vez creado el proyecto, en el archivo application.properties correspondiente a la configuración del proyecto y con ayuda de Hibernate y JPA, se establecen los parámetros necesarios para la conexión de la base de datos ya que se utiliza por medio del gestor del repositorio Maven los drivers de postgres que se encuentra declarado en el archivo pom como una dependencia.

A screenshot of a code editor window titled 'application.properties'. The window contains eight lines of configuration text for a Spring application. The text is as follows:

```
1 spring.jpa.database=postgresql
2 spring.jpa.show-sql=true
3 spring.jpa.hibernate.ddl-auto=update
4 spring.datasource.driver-class-name=org.postgresql.Driver
5 spring.datasource.url=jdbc:postgresql://localhost/texas_db
6 spring.datasource.username=admin
7 spring.datasource.password=
8 spring.jpa.hibernate.jdbc.lob.non_contextual_creation=true
```

Figura 19: Application.properties

Fuente: Autor

Estos archivos java utilizan una serie de decoradores (@) para permitir que la aplicación una vez conectada sepa que existan tablas y estas tienen una serie de atributos y relaciones, el @Entity le da a entender a mi clase java que ya no va actuar únicamente como una clase si no que ésta interpretará una tabla que está en base de datos y para ello se hace uso de @Table que tiene un parámetro name y con él se le indica el nombre de la tabla a la cual va estar asociado nuestro archivo java; así mismo se tiene establecidas otra serie de anotaciones que en este caso facilitan la conversión de los tipos de datos de los atributos utilizados para la base de datos como por ejemplo VARCHAR a String, restricciones como NOT NULL, longitud de cadenas de texto o unicidad para atributos.

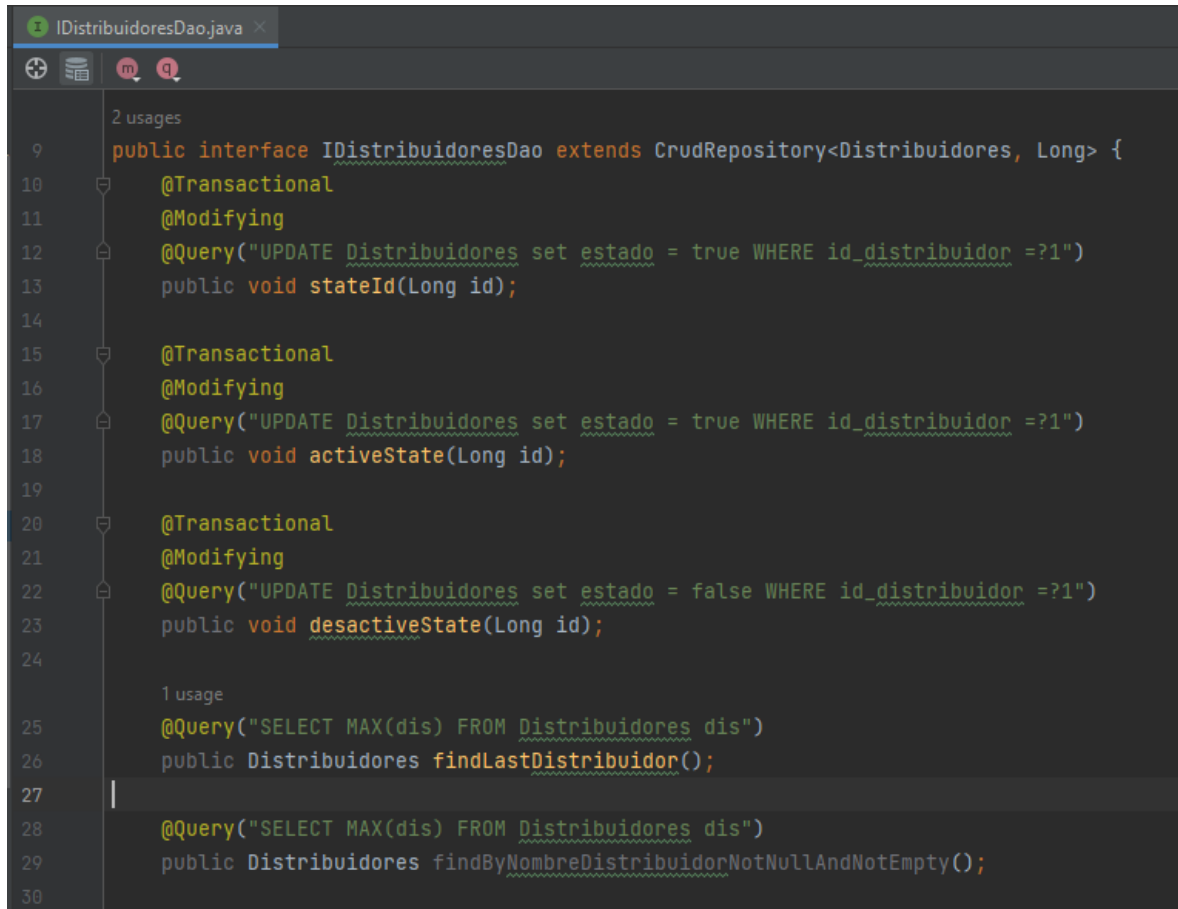
Estas decoraciones son el @column con el parámetro de name indicando en este caso el nombre del atributo que hace referencia en la tabla, @Id para indicar que el atributo es una llave primaria y los @JoinColumn, @ManyToOne, @OneToOne como las estructuras para generar las asociaciones entre las entidades mapeadas en base de datos.

#### 6.4.3. Creación de Archivos DAO

Los archivos DAO son interfaces que pueden definirse como la capa de persistencia de la aplicación al tener una colección de métodos que son declarados mas no implementado, y que al extender del CRUD Repository provee las operaciones básicas como crear, editar, eliminar y listar. Así mismo, se convirtieron en los repositorios que almacenaron las consultas SQL referentes a la lógica de negocio

de la aplicación; esto por medio del decorador @Query que al apuntar a una clase java facilitaba los procesos al hacer uso de POO.

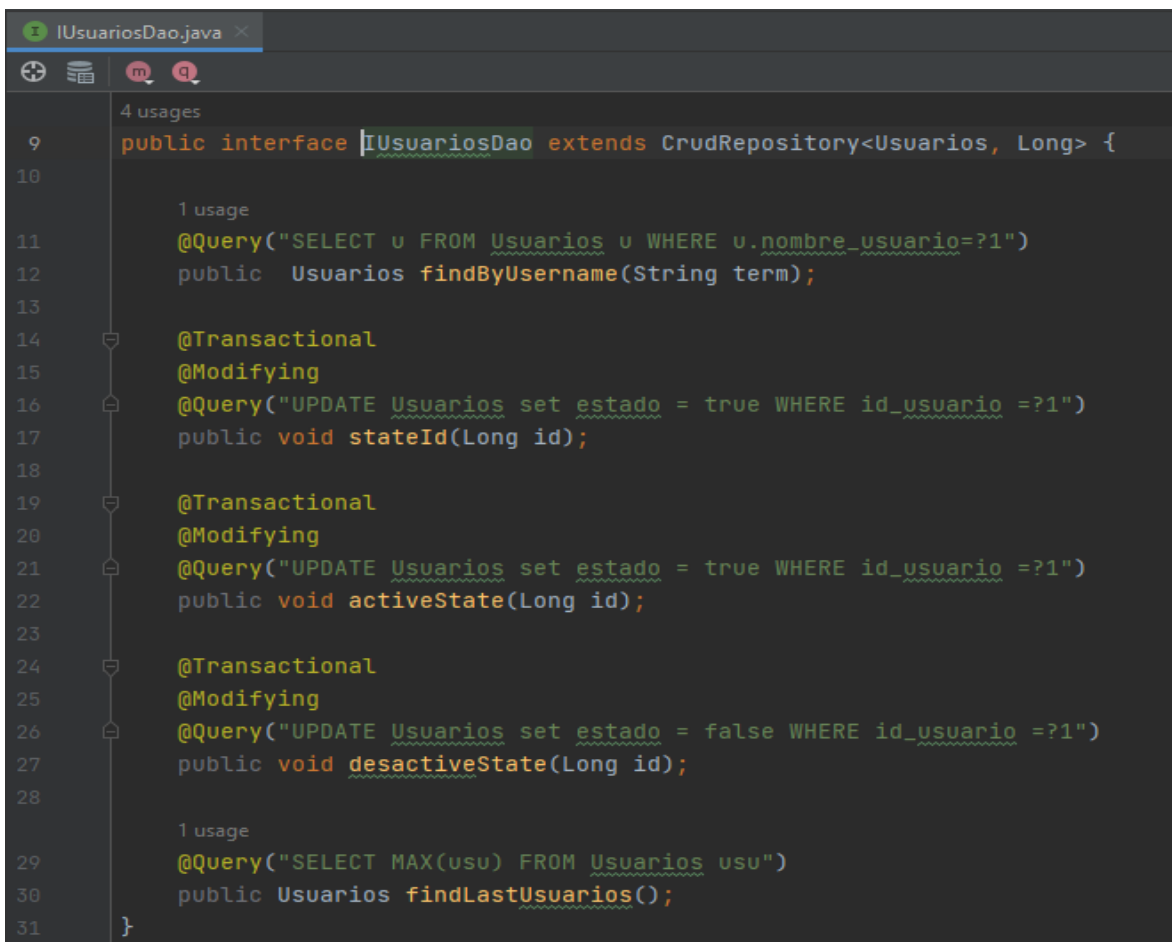
Cabe resaltar que el CRUD Repository es otra interfaz que pide como parámetros la entidad o archivo java al que hace referencia y el tipo de dato de la llave primaria que este tiene.



```
9 public interface IDistribuidoresDao extends CrudRepository<Distribuidores, Long> {
10     @Transactional
11     @Modifying
12     @Query("UPDATE Distribuidores set estado = true WHERE id_distribuidor =?1")
13     public void stateId(Long id);
14
15     @Transactional
16     @Modifying
17     @Query("UPDATE Distribuidores set estado = true WHERE id_distribuidor =?1")
18     public void activeState(Long id);
19
20     @Transactional
21     @Modifying
22     @Query("UPDATE Distribuidores set estado = false WHERE id_distribuidor =?1")
23     public void desactiveState(Long id);
24
25     1 usage
26     @Query("SELECT MAX(dis) FROM Distribuidores dis")
27     public Distribuidores findLastDistribuidor();
28
29     @Query("SELECT MAX(dis) FROM Distribuidores dis")
30     public Distribuidores findByNombreDistribuidorNotNullAndNotEmpty();
```

Figura 20: Distribuidores Dao

Fuente: Autor



```
9 public interface IUserariosDao extends CrudRepository<Usuarios, Long> {
10
11     1 usage
12     @Query("SELECT u FROM Usuarios u WHERE u.nombre_usuario=?1")
13     public Usuarios findByUsername(String term);
14
15     @Transactional
16     @Modifying
17     @Query("UPDATE Usuarios set estado = true WHERE id_usuario =?1")
18     public void stateId(Long id);
19
20     @Transactional
21     @Modifying
22     @Query("UPDATE Usuarios set estado = true WHERE id_usuario =?1")
23     public void activeState(Long id);
24
25     @Transactional
26     @Modifying
27     @Query("UPDATE Usuarios set estado = false WHERE id_usuario =?1")
28     public void desactiveState(Long id);
29
30     1 usage
31     @Query("SELECT MAX(usu) FROM Usuarios usu")
32     public Usuarios findLastUsuarios();
33 }
```

Figura 21: Usuarios Dao  
Fuente: Autor

#### 6.4.4. Creación de Servicios

Los servicios también son interfaces que permiten declarar mas no implementar los métodos que consumirán un método específico del DAO, esto incluye tanto las consultas CRUD como aquellas asociadas con la lógica de negocio.

```
IDistribuidoresServices.java x
1 package com.usta.texas.models.services;
2
3 import ...
7
6 usages 1 implementation
8 public interface IDistribuidoresServices {
9
10 1 implementation
11 public List<Distribuidores> findAll();
12
13 1 usage 1 implementation
14 public Distribuidores modiDistribuidor(Long id, Distribuidores modiDistribuidor);
15
16 1 implementation
17 public void guardar(Distribuidores distribuidores);
18
19 /* Metodo para eliminar por id - CRUDREPOSITORY */
20 1 implementation
21 public void borrarId(Long id);
22
23 /*Metodo Cambiar Estado*/
24 1 implementation
25 public void estadoId(Long id);
26
27 1 implementation
28 public void activarEstado(Long id);
```

Figura 22: Distribuidores Services Parte 1  
Fuente: Autor

```
1 implementation
2 public void borrarEstado(Long id);
3
4 1 implementation
5 public Distribuidores findOne(Long id);
6
7 1 implementation
8 public Distribuidores findLastDistribuidor();
9
10 1 implementation
11 List<Distribuidores> findByNombreDistribuidorNotNullAndNotEmpty();
```

Figura 23: Distribuidores Services Parte 2  
Fuente: Autor

### 6.4.5. Implementación de Servicios.

Estos son archivos Java que hacen uso de la herencia de la interfaz del servicio por medio de la palabra reservada `implements`, y pone en práctica los decoradores `@Service`, `@Autowired` que permiten utilizar los métodos declarados en el DAO como implementación de los métodos traídos del servicio.

```

1 package com.usta.texas.models.services.implement;
2
3 import ...
4
11
12
13 @Service
14 public class DistribuidoresServicesImplement implements IDistribuidoresServices {
15
16     10 usages
17     @Autowired
18     private IDistribuidoresDao IDistribuidoresDao;
19
20     /**
21      * @Desc Este metodo es la implementacion del metodo creado en mi servicio para
22      * la entidad de Bar Cafeteria Texas y retorna la lista de los registros.
23      * @Create 29/08/2022
24      * @autor Gustavo Adolfo Gúiza Walteross
25      * @required Interfaz Services
26      * **/
27
28     @Override
29     @Transactional
30     public List<Distribuidores> findAll() { return (List<Distribuidores>) IDistribuidoresDao.findAll(); }
    
```

Figura 24: Distribuidores Services Implement Parte 1

Fuente: Autor

```

33
34     /**
35      * @Desc Este metodo es la implementacion del metodo creado en mi servicio para
36      * la entidad de Bar Cafeteria Texas e inserta un nuevo registro.
37      * @Create 29/08/2022
38      * @autor Gustavo Adolfo Gúiza Walteross
39      * @required Interfaz Services
40      * @param distribuidores
41      * **/
42
43     @Override
44     @Transactional
45     public void guardar(Distribuidores distribuidores) { IDistribuidoresDao.save(distribuidores); }
46
47
48     /**
49      * @Desc Este metodo es la implementacion del metodo creado en mi servicio para
50      * la entidad de Bar Cafeteria Texas y elimina un registro de la base de datos por ID
51      * utilizando el CrudRepository.
52      * @Create 29/08/2022
53      * @autor Gustavo Adolfo Gúiza Walteross
54      * @required Interfaz Services
55      * @param id
56      * **/
57
58     @Override
59     @Transactional
60     public void borrarId(Long id) { IDistribuidoresDao.deleteById(id); }
    
```

Figura 25: Distribuidores Services Implement Parte 2  
Fuente: Autor

```
63  /**
64  * @Desc Este metodo es la implementacion del metodo creado en mi servicio para
65  * La entidad de Bar Cafeteria Texas y cambia el estado del registro por el ID.
66  * @Create 29/08/2022
67  * @autor Gustavo Adolfo Gúiza Walteross
68  * @required Interfaz Services
69  * @param id
70  * **/
71  @Override
72  @Transactional
73  public void estadoId(Long id) { IDistribuidoresDao.stateId(id); }
76
77  /**
78  * @Desc Este metodo es la implementacion del metodo creado en mi servicio para
79  * La entidad de Bar Cafeteria Texas y cambia el estado del registro por el ID.
80  * @Create 29/08/2022
81  * @autor Gustavo Adolfo Gúiza Walteross
82  * @required Interfaz Services
83  * @param id
84  * **/
85
86  @Override
87  @Transactional
88  public void borrarEstado(Long id) { IDistribuidoresDao.desactiveState(id); }
```

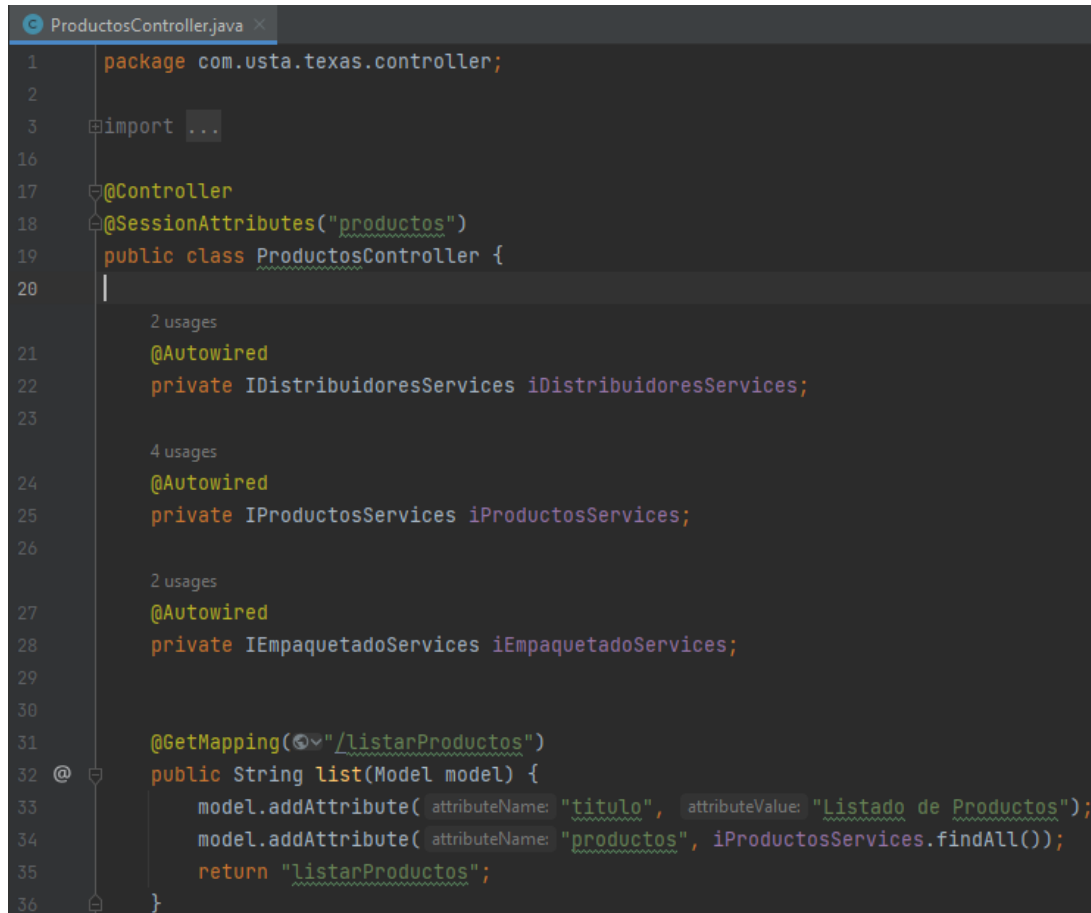
Figura 26: Distribuidores Services Implement Parte 3  
Fuente: Autor

```
92  @Override
93  @Transactional
94  public void activarEstado(Long id) { IDistribuidoresDao.activeState(id); }
97
98  @Override
99  @Transactional
100  public Distribuidores findOne(Long id) { return IDistribuidoresDao.findById(id).orElse( other: null); }
103
104  @Transactional(readOnly = true)
105  public Distribuidores findLastDistribuidor() { return IDistribuidoresDao.findLastDistribuidor(); }
108
109  @Override
110  public List<Distribuidores> findByNombreDistribuidorNotNullAndNotEmpty() { return null; }
113
114  1 usage
114  public Distribuidores modiDistribuidor(Long id, Distribuidores modiDistribuidor) {
115      Distribuidores distribuidores = IDistribuidoresDao.findById(id).orElse( other: null);
116      if (distribuidores != null) {
117          distribuidores.setNombre_distribuidor(modiDistribuidor.getNombre_distribuidor());
118          IDistribuidoresDao.save(distribuidores);
119      }
120      return null;
121  }
```

Figura 27: Distribuidores Services Implement Parte 4  
Fuente: Autor

#### 6.4.6. Mapeo del Front-End (HTML) con Controladores

Los controladores son archivos java que con el decorador `@Controller` facilita la comunicación entre el BackEnd y FrontEnd, esto con el uso de enrutadores que poseen los verbos de una petición HTTP (GET, POST, PUT, DELETE) acompañados de la palabra Mapping. Así mismo, se hace uso del Model como un parámetro que permite traer información de los servicios.



```
1 package com.usta.texas.controller;
2
3 import ...
4
5
6
7
8
9
10
11
12
13
14
15
16
17 @Controller
18 @SessionAttributes("productos")
19 public class ProductosController {
20
21     2 usages
22     @Autowired
23     private IDistribuidoresServices iDistribuidoresServices;
24
25     4 usages
26     @Autowired
27     private IProductosServices iProductosServices;
28
29     2 usages
30     @Autowired
31     private IEmpaquetadoServices iEmpaquetadoServices;
32
33     @GetMapping("/listarProductos")
34     public String list(Model model) {
35         model.addAttribute( attributeName: "titulo", attributeValue: "Listado de Productos");
36         model.addAttribute( attributeName: "productos", iProductosServices.findAll());
37         return "listarProductos";
38     }
39 }
```

Figura 28: Productos Controller Parte 1

Fuente: Autor

```
38 @GetMapping("/{createProductos}")
39 @ public String crear(Model model) {
40     model.addAttribute( attributeName: "productos", new Productos());
41     model.addAttribute( attributeName: "listar_distribuidores", iDistribuidoresServices.findAll());
42     model.addAttribute( attributeName: "listar_empaquetados", iEmpaquetadoServices.findAll());
43     model.addAttribute( attributeName: "titulo", attributeValue: "Crear Producto");
44     return "createProductos";
45 }
46
47 @GetMapping("/{createProductos/{id}")
48 @ public String editarProductoForm(@PathVariable Long id, Model model) {
49     Productos productos = iProductosServices.findOne(id);
50     model.addAttribute( attributeName: "productos", productos);
51     model.addAttribute( attributeName: "listar_distribuidores", iDistribuidoresServices.findAll());
52     model.addAttribute( attributeName: "listar_empaquetados", iEmpaquetadoServices.findAll());
53     return "createProductos";
54 }
55
56 @PostMapping("/{createProductos/{id}")
57 public String editarProducto(@PathVariable Long id, @ModelAttribute Productos actualizarProducto) {
58     iProductosServices.actulizarProducto(id, actualizarProducto);
59     return "redirect:/createProductos";
60 }
```

Figura 29: Productos Controller Parte 2

Fuente: Autor

#### 6.4.7. Creación de Interfaces

Las interfaces son un conjunto de archivos HTML que, acompañados de CSS, Bootstrap y Thymeleaf facilitan la construcción de las interfaces para que el navegador pueda mostrar al usuario los flujos declarados en las historias de usuario.

Por medio de una dependencia en el pom provee una capa de seguridad para la creación y autenticación de usuarios.

```
SpringSecurityConfig.java
1 package com.usta.texas;
2
3 import ...
13 |
14 @EnableGlobalMethodSecurity(securedEnabled = true)
15 @Configuration
16 public class SpringSecurityConfig extends WebSecurityConfigurerAdapter {
17
18     1 usage
19     @Autowired
20     private BCryptPasswordEncoder passwordEncoder;
21
22     1 usage
23     @Autowired
24     private LoginSuccessHandler successHandler;
25
26     1 usage
27     @Autowired
28     private JpaUserDetailsService userDetailsService;
```

Figura 28: SpringSecurityConfig Parte 1  
Fuente: Autor

Se utiliza una sección de roles para restringir los accesos a las vistas

```
47 @Override
48 protected void configure(HttpSecurity http) throws Exception {
49     http.authorizeRequests().antMatchers("/","/css/**","/js/**","/img/**","/scss/**","/vendor/**").permitAll()
50     .antMatchers("/listarProductos").permitAll()
51     .antMatchers("/listarDistribuidores").permitAll()
52     .antMatchers("/createProductos").hasAnyRole("SUPERADMIN","ADMIN")
53     .antMatchers("/createDistribuidores").hasAnyRole("SUPERADMIN","ADMIN")
54     .antMatchers("/listarUsuarios").hasAnyRole("SUPERADMIN","ADMIN")
55     .antMatchers("/createUsuarios").hasAnyRole("SUPERADMIN","ADMIN")
56     .antMatchers("/dashboard").permitAll()
57     .antMatchers("/registrar").permitAll()
58     .antMatchers("/").permitAll()
59     .antMatchers("/index").permitAll()
60     .antMatchers("/notFound").permitAll()
61     .and() HttpSecurity
62     .formLogin().successHandler(successHandler) FormLoginConfigurer<HttpSecurity>
63     .loginPage("/login").permitAll()
64     .loginProcessingUrl("/login")
65     .defaultSuccessUrl(defaultSuccessUrl: "/dashboard", alwaysUse: true)
66     .and() HttpSecurity
67     .logout().permitAll() LogoutConfigurer<HttpSecurity>
68     .and() HttpSecurity
69     .exceptionHandling().accessDeniedPage(accessDeniedUrl: "/notFound");
70 }
```

Figura 30: SpringSecurityConfig Parte 2  
Fuente: Autor

```
52 @Autowired
53 @ public void configurerGlobal(AuthenticationManagerBuilder builder) throws Exception{
54
55     builder.userDetailsService(userDetailsService)
56     .passwordEncoder(passwordEncoder);
57 }
```

Figura 30: SpringSecurityConfig Parte 3  
Fuente: Autor

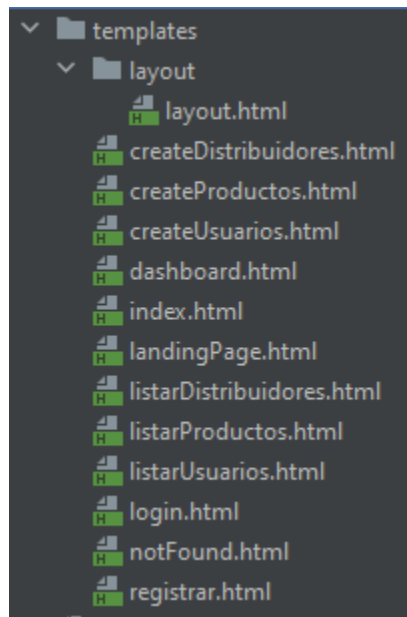


Figura 31: Archivos HTML  
Fuente: Autor

En la anterior figura mostrada o la figura 31 se visualizan los HTML donde fue codificado el FrontEnd para el desarrollo de la interfaz de usuario del sistema, a continuación, se presentarán fragmentos de código de cada archivo.

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org"
      xmlns:sec="http://www.thymeleaf.org/extras/spring-security">
<!-- HEAD -->
<head th:fragment="head">
  <title>Texas Cafeteria-Bar</title>
  <meta charset="UTF-8" /> <!-- Codificación de caracteres -->
  <meta name="description" content="Aplicativo Web de Sistema de Gestion de Inventario" />
  <meta name="author" content="Gustavo Adolfo Gúiza Walteros" />
  <meta name="keywords" content="Aplicativo Web, Contabilidad, Gestion de Inventario" />
  <link rel="stylesheet" th:href="@{/css/bootstrap.min.css}"/>
  <link rel="stylesheet" th:href="@{/css/styles.css}"/>
  <link rel="stylesheet" th:href="@{/css/login.css}"/>
  <link rel="stylesheet" th:href="@{/css/dashboard.css}"/>
  <link rel="stylesheet" th:href="@{/css/jquery-ui.min.css}"/>
  <link rel="stylesheet"
        href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"
        integrity="sha384-gg0yR0iXcBMqV3Xiipma34MD+dH/1fQ784/j6cY/iJTQU0hcWr7x9JvoRxt2MZw1T"
        crossorigin="anonymous"/>
  <link rel="stylesheet"
        href="https://use.fontawesome.com/releases/v5.7.0/css/all.css"
        integrity="sha384-LZN37f5Q6tY3VHg1sS14W3ExzMWZxybE1SJSesQp9S+oqd12jhcu+A56Ebc1zFSJ"
        crossorigin="anonymous"/>
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/twitter-bootstrap@4.1.3/css/bootstrap.css">
  <link rel="stylesheet" href="https://cdn.datatables.net/1.10.20/css/dataTables.bootstrap4.min.css">
```

Figura 32: Layout.html  
Fuente: Autor

```
<div class="d-flex justify-content-center py-4">
  <a class="logo d-flex align-items-center w-auto">
    
    <span class="d-none d-lg-block">Texas Cafeteria-Bar</span>
  </a>
</div>
<!-- End Logo -->

<div class="card mb-3">
  <div class="card-body">
    <div class="pt-4 pb-2">
      <h5 class="card-title text-center pb-0 fs-4">Ingresa un Distribuidor </h5>
    </div>

    <form class="row" th:object="{distribuidores}" th:action="{createDistribuidores}" method="post" novalidate>
      <br>
      <label for="nombre_distribuidor">Nombre Distribuidor: </label>
      <input id="nombre_distribuidor" type="text" placeholder="Ingresa el Nombre del Distribuidor" class="form-control" name="nombre_d
      <button class="btn btn-primary w-100" type="submit">Guardar</button>
    </form>
  </div>
</div>
</div>
```

Figura 33: createDistribuidor.html

Fuente: Autor

```
<label for="nombre_producto">Nombre Producto: </label>
<input id="nombre_producto" type="text" placeholder="Ingresa el Nombre del Producto" class="form-control" name="nombre_productos" th:field

<label for="cantidad_unitaria">Cantidad Unitaria: </label>
<input id="cantidad_unitaria" type="text" placeholder="Ingresa la Cantidad Unitaria" class="form-control" name="cantidad_unitaria" th:fiel

<label for="precio_unitario">Precio Unitario: </label>
<input id="precio_unitario" type="text" placeholder="Ingresa el Precio Unitario" class="form-control" name="precio_unitario" th:field="*{p

<div class="form-group">
  <label for="tipo_empaquetado">Tipo Empaquetado:</label>
  <select class="form-control" id="tipo_empaquetado" name="id_empaquetado">
    <option selected>Seleccionar</option>
    <option th:each="productos : {listar_empaquetados}" th:value="{productos.id_empaquetado}" th:text="{productos.tipo_empaquetado}
  </select>
</div>

<label for="cantidad_al_mayor">Cantidad al Mayor: </label>
<input id="cantidad_al_mayor" type="text" placeholder="Ingresa la Cantidad al Mayor" class="form-control" name="cantidad_al_mayor" th:fiel

<label for="precio_al_mayor">Precio al Mayor: </label>
<input id="precio_al_mayor" type="text" placeholder="Ingresa el Precio al Mayor" class="form-control" name="precio_al_mayor" th:field="*{p

<button class="btn btn-primary w-100" type="submit">Guardar</button>
```

Figura 34: createProductos.html

Fuente: Autor

```
<div class="card mb-3">
  <div class="card-body">

    <div class="pt-4 pb-2">
      <h5 class="card-title text-center pb-0 fs-4">Cuenta</h5>
      <p class="text-center small">Ingrese su Información Personal para Crear una Cuenta</p>
    </div>

    <form class="row g-3 needs-validation" th:action="@{/createUsuarios}" method="post" novalidate>

      <div class="col-12">
        <label for="cedula" class="form-label">Cedula</label>
        <input type="text" name="cedula" class="form-control" id="cedula" required>
        <div class="invalid-feedback">Por favor, Ingrese su Cedula!</div>
      </div>

      <div class="col-12">
        <label for="yourUsername" class="form-label">Nombre de Usuario</label>
        <div class="input-group has-validation">
          <input type="text" name="nombre_usuario" class="form-control" id="yourUsername" required>
          <div class="invalid-feedback">Por Favor, Ingrese el Usuario.</div>
        </div>
      </div>
    </div>
  </div>
```

Figura 35: createUsuarios.html

Fuente: Autor

```
<nav id="navbar" class="navbar order-last order-lg-0">
  <ul>
    <li><a class="nav-link scrollto" href="#dashboard">Home</a></li>
    <li class="nav-item dropdown">
      <a class="nav-link dropdown-toggle" href="#" id="navbarDropdown" role="button" data-bs-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
        Empresa
      </a>
      <div class="dropdown-menu" aria-labelledby="navbarDropdown">
        <a class="dropdown-item" href="#listarDistribuidores">Distribuidores</a>
        <a class="dropdown-item" href="#listarProductos">Productos</a>
      </div>
    </li>
    <li><a class="nav-link scrollto" href="#listarUsuarios">Recursos Humanos</a></li>

    <li class="nav-link scrollto" style="margin-left: 20px;"> <!-- Puedes ajustar el valor del margen según tus necesidades -->
      <form sec:authorize="isAuthenticated()" id="logoutForm" th:action="@{/logout}" method="post">
        <button class="btn btn-outline-dark" onclick="document.getElementById('logoutForm').submit();" type="submit">
          logout
        </button>
        <input type="hidden" th:name="{_csrf.parameterName}" th:value="{_csrf.token}"/>
      </form>
    </li>
  </ul>
```

Figura 36: Dashboard.html

Fuente: Autor

```
!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org" xmlns:sec="http://www.thymeleaf.org/extras/spring-security" lang="en">
<head th:replace="layout/layout :: head">
</head>
<body>
<div th:replace="layout/layout :: informacion"></div>
<header th:replace="layout/layout :: header"></header>
<br/>
<br/>
<footer th:replace="layout/layout :: footer"></footer>
</body>
</html>
```

Figura 37: Index.html

Fuente: Autor

```
<!-- ===== About Section ===== -->
<section id="about" class="about">
  <div class="container">
    <div class="row">
      <div class="col-lg-6" data-aos="zoom-in">
        
      </div>
      <div class="col-lg-6 d-flex flex-column justify-content-center" data-aos="fade-left">
        <div class="content pt-4 pt-lg-0">
          <h3>Acerca de Nosotros</h3>
          <p class="fst-italic">
            Estamos destinados a la venta de alimentos y bebidas para todo publico,
            con el fin de brindar la mejor eficiencia y atencion al cliente, dando un espacio sano
            y comodo para todas las personas.
          </p>
        </div>
      </div>
    </div>
  </div>
</section><!-- End About Section -->

<!-- ===== Team Section ===== -->
<section id="team" class="team">
  <div class="container">
```

Figura 38: LandingPage.html

Fuente: Autor

```
<div class="member" data-aos="zoom-in" data-aos-delay="200">
  <div class="container">
    <table class="table table-striped">
      <thead>
        <tr>
          <th scope="col-2">Nombre</th>
          <th scope="col-3">Estado</th>
          <th scope="col-4">Acciones</th>
          <th scope="col-5"></th>
        </tr>
      </thead>
      <tbody>
        <tr th:each="distribuidores:${distribuidores}">
          <td class="col-2" th:text="${distribuidores.nombre_distribuidor}"></td>
          <td th:if="${distribuidores.estado} == true" th:text="Habilitado"></td>
          <td th:if="${distribuidores.estado} == false" th:text="Deshabilitado"></td>
          <th class="col-4"><a class="btn btn-info btn-xs" th:href="@{/createDistribuidores/} + ${distribuidores.id_distribuidor}" th:text="
            <td <a class="btn bg-danger btn-xs" th:href="@{/borrarDistribuidor/} + ${distribuidores.id_distribuidor}" th:text="Borrar" onclick="return confirm('Seguro
          </tr>
        </tbody>
      </table>
    <a th:href="@{/createDistribuidores}" class="btn btn-outline-primary">Crear Distribuidor</a>
  </div>
</div>
```

Figura 39: listarDistribuidores.html

Fuente: Autor

```
<input class="form-control mb-3" id="searchInput" type="text" placeholder="Buscar...">
<table class="table table-striped">
  <thead>
    <tr>
      <th scope="col">Nombre</th>
      <th scope="col">Distribuidor</th>
      <th scope="col">Cantidad Unitaria</th>
      <th scope="col">Precio Unitario</th>
      <th scope="col">Empaquetado</th>
      <th scope="col">Cantidad Al Mayor</th>
      <th scope="col">Precio Al Mayor</th>
      <th scope="col">Acciones</th>
    </tr>
  </thead>
  <tbody>
    <tr th:each="productos:${productos}">
      <td class="filterable" th:text="${productos.nombre_productos}"></td>
      <td class="filterable" th:text="${productos.id_distribuidor.nombre_distribuidor}"></td>
      <td th:text="${productos.cantidad_unitaria}"></td>
      <td th:text="${productos.precio_unitario}"></td>
      <td th:text="${productos.id_empaquetado.tipo_empaquetado}"></td>
      <td th:text="${productos.cantidad_al_mayor}"></td>
      <td th:text="${productos.precio_al_mayor}"></td>
      <th class="col-4"><a class="btn btn-info btn-xs" th:href="@{/createProductos/} + ${productos.id_producto}" th:text="Editar"></a></th>
      <td <a class="btn bg-danger btn-xs" th:href="@{/borrarProducto/} + ${productos.id_producto}" th:text="Borrar" onclick="return confirm('Seguro
    </tr>
  </tbody>
</table>
```

Figura 40: listarProductos.html

Fuente: Autor

```
<div class="member" data-aos="zoom-in" data-aos-delay="200">
  <div class="container-fluid">
    <table class="table table-striped">
      <thead>
        <tr>
          <th scope="col">Nombre Usuario</th>
          <th scope="col">Cedula</th>
          <th scope="col">Estado</th>
          <th scope="col">Acciones</th>
          <th scope="col"></th>
          <th scope="col"></th>
        </tr>
      </thead>
      <tbody>
        <tr th:each="usuarios: ${usuarios}">
          <td th:text="${usuarios.nombre_usuario}"></td>
          <td th:text="${usuarios.cedula}"></td>
          <td th:if="${usuarios.estado} == true" th:text="Habilitado"></td>
          <td th:if="${usuarios.estado} == false" th:text="Deshabilitado"></td>
          <td><a class="btn btn-info btn-xs" th:href="@{/createUsuarios/} + ${usuarios.id_usuario}" th:text="Editar"></a></td>
          <td><a class="btn bg-danger btn-xs" th:href="@{/borrarUsuarios/} + ${usuarios.id_usuario}" th:text="Borrar" onclick="return confirm('S"></td>
        </tr>
      </tbody>
    </table>
  </div>
</div>
```

Figura 41: listarUsuarios.html

Fuente: Autor

```
<form class="row g-3 needs-validation" th:action="@{/login}" method="post" novalidate>
  <div class="col-12">
    <label for="username" class="form-label">Nombre de Usuario</label>
    <div class="input-group has-validation">
      <input type="text" name="username" class="form-control" id="username" required>
      <div class="invalid-feedback">Por Favor Ingrese el Nombre de Usuario!</div>
    </div>
  </div>
  <div class="col-12">
    <label for="password" class="form-label">Contraseña</label>
    <input type="password" name="password" class="form-control" id="password" required>
    <div class="invalid-feedback">Por Favor Ingrese la Contraseña!</div>
  </div>
  <div class="col-12">
    <button class="btn btn-primary w-100" type="submit">Ingresar</button>
  </div>
  <div class="col-12">
    <p class="small mb-0">No Tienes una Cuenta? <a href="/registrar">Crear una Cuenta</a></p>
    <p class="small mb-0">Quieres Regresar a la Pagina Principal? <a href="/">Home</a></p>
  </div>
</form>
```

Figura 42: login.html

Fuente: Autor

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="utf-8">
  <meta content="width=device-width, initial-scale=1.0" name="viewport">

  <title>Not Found 404</title>
  <meta content="" name="description">
  <meta content="" name="keywords">

  <!-- Favicons -->
  <link href="assets/img/favicon.png" rel="icon">
  <link href="assets/img/apple-touch-icon.png" rel="apple-touch-icon">

  <!-- Google Fonts -->
  <link href="https://fonts.googleapis.com/css?family=Open+Sans:300,300i,400,400i,600,600i,700,700i|Raleway:300,300i,400,400i,500,500i,600,600i,700,700i|Pop

  <!-- Vendor CSS Files -->
  <link href="assets/vendor/aos/aos.css" rel="stylesheet">
  <link href="assets/vendor/bootstrap/css/bootstrap.min.css" rel="stylesheet">
  <link href="assets/vendor/bootstrap-icons/bootstrap-icons.css" rel="stylesheet">
  <link href="assets/vendor/boxicons/css/boxicons.min.css" rel="stylesheet">
  <link href="assets/vendor/glightbox/css/glightbox.min.css" rel="stylesheet">
  <link href="assets/vendor/swiper/swiper-bundle.min.css" rel="stylesheet">
```

Figura 43: notFound.html

Fuente: Autor

```
<div class="col-12">
  <label for="yourUsername" class="form-label">Nombre de Usuario</label>
  <div class="input-group has-validation">
    <input type="text" name="nombre_usuario" class="form-control" id="yourUsername" required>
    <div class="invalid-feedback">Por Favor, Ingrese el Usuario.</div>
  </div>
</div>

<div class="col-12">
  <label for="contrasena" class="form-label">Contraseña</label>
  <input type="password" name="contrasena" class="form-control" id="contrasena" required>
  <div class="invalid-feedback">Por Favor, Ingrese la Contraseña!</div>
</div>

<div class="col-12">
  <div class="form-check">
    <input class="form-check-input" name="terms" type="checkbox" value="" id="acceptTerms" required>
    <label class="form-check-label" for="acceptTerms">Aceptar Terminos y Condiciones</label>
    <div class="invalid-feedback">Debe de estar de Acuerdo con Los Terminos.</div>
  </div>
</div>

<div class="col-12">
  <button href="registrar" class="btn btn-primary w-100" type="submit" >Crear Cuenta</button>
  <p class="small mb-0">Quieres Regresar a la Pagina de Login? <a href="/login">Login</a></p>
</div>
```

Figura 44: registrar.html

Fuente: Autor

## 6.5. Fase 5. Pruebas



Pruebas
<b>N° de Prueba: 001</b>
<b>Descripción:</b> Se hace la prueba de los hipervínculos de las distintas pestañas situados en la landing page.
<b>Proceso:</b> Nos dirigimos al apartado de home para hacer la prueba del botón ingresar.
<b>Resultado:</b> Al dar clic en el botón este redirige correctamente al formulario de Login para ingresar al sistema.
<b>Evidencia:</b>  <p><i>Figura 45: Evidencia 1</i> <i>Fuente: Autor</i></p>  <p><i>Figura 46: Evidencia 2</i> <i>Fuente: Autor</i></p>

Tabla 20: Prueba 1

Fuente: Autor

Pruebas
<b>N° de Prueba: 002</b>
<b>Descripción:</b> Se hace la prueba de los hipervínculos de las distintas pestañas situados en la landing page.
<b>Proceso:</b> Nos dirigimos al apartado de la esquina derecha de la pantalla en el desplegable que dice Texas Cafetería-Bar y damos clic en Acerca de Nosotros.
<b>Resultado:</b> Al dar clic la página se desplaza a la sección de acerca de nosotros donde se visualiza una pequeña descripción.

**Evidencia:**

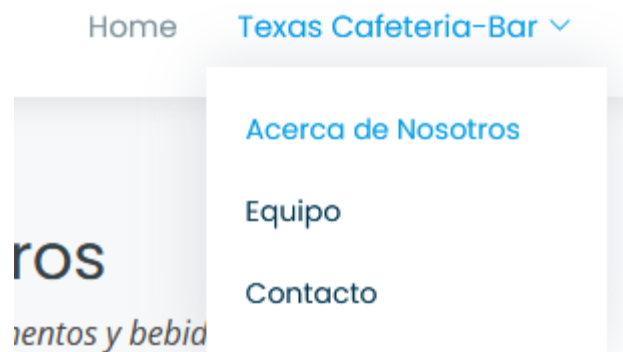


Figura 47: Evidencia 3

Fuente: Autor



### Acerca de Nosotros

Estamos destinados a la venta de alimentos y bebidas para todo público, con el fin de brindar la mejor eficiencia y atención al cliente, dando un espacio sano y comodo para todas las personas.

Figura 48: Evidencia 4

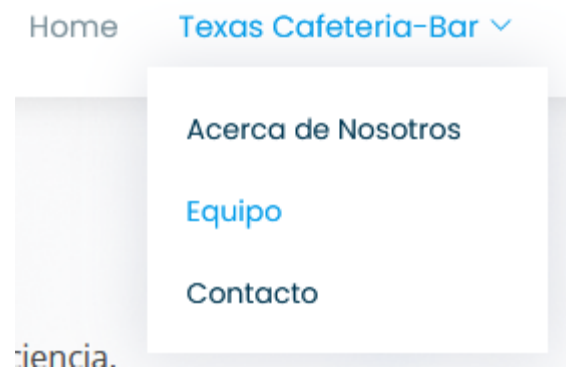
Fuente: Autor

Tabla 21: Prueba 2

Fuente: Autor

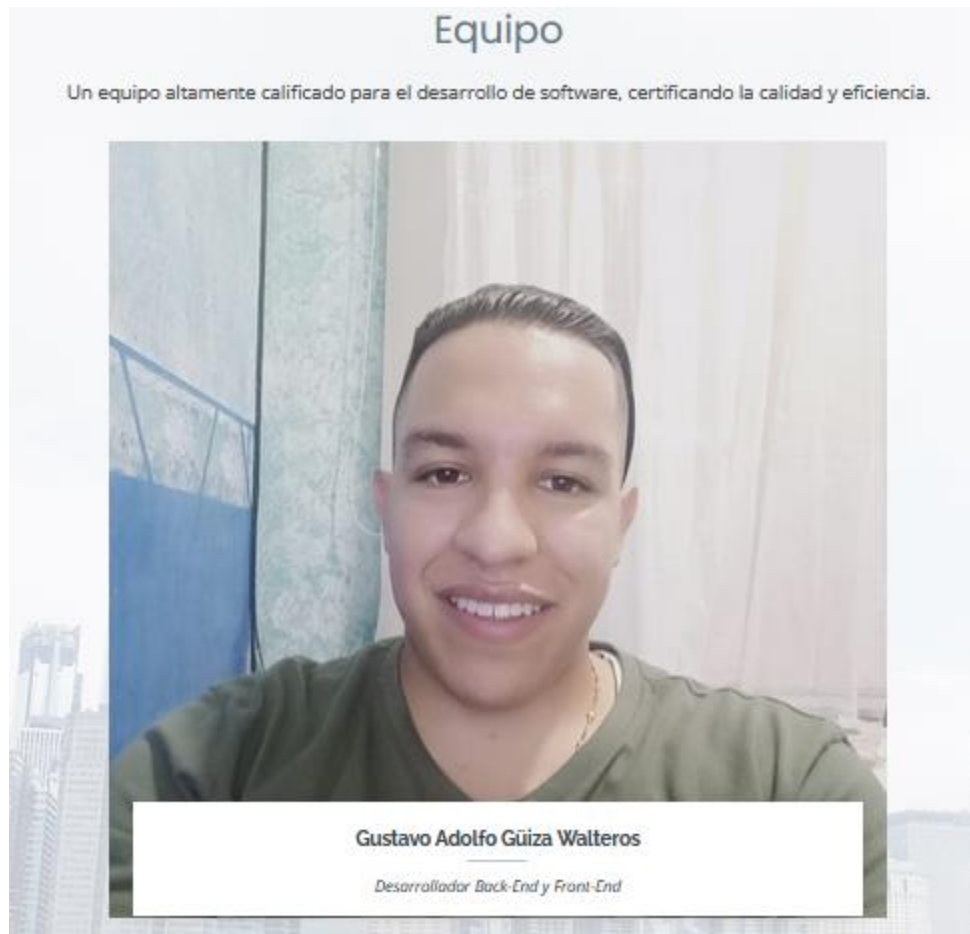
Pruebas
<b>N° de Prueba: 003</b>
<b>Descripción:</b> Se hace la prueba de los hipervínculos de las distintas pestañas situados en la landing page.
<b>Proceso:</b> Nos dirigimos al apartado de la esquina derecha de la pantalla en el desplegable que dice Texas Cafetería-Bar y damos clic en Equipo.
<b>Resultado:</b> Al dar clic la página se desplaza a la sección de Equipo donde se visualiza los integrantes que desarrollaron el aplicativo.

**Evidencia:**



*Figura 49: Evidencia 5*

*Fuente: Autor*



*Figura 50: Evidencia 6*

*Fuente: Autor*

*Tabla 22: Prueba 3*

*Fuente: Autor*

<b>Pruebas</b>
<b>N° de Prueba: 004</b>
<b>Descripción:</b> Se hace la prueba de los hipervínculos de las distintas pestañas situados en la landing page.
<b>Proceso:</b> Nos dirigimos al apartado de la esquina derecha de la pantalla en el desplegable que dice Texas Cafetería-Bar y damos clic en Contacto.
<b>Resultado:</b> Al dar clic la página se desplaza a la sección de Contacto donde se visualiza la información de los integrantes que desarrollaron el aplicativo.
<p><b>Evidencia:</b></p>  <p style="text-align: center;"><i>Figura 51: Evidencia 7</i> Fuente: Autor</p>  <p style="text-align: center;"><i>Figura 52: Evidencia 8</i> Fuente: Autor</p>

*Tabla 23: Prueba 4*  
Fuente: Autor

<b>Pruebas</b>
<b>N° de Prueba: 005</b>
<b>Descripción:</b> Se hace la prueba del formulario para registro e ingreso al aplicativo
<b>Proceso:</b> Nos dirigimos al formulario de ingreso y sin escribir nada le vamos a dar ingresar.
<b>Resultado:</b> Al dar clic en ingresar muestra unos recuadros en rojo especificando que ingrese el nombre de usuario y la contraseña.

**Evidencia:**

Crear una Cuenta' and 'Quieres Regresar a la Pagina Principal? [Home](#)'."/>

**Ingresa con tu Cuenta**

Nombre de Usuario y Contraseña para Ingresar

Nombre de Usuario

Por Favor Ingrese el Nombre de Usuario!

Contraseña

Por Favor Ingrese la Contraseña!

Ingresar

No Tienes una Cuenta? [Crear una Cuenta](#)

Quieres Regresar a la Pagina Principal? [Home](#)

Figura 55: Evidencia 11

Fuente: Autor

Tabla 24: Prueba 5

Fuente: Autor

<b>Pruebas</b>
<b>N° de Prueba: 006</b>
<b>Descripción:</b> Se hace la prueba del formulario para registro e ingreso al aplicativo
<b>Proceso:</b> Nos dirigimos al formulario de ingreso y solo escribiendo el nombre de usuario.
<b>Resultado:</b> Al dar clic en ingresar muestra un recuadro en verde señalando que esta correcto al digitar el nombre de usuario y un recuadro en rojo especificando que ingrese la contraseña.

**Evidencia:**

**Ingresa con tu Cuenta**

Nombre de Usuario y Contraseña para Ingresar

Nombre de Usuario

Gustavo Walteros ✓

Contraseña

⚠

Por Favor Ingrese la Contraseña!

Ingresar

No Tienes una Cuenta? [Crear una Cuenta](#)  
Quieres Regresar a la Pagina Principal? [Home](#)

Figura 56: Evidencia 12  
Fuente: Autor

Tabla 25: Prueba 6  
Fuente: Autor

<b>Pruebas</b>
<b>N° de Prueba: 007</b>
<b>Descripción:</b> Se hace la prueba del formulario para registro e ingreso al aplicativo
<b>Proceso:</b> Nos dirigimos al formulario de ingreso y escribimos nombre de usuario y contraseña válidos para ingresar.
<b>Resultado:</b> Al llenar el formulario con el nombre de usuario y la contraseña esta se muestra oculta al momento de digitarla otorgando privacidad, después se da ingresar accediendo al aplicativo.

**Evidencia:**

## Ingresar con tu Cuenta

Nombre de Usuario y Contraseña para  
Ingresar

Nombre de Usuario

Gustavo Walteros ✓

Contraseña

••••| ✓

Ingresar

No Tienes una Cuenta? [Crear una Cuenta](#)

Quieres Regresar a la Pagina Principal? [Home](#)

*Figura 57: Evidencia 13*

*Fuente: Autor*



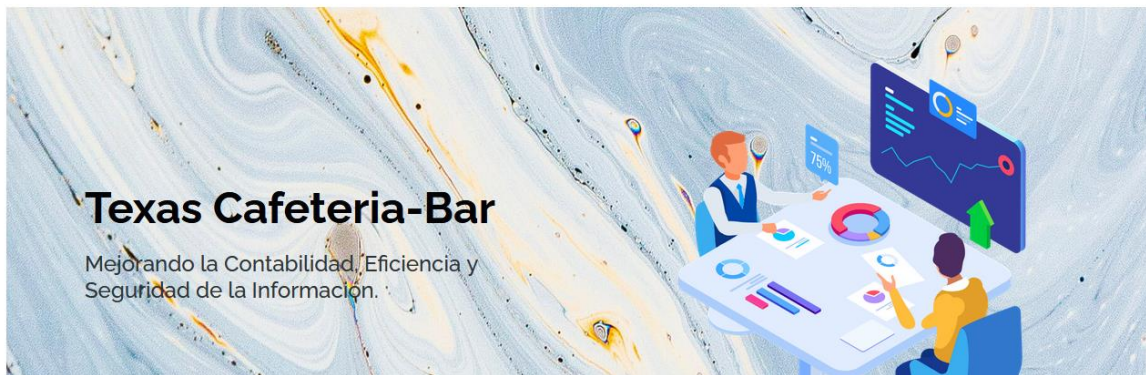
Bienvenido Gustavo Walteros !

[Home](#)

[Empresa](#)

[Recursos Humanos](#)

[logout](#)



*Figura 58: Evidencia 14*

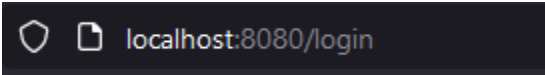
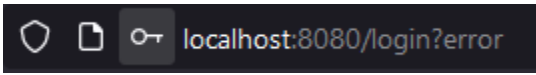
*Fuente: Autor*

*Tabla 26: Prueba 7*


*Fuente: Autor*

**Pruebas**

**N° de Prueba: 008**

<b>Descripción:</b> Se hace la prueba del formulario para registro e ingreso al aplicativo
<b>Proceso:</b> Nos dirigimos al formulario de ingreso y escribiendo nombre de usuario y contraseña no válidos para ingresar.
<b>Resultado:</b> Al llenar el formulario con el nombre de usuario y la contraseña que no son válidos después de dar clic en ingresar no arroja ninguna advertencia solo vuelve y redirige al mismo formulario para volverlo a llenar la única diferencia es el link de la página que cambia con el original.
<b>Evidencia:</b>  <p><i>Figura 59: Evidencia 15</i> <i>Fuente: Autor</i></p>  <p><i>Figura 60: Evidencia 16</i> <i>Fuente: Autor</i></p>

*Tabla 27: Prueba 8*  
*Fuente: Autor*

<b>Pruebas</b>
<b>N° de Prueba: 009</b>
<b>Descripción:</b> Se hace la prueba del formulario para registro e ingreso al aplicativo
<b>Proceso:</b> Nos dirigimos al finalizar el formulario en el apartado de crear cuenta y damos clic.
<b>Resultado:</b> Al dar clic en crear cuenta nos redirigimos al formulario de registro y creación de usuario para poder ingresar al aplicativo.
<b>Evidencia:</b>  <p><i>Figura 61: Evidencia 17</i> <i>Fuente: Autor</i></p>

**Crear Cuenta**

Ingrese su Información Personal para Crear una Cuenta

Cedula

Nombre de Usuario

Contraseña

Aceptar Terminos y Condiciones

**Crear Cuenta**

Quieres Regresar a la Pagina de Login? [Login](#)

Figura 62: Evidencia 18

Fuente: Autor

Tabla 28: Prueba 9

Fuente: Autor

Pruebas
<b>N° de Prueba: 010</b>
<b>Descripción:</b> Se hace la prueba del formulario para registro y crear usuario para ingresar al aplicativo
<b>Proceso:</b> Se procede a dejar vacíos los campos del formulario, después dar clic en crear cuenta.
<b>Resultado:</b> Al dar clic en crear cuenta se nos muestran los recuadros del formulario en rojo dando advertencia de que se deben llenar con la información pedida debajo de cada recuadro y después aceptando los términos y condiciones.

**Evidencia:**

Cedula

Por favor, Ingrese su Cedula!

Nombre de Usuario

Por Favor, Ingrese el Usuario.

Contraseña

Por Favor, Ingrese la Contraseña!

Aceptar Terminos y Condiciones  
Debe de estar de Acuerdo con los Terminos.

*Figura 63: Evidencia 19*  
*Fuente: Autor*

*Tabla 29: Prueba 10*  
*Fuente: Autor*

<b>Pruebas</b>
<b>N° de Prueba: 011</b>
<b>Descripción:</b> Se hace la prueba del formulario para registro y crear usuario para ingresar al aplicativo por medio de la cédula.
<b>Proceso:</b> Se procede a ingresar la cedula y dejar el resto de los campos vacíos del formulario, después dar clic en crear cuenta.
<b>Resultado:</b> Al dar clic en crear cuenta se nos muestran que la parte de cedula se marca en verde indicando de que se digitaron los datos pedidos y los recuadros del rojo dando advertencia de que se deben llenar con la información pedida debajo de cada recuadro y después aceptando los términos y condiciones.

**Evidencia:**

Cedula

1234567890 ✓

Nombre de Usuario

! Por Favor, Ingrese el Usuario.

Contraseña

! Por Favor, Ingrese la Contraseña!

Aceptar Terminos y Condiciones  
Debe de estar de Acuerdo con los Terminos.

*Figura 63: Evidencia 19*  
*Fuente: Autor*

*Tabla 30: Prueba 11*  
*Fuente: Autor*

Pruebas
<b>N° de Prueba: 012</b>
<b>Descripción:</b> Se hace la prueba del formulario para registro y crear usuario para ingresar al aplicativo por medio de la cédula y nombre.
<b>Proceso:</b> Se procede a ingresar la cedula, nombre de usuario y dejar el resto de los campos vacíos del formulario, después dar clic en crear cuenta.
<b>Resultado:</b> Al dar clic en crear cuenta se nos muestran que la parte de cedula y nombre de usuario se marca en verde indicando de que se digitaron los datos pedidos y los recuadros del rojo dando advertencia de que se deben llenar con la información pedida debajo de cada recuadro y después aceptando los términos y condiciones.

**Evidencia:**

Cedula

1234567890 ✓

Nombre de Usuario

Pepe ✓

Contraseña

ⓘ

Por Favor, Ingrese la Contraseña!

Aceptar Terminos y Condiciones  
Debe de estar de Acuerdo con los Terminos.

*Figura 64: Evidencia 20*  
*Fuente: Autor*

*Tabla 31: Prueba 12*  
*Fuente: Autor*

<b>Pruebas</b>
<b>N° de Prueba: 013</b>
<b>Descripción:</b> Se hace la prueba del formulario para registro y crear usuario para ingresar al aplicativo
<b>Proceso:</b> Se procede a ingresar la cedula, nombre de usuario, contraseña y no aceptar los términos, después dar clic en crear cuenta.
<b>Resultado:</b> Al dar clic en crear cuenta se nos muestran que la parte de cedula, nombre de usuario y contraseña se marca en verde indicando de que se digitaron los datos pedidos y se visualiza la parte de aceptar términos y condiciones en rojo indicando de que debemos aceptarlos para continuar.
<b>Evidencia:</b>

Cedula

1234567890 ✓

Nombre de Usuario

Pepe ✓

Contraseña

●●●●| ✓

Aceptar Terminos y Condiciones

Debe de estar de Acuerdo con los Terminos.

*Figura 65: Evidencia 21*  
*Fuente: Autor*

*Tabla 32: Prueba 13*  
*Fuente: Autor*

Pruebas
<b>N° de Prueba: 014</b>
<b>Descripción:</b> Se hace la prueba del formulario para registro y crear usuario para ingresar al aplicativo
<b>Proceso:</b> Se procede a ingresar los datos del formulario correctamente y aceptando los términos y condiciones, siguiendo con dar clic en crear cuenta.
<b>Resultado:</b> Al dar clic en crear cuenta se nos redirige a la página de inicio de sesión o Login para ingresar con el usuario existente en el aplicativo o ya creado previamente.
<b>Evidencia:</b>

Cedula

1234567890 ✓

Nombre de Usuario

Pepe ✓

Contraseña

•••• ✓

Aceptar Terminos y Condiciones

*Figura 66: Evidencia 22*  
*Fuente: Autor*

**Ingresa con tu Cuenta**  
Nombre de Usuario y Contraseña para Ingresar

Nombre de Usuario

Contraseña

Ingresar

No Tienes una Cuenta? [Crear una Cuenta](#)  
Quieres Regresar a la Pagina Principal? [Home](#)

*Figura 67: Evidencia 23*  
*Fuente: Autor*

*Tabla 33: Prueba 14*  
*Fuente: Autor*

<b>Pruebas</b>
<b>N° de Prueba: 015</b>
<b>Descripción:</b> Se hace la prueba del formulario para registro y crear usuario para ingresar al aplicativo
<b>Proceso:</b> Nos dirigimos a la parte final del formulario al apartado de ya tienes cuenta para darle clic al vinculo que tiene la palabra Login.
<b>Resultado:</b> Al dar clic nos redirige a la página de Login para digitar los datos ingresando así al aplicativo.

**Evidencia:**

Ya tienes Cuenta? [Login](#)

*Figura 68: Evidencia 24*

*Fuente: Autor*

The screenshot shows a login form with the following elements:

- Title: **Ingresa con tu Cuenta**
- Subtitle: Nombre de Usuario y Contraseña para Ingresar
- Input field: Nombre de Usuario
- Input field: Contraseña
- Button: Ingresar
- Links: No Tienes una Cuenta? [Crear una Cuenta](#); Quieres Regresar a la Pagina Principal? [Home](#)

*Figura 69: Evidencia 25*

*Fuente: Autor*

*Tabla 34: Prueba 15*

*Fuente: Autor*

**Pruebas**

**N° de Prueba: 016**

**Descripción:** Se hace prueba del aplicativo y todas las acciones disponibles

**Proceso:** Después de haber iniciado sesión con los datos validos nos dirigimos a la esquina derecha al apartado de recursos humanos.

**Resultado:** Al dar clic nos redirige a la página de recursos humanos donde se puede visualizar los usuarios existentes activos dentro del aplicativo.

**Evidencia:**

Recursos Humanos				
Nombre Usuario	Cedula	Estado	Acciones	
Gustavo Walteros	123456789	Habilitado	<a href="#">Editar</a>	<a href="#">Borrar</a>
Emilcen Walteros	098765432	Habilitado	<a href="#">Editar</a>	<a href="#">Borrar</a>
Lady Bermudez	87456321	Habilitado	<a href="#">Editar</a>	<a href="#">Borrar</a>
Pepe	123456	Habilitado	<a href="#">Editar</a>	<a href="#">Borrar</a>

*Figura 70: Evidencia 26*  
 Fuente: Autor

Tabla 35: Prueba 16  
 Fuente: Autor

Pruebas				
<b>N° de Prueba: 017</b>				
<b>Descripción:</b> Se hace prueba del aplicativo y todas las acciones disponibles				
<b>Proceso:</b> En la página de recursos humanos se va a dar clic en el botón de borrar.				
<b>Resultado:</b> Al dar clic se borra el registro del usuario seleccionado.				
<b>Evidencia:</b>				
Nombre Usuario	Cedula	Estado	Acciones	
Gustavo Walteros	123456789	Habilitado	<a href="#">Editar</a>	<a href="#">Borrar</a>
Emilcen Walteros	098765432	Habilitado	<a href="#">Editar</a>	<a href="#">Borrar</a>
Lady Bermudez	87456321	Habilitado	<a href="#">Editar</a>	<a href="#">Borrar</a>
Pepe	123456	Habilitado	<a href="#">Editar</a>	<a href="#">Borrar</a>

*Figura 71: Evidencia 27*  
 Fuente: Autor

Tabla 36: Prueba 17  
 Fuente: Autor

<b>Pruebas</b>
<b>N° de Prueba: 018</b>
<b>Descripción:</b> Se hace prueba del aplicativo y todas las acciones disponibles
<b>Proceso:</b> En la página de recursos humanos se va a dar clic en el botón de editar.
<b>Resultado:</b> Al dar clic nos redirige a la página donde se podrá modificar los datos del usuario, luego de guardar los datos modificados nos redirigiremos a recursos humanos visualizando todos los usuarios registrados.

**Evidencia:**

**Recursos Humanos**  
 Digite los Datos Pedidos en el Formulario

Cedula

Nombre de Usuario

*Figura 72: Evidencia 28*  
 Fuente: Autor

Nombre Usuario	Cedula	Estado	Acciones	
Gustavo Walteros	123456789	Habilitado	<input type="button" value="Editar"/>	<input type="button" value="Borrar"/>
Emilcen Walteros	098765432	Habilitado	<input type="button" value="Editar"/>	<input type="button" value="Borrar"/>
Lady Bermudez	87456321	Habilitado	<input type="button" value="Editar"/>	<input type="button" value="Borrar"/>
Pepes	123456	Habilitado	<input type="button" value="Editar"/>	<input type="button" value="Borrar"/>

*Figura 73: Evidencia 29*  
 Fuente: Autor

*Tabla 37: Prueba 18*  
 Fuente: Autor

<b>Pruebas</b>
<b>N° de Prueba: 019</b>
<b>Descripción:</b> Se hace prueba del aplicativo y todas las acciones disponibles
<b>Proceso:</b> Después de haber iniciado sesión con los datos validos nos dirigimos a la esquina derecha al apartado de distribuidores.
<b>Resultado:</b> Al dar clic nos redirige a la página de distribuidores donde se puede visualizar los distribuidores existentes activos dentro del aplicativo.

**Evidencia:**

Distribuidores		
Nombre	Estado	Acciones
Coca-Cola	Habilitado	<input type="button" value="Editar"/> <input type="button" value="Borrar"/>
Monster	Habilitado	<input type="button" value="Editar"/> <input type="button" value="Borrar"/>
Pepsi	Habilitado	<input type="button" value="Editar"/> <input type="button" value="Borrar"/>
Spartan	Habilitado	<input type="button" value="Editar"/> <input type="button" value="Borrar"/>

*Figura 74: Evidencia 30*  
 Fuente: Autor

Tabla 38: Prueba 19  
 Fuente: Autor

Pruebas																
<b>N° de Prueba: 020</b>																
<b>Descripción:</b> Se hace prueba del aplicativo y todas las acciones disponibles																
<b>Proceso:</b> En la página de distribuidores se va a dar clic en el botón de borrar.																
<b>Resultado:</b> Al dar clic se borra el registro del usuario seleccionado.																
<b>Evidencia:</b>																
<table border="1"> <thead> <tr> <th>Nombre</th> <th>Estado</th> <th>Acciones</th> </tr> </thead> <tbody> <tr> <td>Monster</td> <td>Habilitado</td> <td> <input type="button" value="Editar"/> <input type="button" value="Borrar"/> </td> </tr> <tr> <td>Pepsi</td> <td>Habilitado</td> <td> <input type="button" value="Editar"/> <input type="button" value="Borrar"/> </td> </tr> <tr> <td>Coca-Cola</td> <td>Habilitado</td> <td> <input type="button" value="Editar"/> <input type="button" value="Borrar"/> </td> </tr> <tr> <td>Monsters</td> <td>Habilitado</td> <td> <input type="button" value="Editar"/> <input type="button" value="Borrar"/> </td> </tr> </tbody> </table>	Nombre	Estado	Acciones	Monster	Habilitado	<input type="button" value="Editar"/> <input type="button" value="Borrar"/>	Pepsi	Habilitado	<input type="button" value="Editar"/> <input type="button" value="Borrar"/>	Coca-Cola	Habilitado	<input type="button" value="Editar"/> <input type="button" value="Borrar"/>	Monsters	Habilitado	<input type="button" value="Editar"/> <input type="button" value="Borrar"/>	
Nombre	Estado	Acciones														
Monster	Habilitado	<input type="button" value="Editar"/> <input type="button" value="Borrar"/>														
Pepsi	Habilitado	<input type="button" value="Editar"/> <input type="button" value="Borrar"/>														
Coca-Cola	Habilitado	<input type="button" value="Editar"/> <input type="button" value="Borrar"/>														
Monsters	Habilitado	<input type="button" value="Editar"/> <input type="button" value="Borrar"/>														

*Figura 75: Evidencia 31*  
 Fuente: Autor

Tabla 39: Prueba 20  
 Fuente: Autor

Pruebas	
<b>N° de Prueba: 021</b>	
<b>Descripción:</b> Se hace prueba del aplicativo y todas las acciones disponibles	
<b>Proceso:</b> En la página de distribuidores se va a dar clic en el botón de editar.	

**Resultado:** Al dar clic nos redirige a la página donde se podrá modificar los datos del distribuidor, luego de guardar los datos modificados nos redirigiremos a distribuidores visualizando todos los distribuidores registrados.

**Evidencia:**

Figura 76: Evidencia 32  
 Fuente: Autor

Nombre	Estado	Acciones	
Coca-Cola	Habilitado	<a href="#">Editar</a>	<a href="#">Borrar</a>
Monster	Habilitado	<a href="#">Editar</a>	<a href="#">Borrar</a>
Pepsi	Habilitado	<a href="#">Editar</a>	<a href="#">Borrar</a>
Spartan	Habilitado	<a href="#">Editar</a>	<a href="#">Borrar</a>

Figura 77: Evidencia 33  
 Fuente: Autor

Tabla 40: Prueba 21  
 Fuente: Autor

Pruebas
<b>N° de Prueba: 022</b>
<b>Descripción:</b> Se hace prueba del aplicativo y todas las acciones disponibles
<b>Proceso:</b> En la página de distribuidores se va a dar clic en el botón de crear distribuidor y guardar un distribuidor.
<b>Resultado:</b> Al dar clic nos redirige a la página donde se podrá crear un distribuidor si el recuadro para ingresar el distribuidor este vacío y le damos guardar este no nos dejara continuar hasta que digitemos el nombre de un distribuidor, al terminar, si los datos están bien se nos redirigirá a la sección de distribuidores donde se visualizara todos los distribuidores y su información.

**Evidencia:**

Nombre Distribuidor:

Ingrese el Nombre del Distribuidor !

Por Favor, Ingrese un Distribuidor!

Figura 78: Evidencia 34

Fuente: Autor

Nombre	Estado	Acciones
Coca-Cola	Habilitado	<span>Edit</span> <span>Borrar</span>
Monster	Habilitado	<span>Edit</span> <span>Borrar</span>
Pepsi	Habilitado	<span>Edit</span> <span>Borrar</span>
Spartan	Habilitado	<span>Edit</span> <span>Borrar</span>

Tabla 41: Prueba 22

Fuente: Autor

**Pruebas**

**N° de Prueba: 023**

**Descripción:** Se hace prueba del aplicativo y todas las acciones disponibles

**Proceso:** Después de haber iniciado sesión con los datos validos nos dirigimos a la esquina derecha al apartado de productos.

**Resultado:** Al dar clic nos redirige a la página de productos donde se puede visualizar los productos existentes activos dentro del aplicativo.

**Evidencia:**

Nombre	Distribuidor	Cantidad Unitaria	Precio Unitario	Empaquetado	Cantidad Al Mayor	Precio Al Mayor	Acciones
Ultra	Monster	12	8000	Caja	1	40000	<span>Edit</span> <span>Borrar</span>
Coca-Cola	Coca-Cola	32	3000	Canasta	1	600000	<span>Edit</span> <span>Borrar</span>
Pepsi	Pepsi	64	3000	Canasta	2	80000	<span>Edit</span> <span>Borrar</span>
Blanco	Monster	6	8000	Caja	1	50000	<span>Edit</span> <span>Borrar</span>

Figura 78: Evidencia 34

Fuente: Autor

Tabla 42: Prueba 23

Fuente: Autor

**Pruebas**

**N° de Prueba: 024**

**Descripción:** Se hace prueba del aplicativo y todas las acciones disponibles

**Proceso:** En la página de productos se va a dar clic en el botón de borrar.

**Resultado:** Al dar clic se borra el registro del producto seleccionado.

**Evidencia:**

Nombre	Distribuidor	Cantidad Unitaria	Precio Unitario	Empaquetado	Cantidad Al Mayor	Precio Al Mayor	Acciones
Pepsi	Pepsi	64	3000	Canasta	2	80000	<a href="#">Editar</a> <a href="#">Borrar</a>
Coca-Cola	Coca-Cola	32	2000	Canasta	1	60000	<a href="#">Editar</a> <a href="#">Borrar</a>
Ultraviolets	Monster	12	8000	Caja	1	40000	<a href="#">Editar</a> <a href="#">Borrar</a>

Figura 79: Evidencia 35

Fuente: Autor

Tabla 43: Prueba 24

Fuente: Autor

### Pruebas

**N° de Prueba: 025**

**Descripción:** Se hace prueba del aplicativo y todas las acciones disponibles

**Proceso:** En la página de distribuidores se va a dar clic en el botón de editar.

**Resultado:** Al dar clic nos redirige a la página donde se podrá modificar los datos del producto, luego de guardar los datos modificados nos redirigiremos a productos visualizando todos los productos registrados.

**Evidencia:**

### Texas Cafeteria-Bar

#### Ingresa los Datos de los Productos

Nombre Distribuidor:

Nombre Producto:

Cantidad Unitaria:

Precio Unitario:

Tipo Empaquetado:

Cantidad al Mayor:

Precio al Mayor:

[Guardar](#)

[Quieres Regresar? Volver](#)

Figura 80: Evidencia 36

Fuente: Autor

Nombre	Distribuidor	Cantidad Unitaria	Precio Unitario	Empaquetado	Cantidad Al Mayor	Precio Al Mayor	Acciones
Pepsi	Pepsi	64	3000	Canasta	2	80000	<a href="#" style="background-color: #17a2b8; color: white; padding: 2px 5px;">Editar</a> <a href="#" style="background-color: #dc3545; color: white; padding: 2px 5px;">Borrar</a>
Coca-Cola	Coca-Cola	32	2000	Canasta	1	60000	<a href="#" style="background-color: #17a2b8; color: white; padding: 2px 5px;">Editar</a> <a href="#" style="background-color: #dc3545; color: white; padding: 2px 5px;">Borrar</a>
Ultra	Monster	12	8000	Caja	1	40000	<a href="#" style="background-color: #17a2b8; color: white; padding: 2px 5px;">Editar</a> <a href="#" style="background-color: #dc3545; color: white; padding: 2px 5px;">Borrar</a>

Figura 81: Evidencia 37

Fuente: Autor

Tabla 44: Prueba 25

Fuente: Autor

<b>Pruebas</b>
<b>N° de Prueba: 026</b>
<b>Descripción:</b> Se hace prueba del aplicativo y todas las acciones disponibles
<b>Proceso:</b> En la página de productos se va a dar clic en el botón de crear distribuidor y guardar un distribuidor.
<b>Resultado:</b> Al dar clic nos redirige a la página donde se podrá crear un producto si los recuadros para ingresar el producto está vacío y le damos guardar este no nos dejara continuar hasta que digitemos el nombre del producto e ingresando los demás datos en los campos, los distribuidores como son seleccionables el software los cuenta como si tuvieran datos dentro pero este no existe si guardamos por defecto algún dato seleccionable como empaquetado o distribuidor el software se dañara porque esta seleccionando un dato que no está o existe procurar siempre seleccionar algo, al terminar, si los datos están bien se nos redirigirá a la sección de productos donde se visualizara todos los productos y su información.
<b>Evidencia:</b>

Nombre Distribuidor:  
 ✓

Nombre Producto:  
 ⓘ  
 Por Favor, Ingrese un Producto!

Cantidad Unitaria:  
 ⓘ  
 Por Favor, Ingrese una Cantidad!

Precio Unitario:  
 ⓘ  
 Por Favor, Ingrese un Precio!

Tipo Empaquetado:  
 ✓

Cantidad al Mayor:  
 ⓘ  
 Por Favor, Ingrese una Cantidad!

Precio al Mayor:  
 ⓘ  
 Por Favor, Ingrese un Precio!

Figura 82: Evidencia 38  
 Fuente: Autor

Nombre	Distribuidor	Cantidad Unitaria	Precio Unitario	Empaquetado	Cantidad Al Mayor	Precio Al Mayor	Acciones
Pepsi	Pepsi	64	3000	Canasta	2	80000	<a href="#">Editar</a> <a href="#">Borrar</a>
Coca-Cola	Coca-Cola	32	2000	Canasta	1	60000	<a href="#">Editar</a> <a href="#">Borrar</a>
Ultra	Monster	12	8000	Caja	1	40000	<a href="#">Editar</a> <a href="#">Borrar</a>

Tabla 45: Prueba 26  
 Fuente: Autor


Pruebas
<b>N° de Prueba: 027</b>
<b>Descripción:</b> Se hace prueba del aplicativo y todas las acciones disponibles
<b>Proceso:</b> En la página de productos se va a dar clic en el botón de logout para cerrar la sesión actual.
<b>Resultado:</b> Al dar clic nos redirige a la página Login donde se podrá volver a iniciar sesión con cualquier usuario valido registrado en el aplicativo.
<b>Evidencia:</b>  <p>Figura 83: Evidencia 39 Fuente: Autor</p> <p>Figura 84: Evidencia 40 Fuente: Autor</p>

Tabla 46: Prueba 27  
Fuente: Autor

- SonarQube

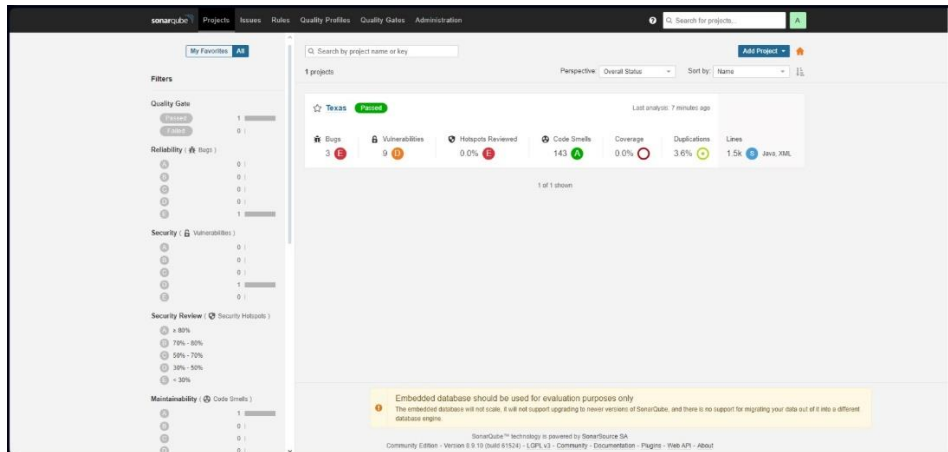


Figura 85: SonarQube  
 Fuente: Autor

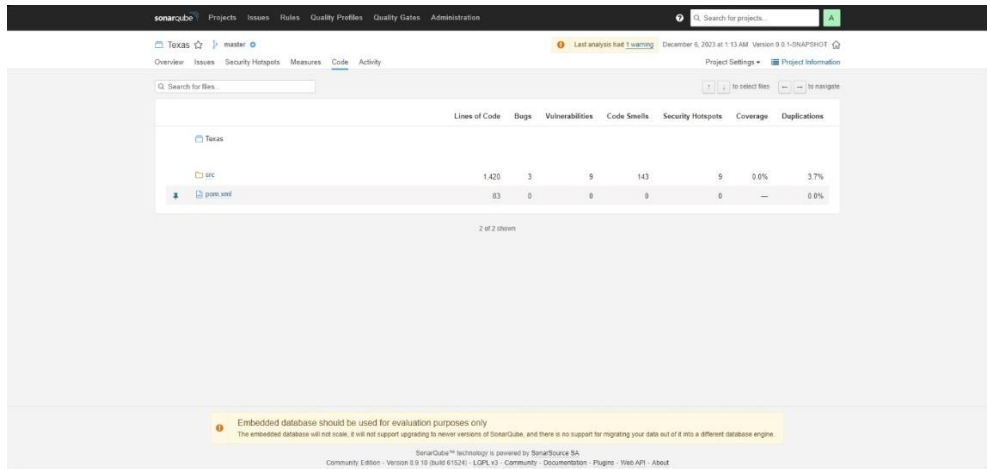


Figura 86: SonarQube  
 Fuente: Autor

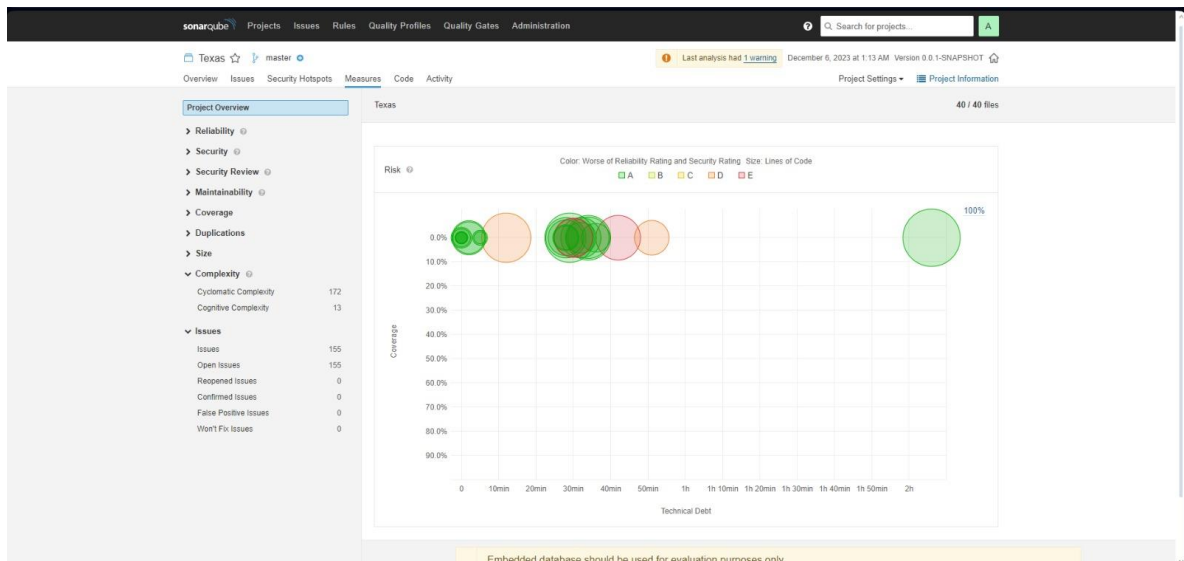


Figura 87: SonarQube  
Fuente: Autor

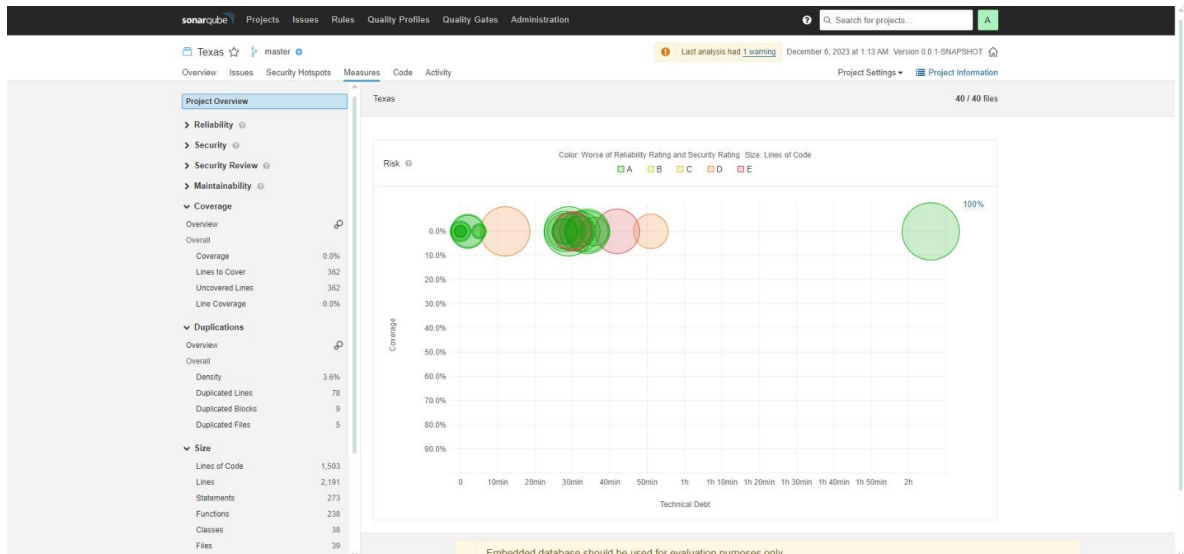


Figura 88: SonarQube  
Fuente: Autor

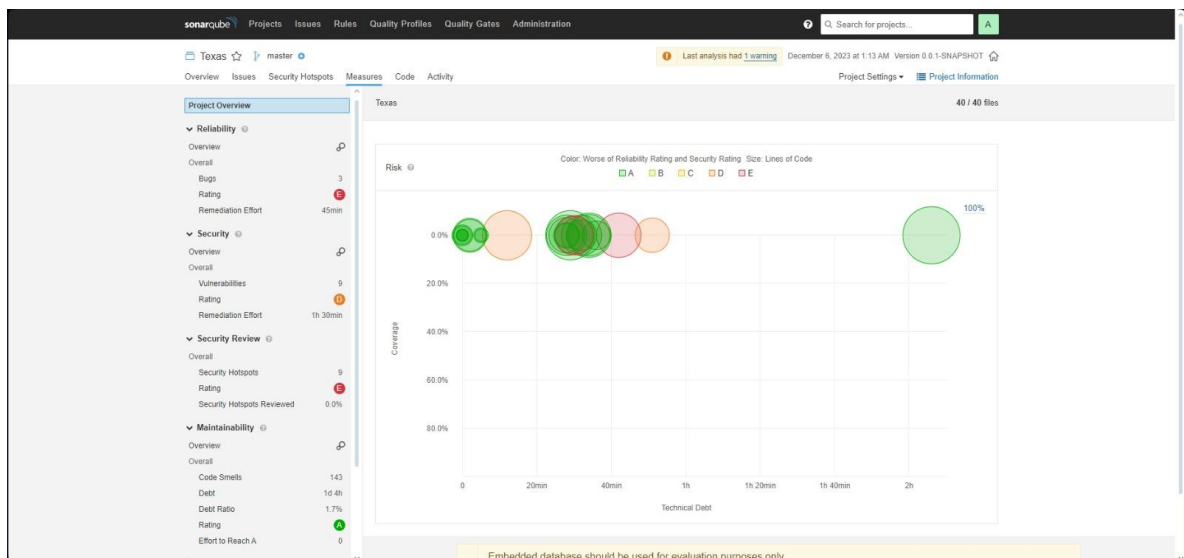


Figura 89: SonarQube

Fuente: Autor

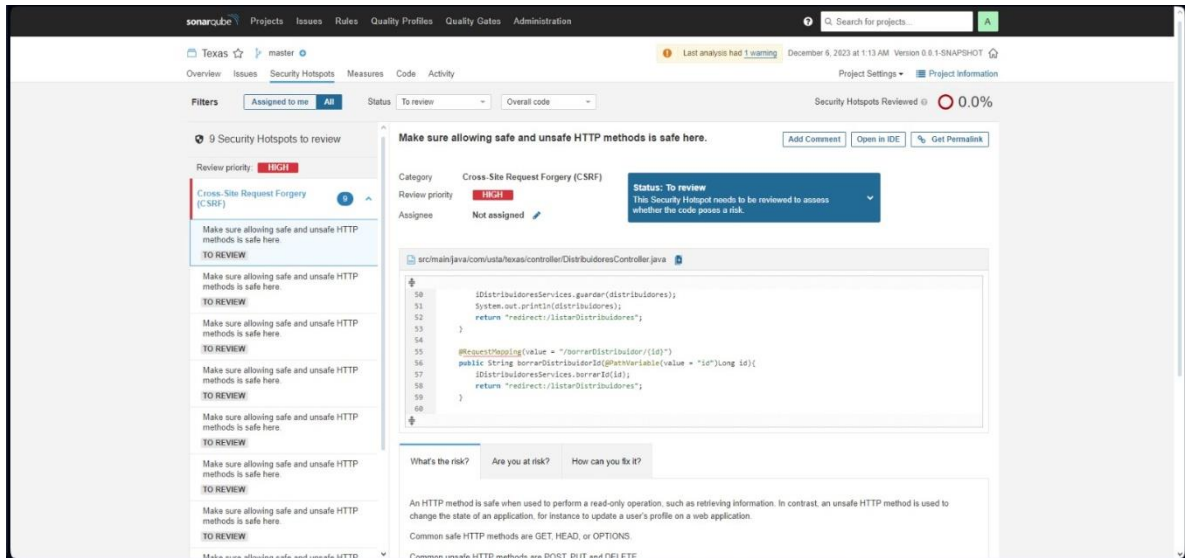


Figura 90: SonarQube

Fuente: Autor

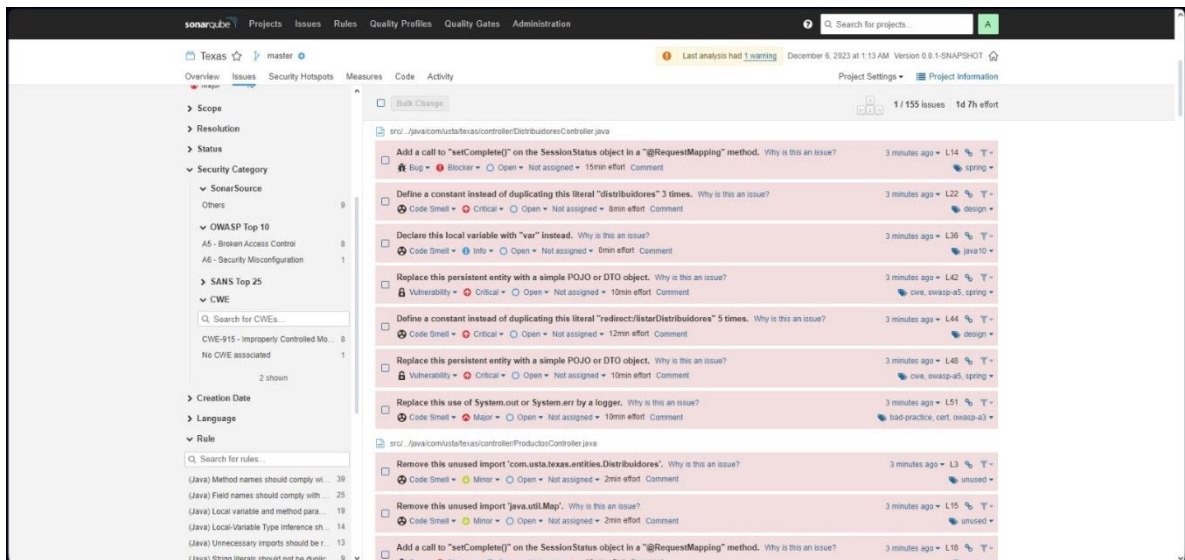


Figura 90: SonarQube

Fuente: Autor

The screenshot displays the SonarQube interface for a project named 'Texas'. The top navigation bar includes 'sonarqube', 'Projects', 'Issues', 'Rules', 'Quality Profiles', 'Quality Gates', and 'Administration'. A search bar is present on the right. The main content area shows a list of issues under the 'Issues' tab. On the left, there is a 'Filters' sidebar with categories like 'Type' (Bug, Vulnerability, Code Smell), 'Severity' (Blocker, Critical, Major, Info, Minor), 'Scope', 'Resolution', 'Status', and 'Security Category'. The issue list includes items such as 'Add the @Override annotation above this method signature', 'Remove this use of "WebSecurityConfigurerAdapter": it is deprecated', 'Define a constant instead of duplicating this literal "SUPERADMIN" 8 times', and 'Rename this method name to match the regular expression "[a-zA-Z0-9]\*"'. Each issue entry shows its severity, effort, and a comment.

Figura 91: SonarQube

Fuente: Autor

The screenshot shows the 'Measures' section of the SonarQube interface. On the left, a green box indicates 'Passed' with the note 'All conditions passed'. The main area displays various quality metrics: 'New Code' (3 Bugs), 'Overall Code' (9 Vulnerabilities), 'Security Hotspots' (9, 0.0% Reviewed), 'Ded' (1d 4h), 'Code Smells' (143), 'Coverage on 362 Lines to cover' (0.0%), 'Unit Tests', 'Duplications on 1.5K Lines' (3.6%), and 'Duplicated Blocks' (9). Each metric is accompanied by a circular progress indicator and a color-coded status.

Figura 92: SonarQube

Fuente: Autor

## 7. CONCLUSIONES Y ANÁLISIS DE RESULTADOS

- La creación del documento de requisitos brinda una comunicación directa con el cliente o la empresa, permitiendo entender mejor las necesidades de la entidad. Se utilizó la metodología ágil Extreme Programming XP, desarrollando las historias de usuario, lo cual brindó facilidad para entender a los trabajadores de la compañía. Al realizar las tareas de ingeniería se logró obtener el conocimiento necesario para iniciar el diseño del software.
- MVC (Modelo, Vista, Controlador) como patrón de arquitectura de software facilitó el desarrollo de la aplicación por medio de capas. Así mismo, con el uso del framework Spring, se dio soporte a los procesos CRUD, de autenticación y manejo de roles. El lenguaje HTML permitió estructurar las interfaces de usuario, desplegando todos los contenidos que se proveerán y la biblioteca Bootstrap para la construcción del sistema de forma rápida y sencilla, proporcionando componentes y plantillas CSS y HTML.
- Se realizaron pruebas que permitieran descubrir posibles fallos o problemas que se presentaran en el aplicativo. Al finalizar se obtuvo un rendimiento satisfactorio del sistema logrando implementar las funcionalidades propuestas en los requerimientos.
- El software brinda a la empresa una herramienta tecnológica que automatice el proceso de gestión y administración de la información, permitiendo así, la transición de estrategias tradicionales hacia la implementación de soluciones informáticas, con el fin de optimizar los procesos.

## 8. REFERENCIAS

- Andalucía, J. d. (s.f.). *Qué es y para qué sirve Hibernate*. Obtenido de Qué es y para qué sirve Hibernate: <https://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/97#:~:text=Hibernate%20es%20una%20capa%20de,ser%20usado%20en%20un%20cluster.>
- Andalucía, J. (s.f.). *Que es y para que sirve JPA*. Obtenido de Que es y para que sirve JPA: <https://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/176#:~:text=JPA%20utiliza%20anotaciones%20de%20metadatos,soporta%20la%20herencia%20y%20polimorfismo.>
- Consuelo, B. O. (s.f.). *LAS TECNOLOGÍAS DE LA INFORMACIÓN Y COMUNICACIÓN (T.I.C.)*. Obtenido de LAS TECNOLOGÍAS DE LA INFORMACIÓN Y COMUNICACIÓN (T.I.C.): <https://www.uv.es/~bellochc/pdf/pwtic1.pdf>
- DeveloPrograming. (2 de Noviembre de 2021). *Que Es y Para Que Sirve IntelliJ Idea*. Obtenido de Que Es y Para Que Sirve IntelliJ Idea: <https://developrograming.com/que-es-intellij-idea/#:~:text=Es%20un%20entorno%20de%20desarrollo,fué%20en%20diciembre%20de%202019.>
- Manuel, H. J. (s.f.). *Evolución de las comunicaciones*. Obtenido de Evolución de las comunicaciones: [https://www.acta.es/medios/articulos/comunicacion\\_e\\_informacion/039009.pdf](https://www.acta.es/medios/articulos/comunicacion_e_informacion/039009.pdf)
- OvHcloud. (2023). *Que Es Postgres, Usos Y Soluciones*. Obtenido de Que Es Postgres, Usos Y Soluciones: <https://www.ovhcloud.com/es/lp/postgresql-definition/#:~:text=PostgreSQL%20es%20un%20sistema%20o,a%20objetos%20%20potentes%20y%20escalables.>
- Pressman, R. (2010). *Ingeniería del Software. Un enfoque práctico*. USA: McGraw-Hill.
- Ramón, R. (2020). *EVOLUCIÓN, HISTORIA Y COMUNICACIÓN EN UN MUNDO DIGITAL (DISCURSO HIPOTÉTICO)*. Sevilla: Anthrosos. Obtenido de EVOLUCIÓN, HISTORIA Y COMUNICACIÓN EN UN MUNDO DIGITAL (DISCURSO HIPOTÉTICO).
- Santander, B. (2023). *Que es Bootstrap*. Obtenido de Que es Bootstrap: <https://www.becas-santander.com/es/blog/que-es-bootstrap.html>

- School, T. (2023). *Que Es Y Para Que Sirve Spring Framework*. Obtenido de Que Es Y Para Que Sirve Spring Framework: <https://www.tokioschool.com/noticias/que-es-spring-framework/#:~:text=Spring%20es%20un%20framework%20de,empleando%20objetos%20de%20java%20sencillos>.
- SINTYA MILENA MELÉNDEZ VALLADAREZ, M. E. (28 de Enero de 2016). *METODOLOGIA ÁGIL DE DESARROLLO DE SOFTWARE PROGRAMACION EXTREMA*. Obtenido de METODOLOGIA ÁGIL DE DESARROLLO DE SOFTWARE PROGRAMACION EXTREMA.: <https://repositorio.unan.edu.ni/1365/1/62161.pdf>
- Vadavo. (2023). *Que Es y Para Que Sirve HTML*. Obtenido de Que Es y Para Que Sirve HTML: <https://www.vadavo.com/blog/html-que-es-y-para-que-sirve/>

## 9. ANEXOS

### 9.1. Manual de Usuario

#### 9.1.1. Navegación Página Principal

Empezaremos situándonos desde la página principal del aplicativo, navegando por las distintas acciones.



Figura 93: Barra de Navegación

Fuente: Autor

Ya estando situados en la barra de navegación se procede a dar en conocimiento las siguientes pestañas:

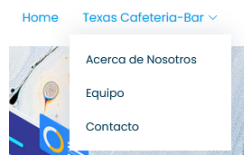


Figura 94: Desplegable de Navegación

Fuente: Autor

Las pestañas de acerca de nosotros, dando una pequeña descripción del servicio que brinda la entidad.

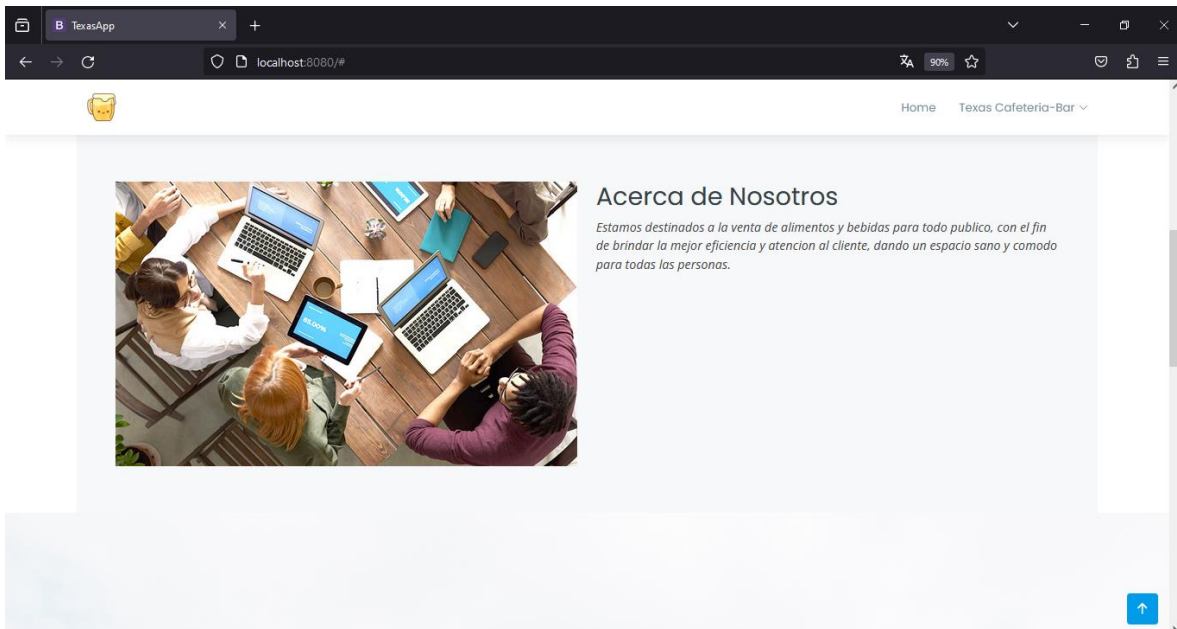


Figura 95: Acerca de Nosotros

Fuente: Autor

La pestaña de equipo, se visualiza al integrante que desarrollo el aplicativo.

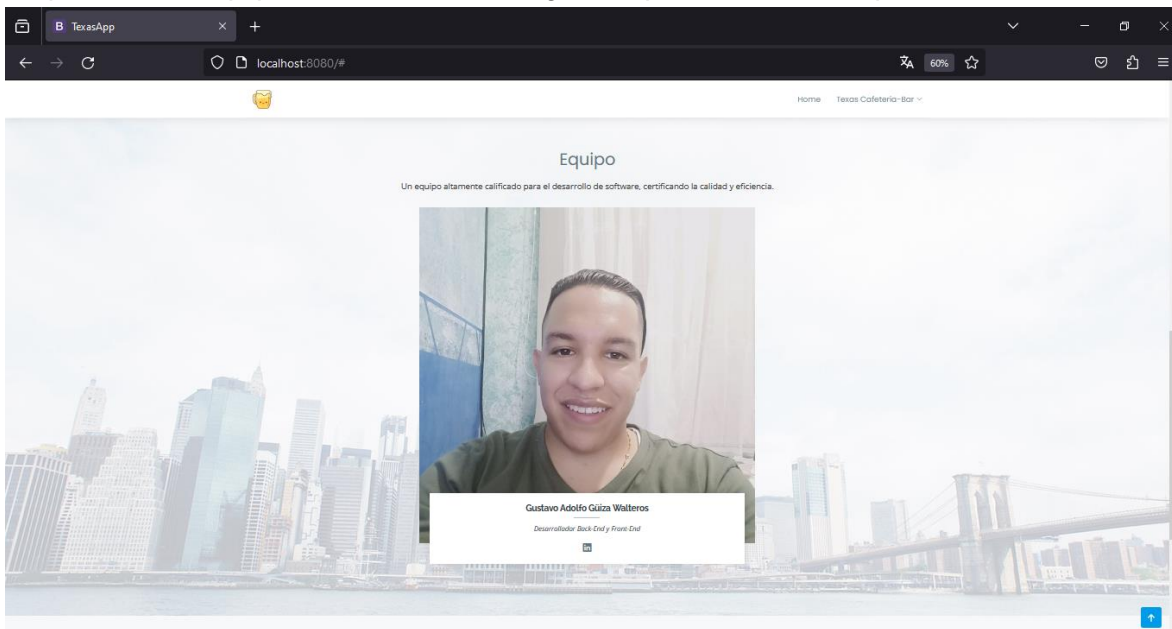


Figura 96: Equipo

Fuente: Autor

La pestaña de contactos, brindando la información que ayuda a la entidad a contactar con el integrante que desarrollo el sistema en caso de una falla.

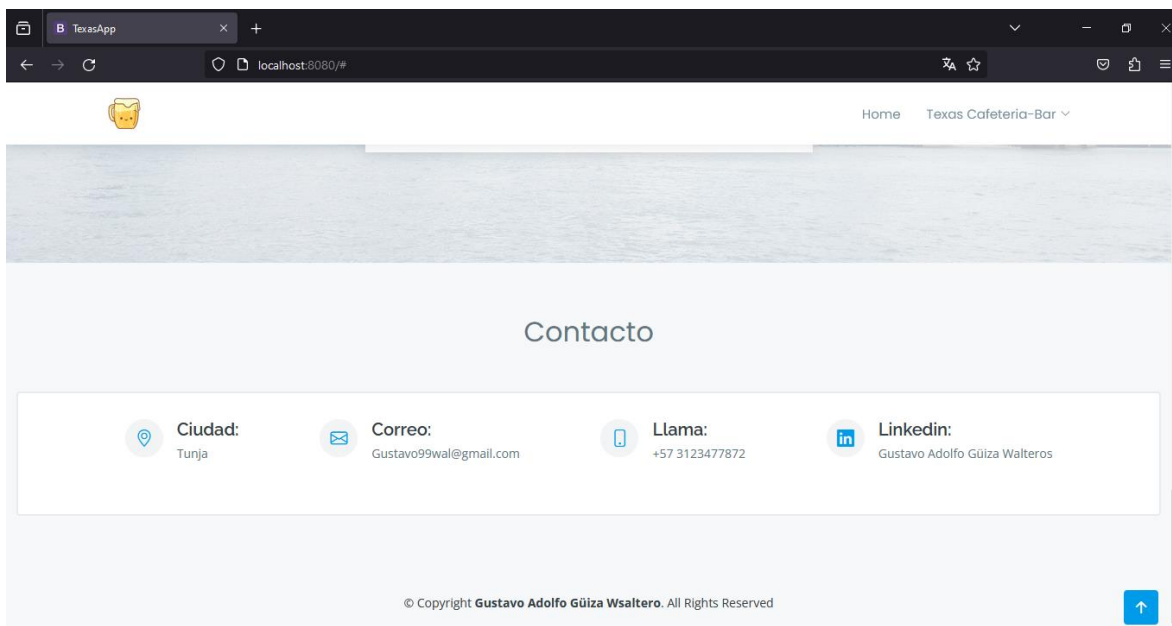


Figura 97: Contacto

Fuente: Autor

Y volvemos a la parte de arriba para ingresar al aplicativo dando clic en la barra de navegacion en la seccion Home.

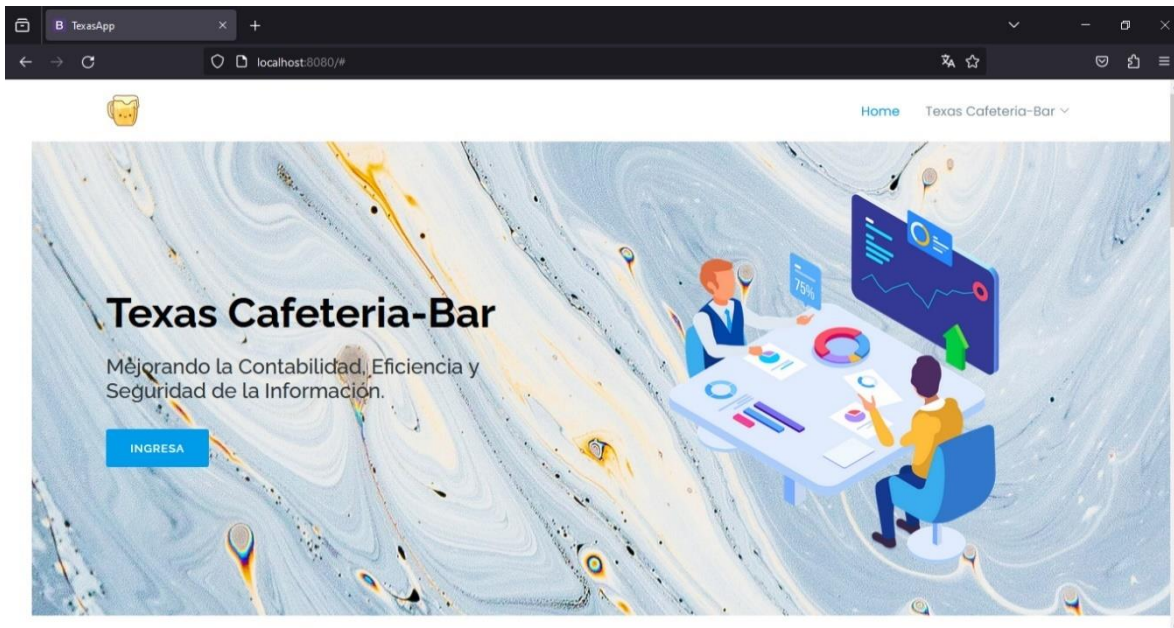
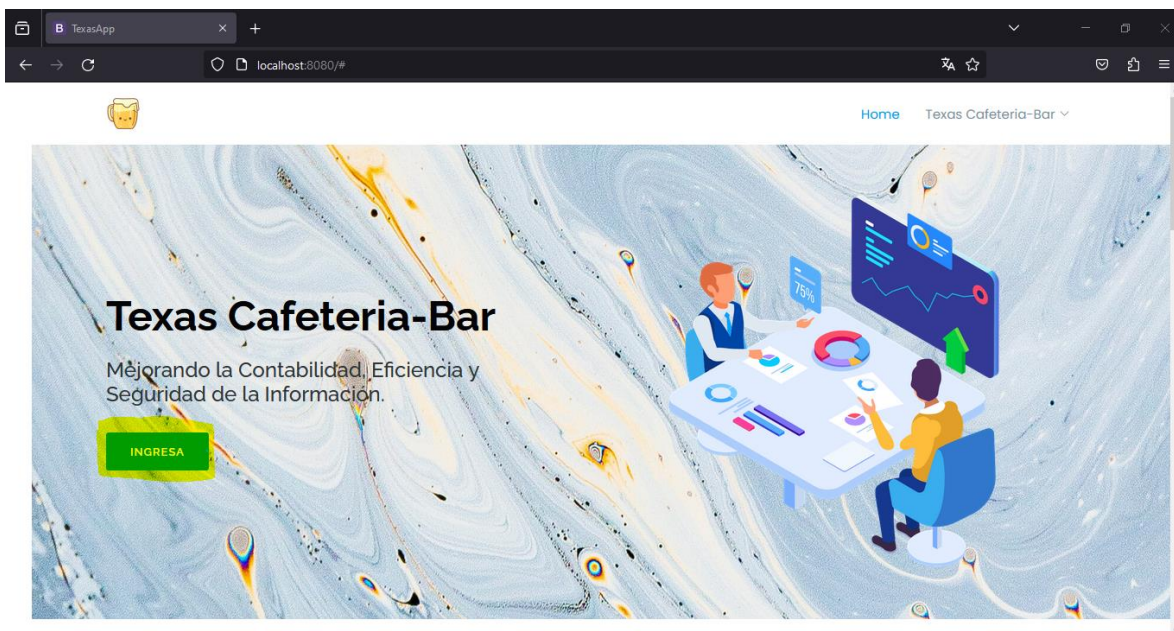
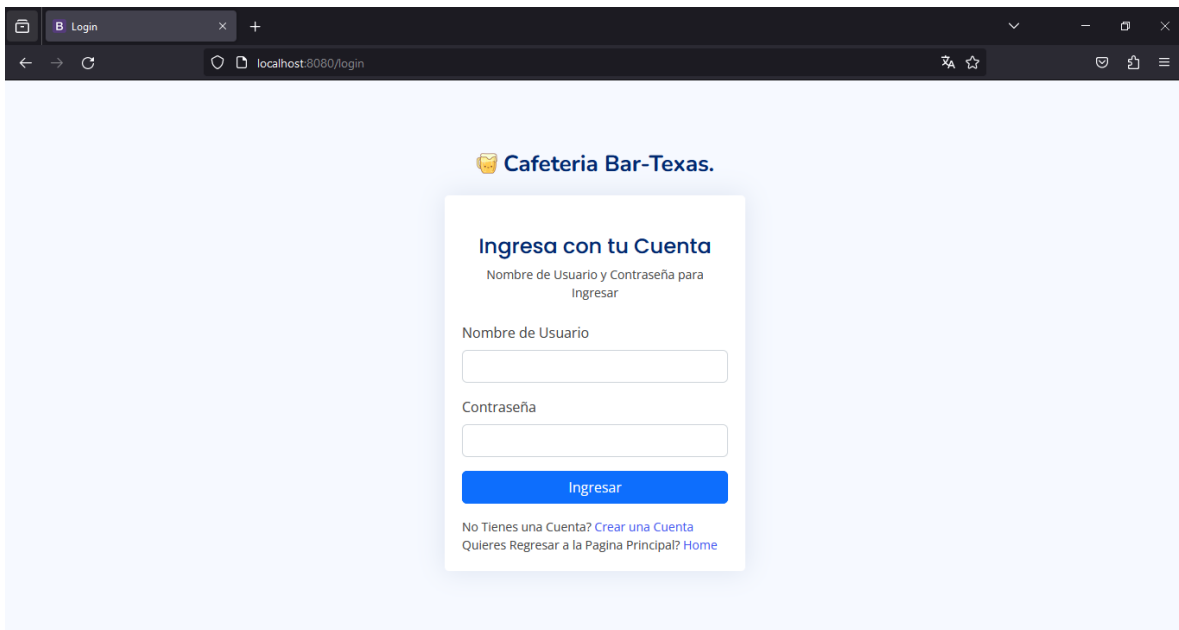


Figura 98: Home  
Fuente: Autor

### 9.1.2. Ingreso y Registro

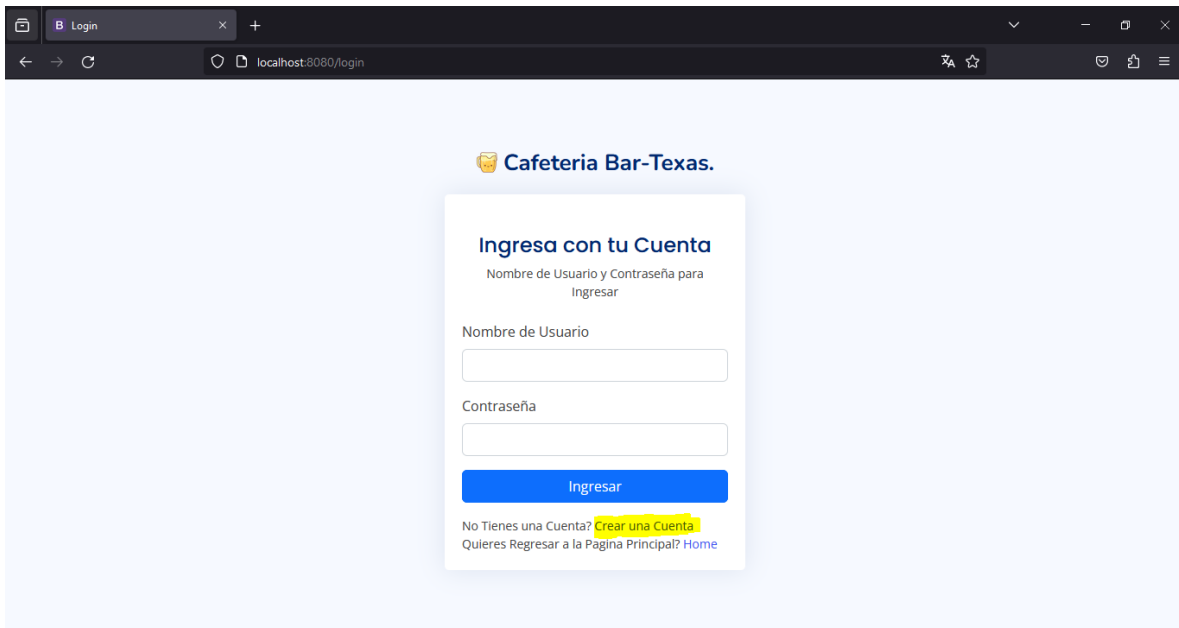
En la sección Home de la página principal habrá un apartado con un botón que dice ingresar resaltado de color amarillo, dando clic se nos redirigirá a la página Login.

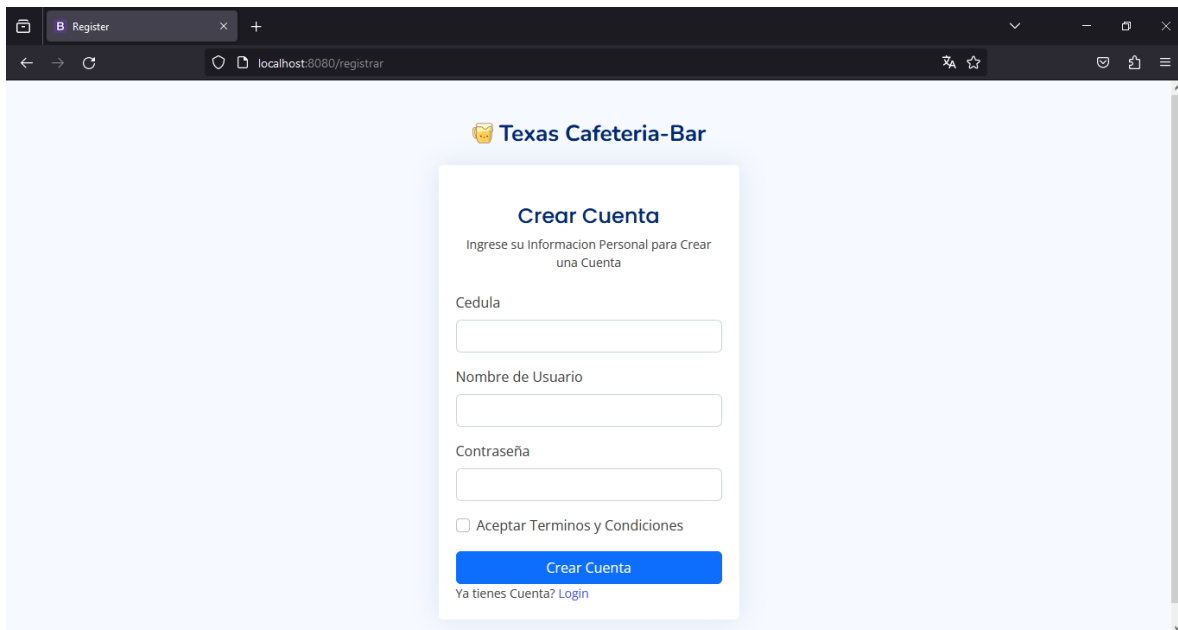




*Figura 99: Login*  
*Fuente: Autor*

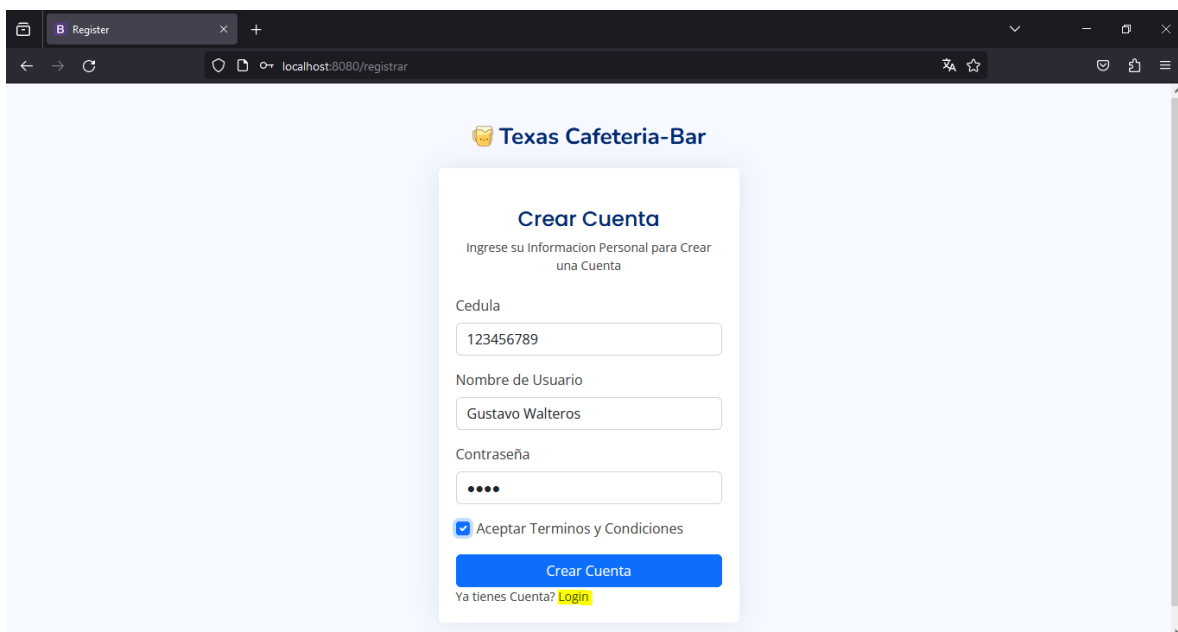
Ya situado en la página del Login previamente se debe registrar un usuario para ingresar al aplicativo, iremos a la parte final del formulario debajo del botón ingresar dice “¿No tienes Cuenta?” al lado habrá una palabra que dice Crear una Cuenta con un hipervínculo señalada de color amarillo, daremos clic.



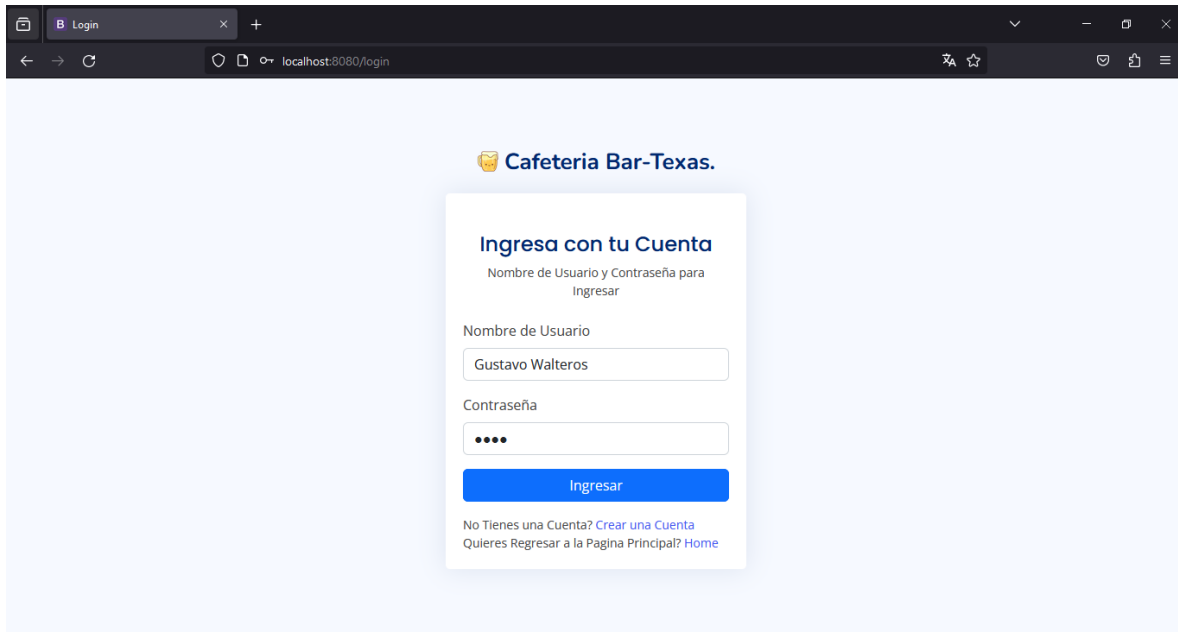


*Figura 100: Registro*  
*Fuente: Autor*

Estando en la página de registro se procede a digitar los campos pedidos por el formulario para el registro de usuario y aceptando los términos y condiciones, después de ser creado se nos redirigirá a la página del Login para iniciar sesión y acceder al aplicativo, en dado caso que ya allá una cuenta solo es darle clic en el apartado señalado de color amarillo al lado de la frase “¿Ya tienes Cuenta?” para regresar a la página de inicio de sesión e ingresar el usuario previamente creado.



Estando en la pagina de login se ingresan los datos validos ya creados, si el nombre de usuario tiene la primera en mayuscula escribirla de la misma manera, si no, desconocera el usuario, ya estando confirmados los datos digitados se accedera al aplicativo para consumir todas sus funciones.



### 9.1.3. Aplicativo Sistema de Inventario.

Después de haber ingresado los datos validos en el formulario de inicio de sesión, se nos redirigirá al menú principal del aplicativo del sistema, visualizaremos una barra de navegación señalada de color amarillo y al lado un botón de logout, dando clic en este botón se nos cerrará la sesión redirigiendo a la página Login para volver a iniciar sesión.

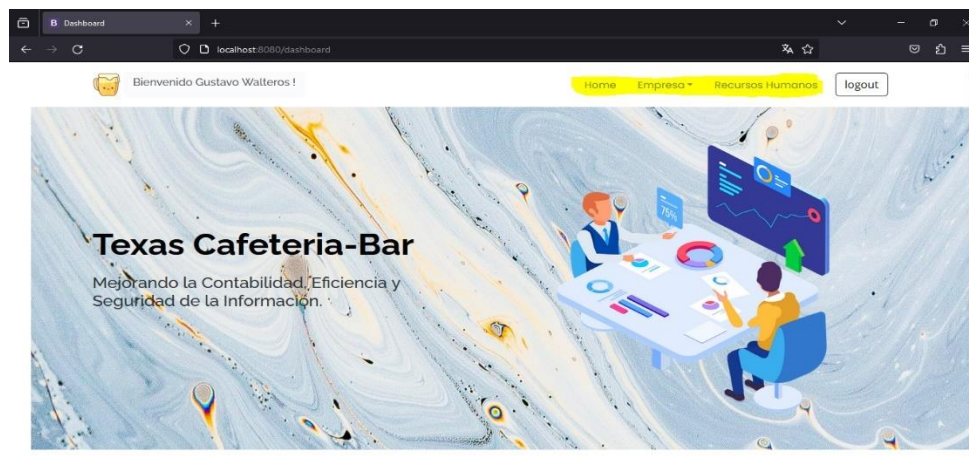


Figura 101: Página Principal Aplicativo  
Fuente: Autor

En la barra de navegación iremos al apartado de Recursos Humanos para visualizar la información de la página.

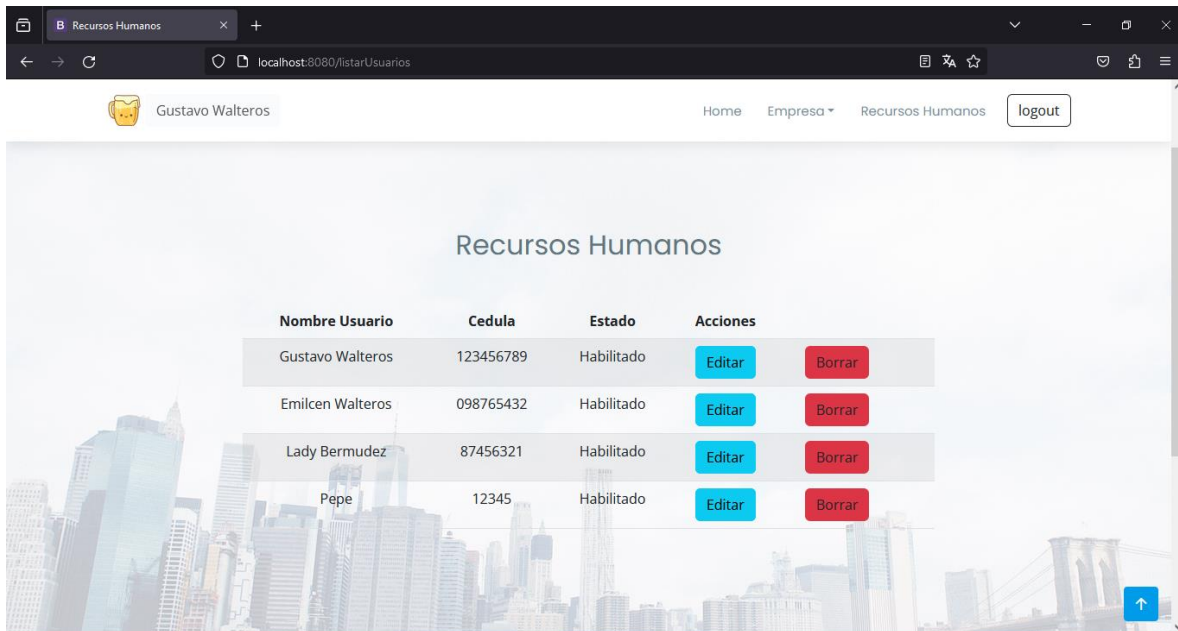


Figura 102: Recursos Humanos

Fuente: Autor

Anteriormente se visualiza la información de los usuarios creados para acceder al aplicativo, esta sección solo se mostrará para los usuarios que tengan el rol administrador, si algún empleado registrado intenta acceder a la sección le aparecerá que esta no existe.

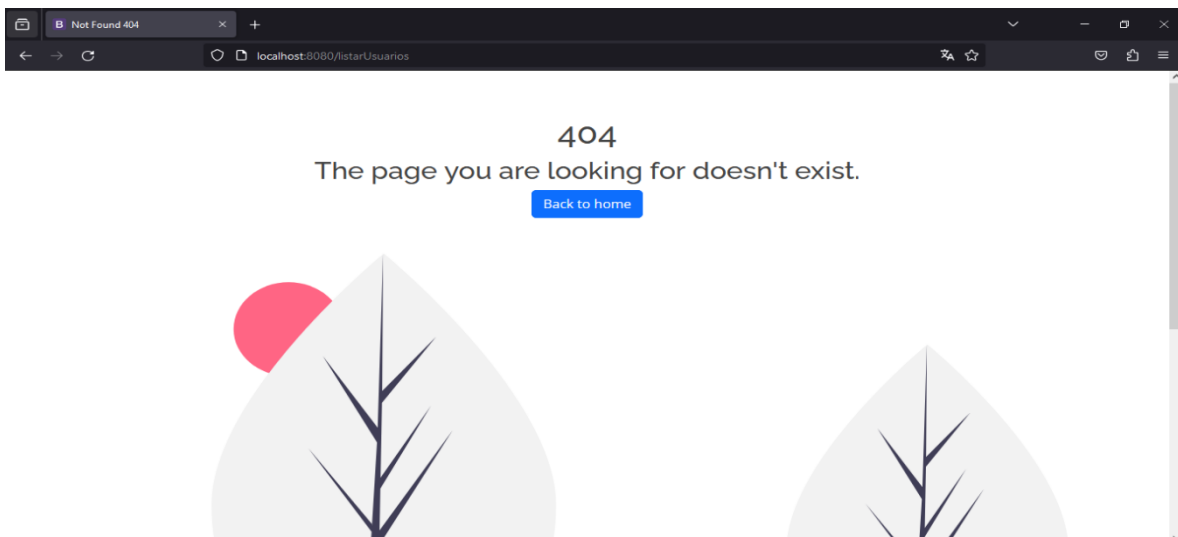


Figura 103: Not Found

Fuente: Autor

En la pagina de recursos humanos podemos interactuar con los botones de acción uno nombrado editar y otro borrar, el botón de editar es para modificar o cambiar los datos ya registrados del usuario existente dentro del sistema, dándole clic se nos redirigirá a otra página para el reemplazo de la información ya digitada y el botón de borrar elimina los datos permanentes seleccionados del sistema.



**Recursos Humanos**  
Digite los Datos Pedidos en el Formulario

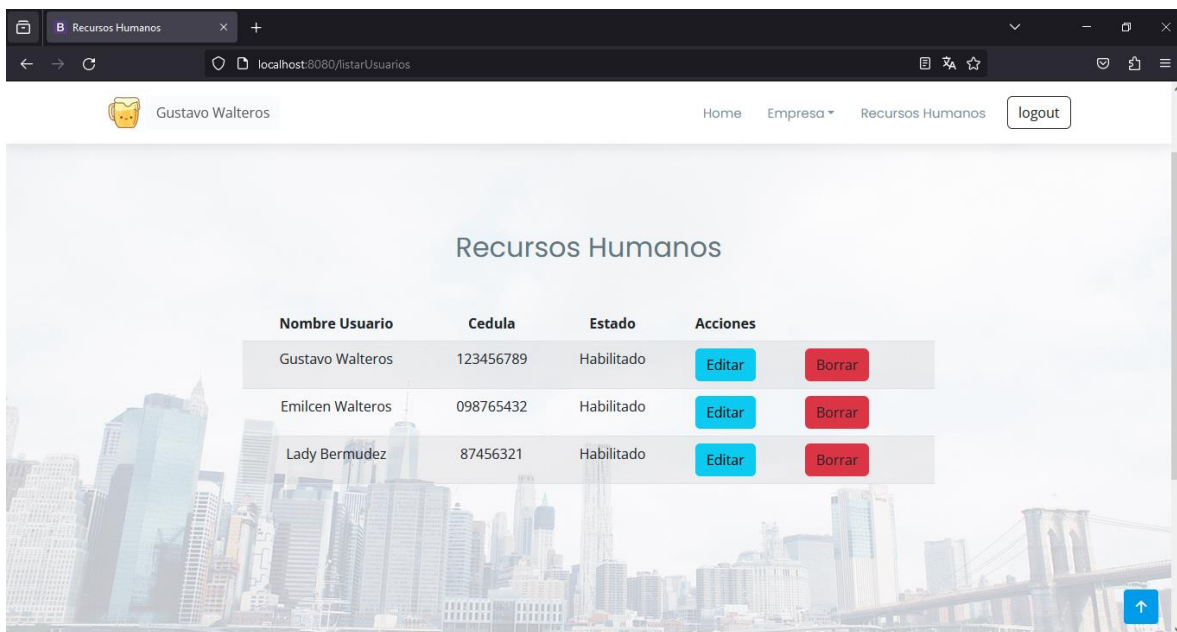
Cedula  
123456

Nombre de Usuario  
Pepes

Guardar

Figura 104: Editar Recursos Humanos

Fuente: Autor



Gustavo Walteros Home Empresa Recursos Humanos logout

### Recursos Humanos

Nombre Usuario	Cedula	Estado	Acciones	
Gustavo Walteros	123456789	Habilitado	Editar	Borrar
Emilcen Walteros	098765432	Habilitado	Editar	Borrar
Lady Bermudez	87456321	Habilitado	Editar	Borrar

Figura 105: Borrar Recursos Humanos

Fuente: Autor

Desplazándonos por la barra de navegación nos dirigiremos a la sección que dice empresa y se desplegara un menú daremos clic en distribuidores, se visualizara la información de los distribuidores existentes dentro del sistema botones de acción y un botón para crear un distribuidor.

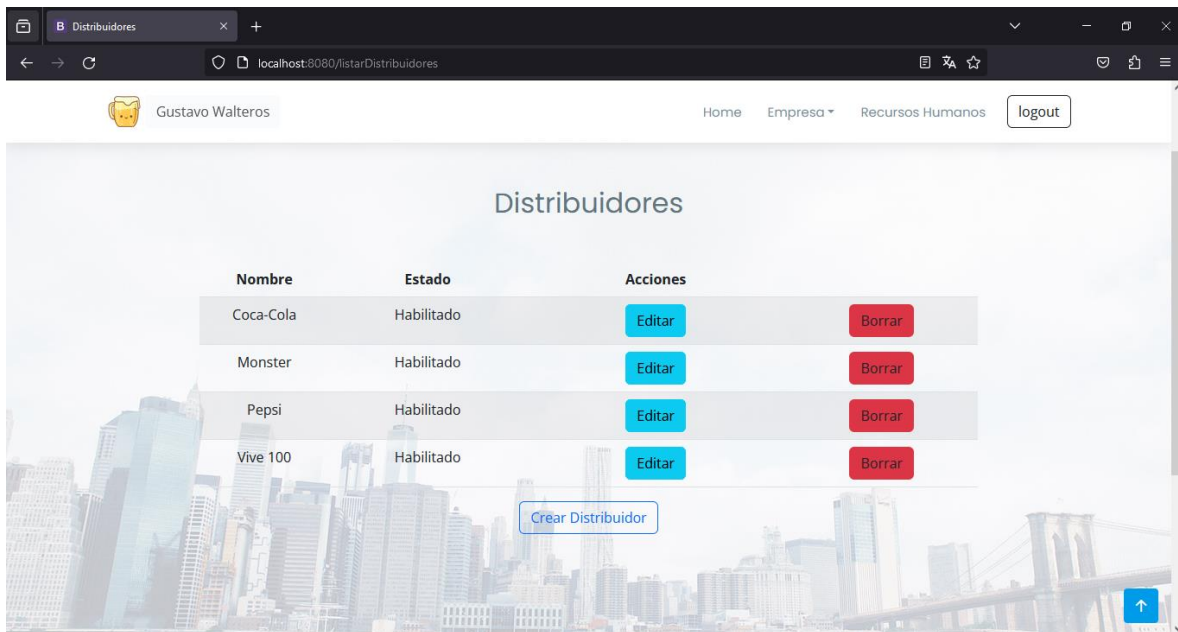


Figura 106: Distribuidores  
Fuente: Autor

Continuamos, dando clic en el botón de crear distribuidor se nos redirigirá a la página de crear distribuidor para digitar el nombre y agregarlo dando clic en el botón de guardar, si tratamos de guardar sin tener digitado nada nos saltará una advertencia en el recuadro pidiendo ingresar un distribuidor.

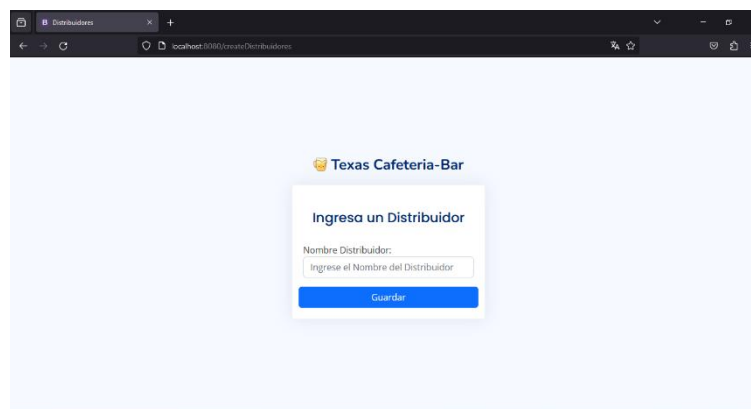


Figura 107: Crear Distribuidores  
Fuente: Autor

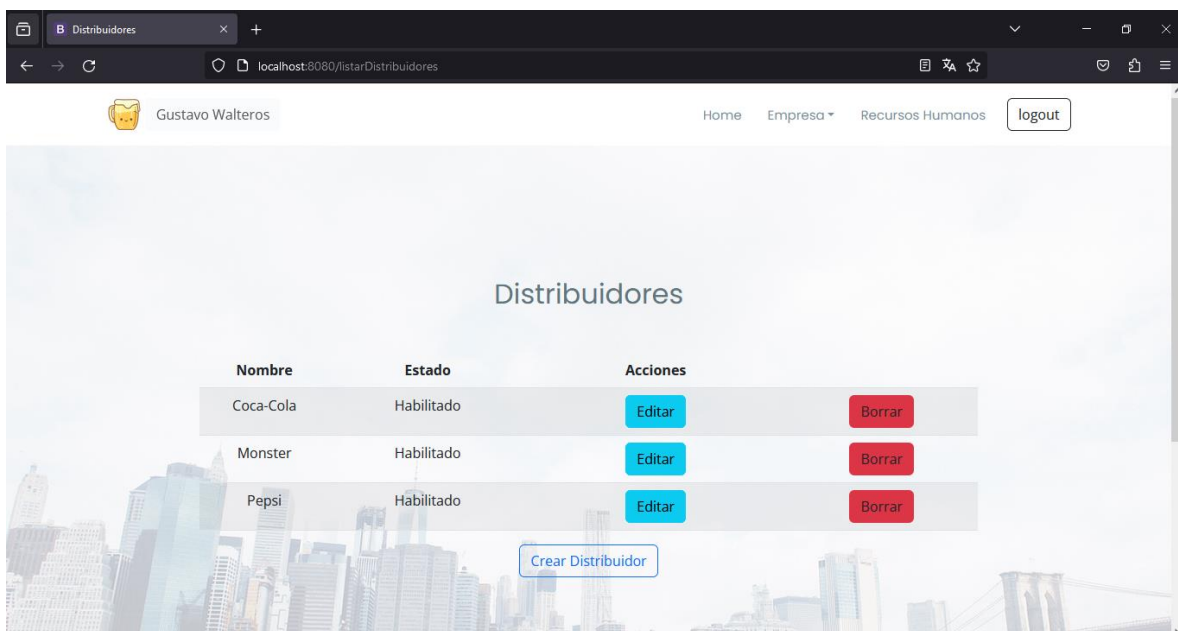
En la página de distribuidores podemos interactuar con los botones de acción uno nombrado editar y otro borrar, el botón de editar es para modificar o cambiar los datos ya registrados del distribuidor existente dentro del sistema, dándole clic se nos redirigirá a otra

página para reemplazar la información ya digitada y el botón de borrar elimina los datos permanentes seleccionados del sistema.



Formulario para ingresar un distribuidor. El título es "Ingresa un Distribuidor". Hay un campo de texto con el valor "Monsters" y un botón azul "Guardar". Debajo del botón hay un enlace "Quieres Regresar? Volver".

Figura 108: Editar Distribuidores  
Fuente: Autor



Captura de pantalla de la interfaz de usuario. La barra de navegación superior muestra el nombre de usuario "Gustavo Walteros" y los enlaces "Home", "Empresa", "Recursos Humanos" y "logout". El título principal de la sección es "Distribuidores".

Nombre	Estado	Acciones
Coca-Cola	Habilitado	<a href="#">Editar</a> <a href="#">Borrar</a>
Monster	Habilitado	<a href="#">Editar</a> <a href="#">Borrar</a>
Pepsi	Habilitado	<a href="#">Editar</a> <a href="#">Borrar</a>

Debajo de la tabla hay un botón "Crear Distribuidor".

Figura 109: Borrar Distribuidores  
Fuente: Autor

Nos desplazamos por la barra de navegación situándonos en la sección de empresa desplegando el menú para acceder a la pestaña de Productos, visualizando la información de todos los productos registrados en el sistema con sus datos, aparte, botones de acción como editar, borrar y un apartado con otro botón crear producto.

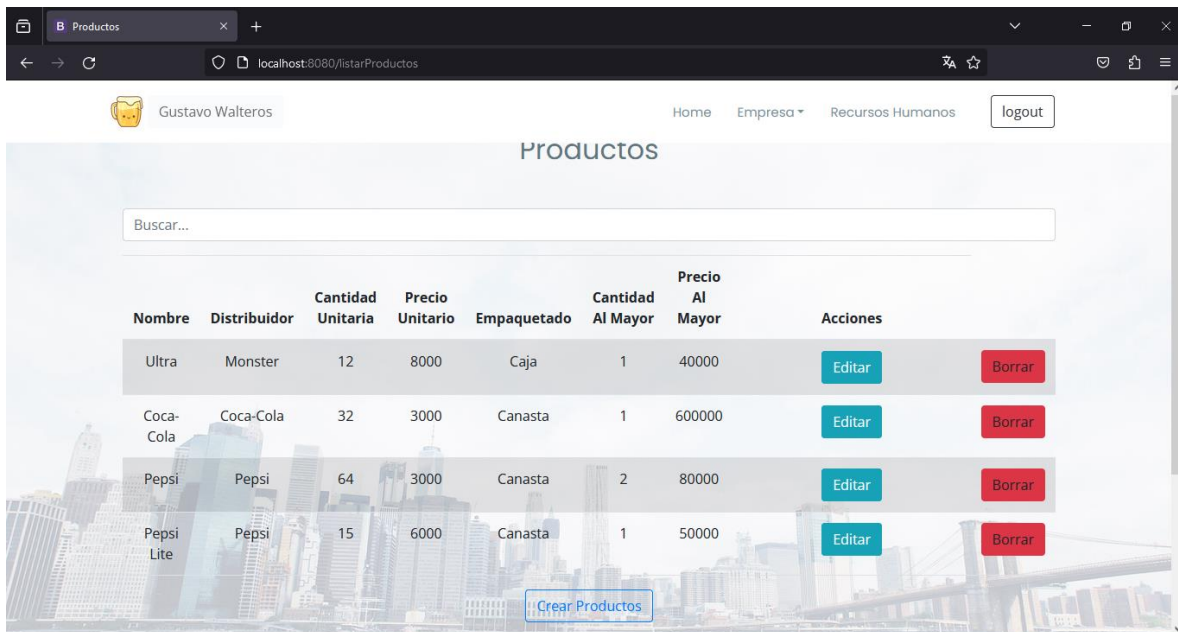


Figura 110: Productos  
Fuente: Autor

Continuamos, dando clic en el botón de crear producto se nos redirigirá a la página de crear producto para digitar los datos pedidos y agregarlo dando clic en el botón de guardar, si tratamos de guardar sin tener digitado nada nos saltará una advertencia en los recuadros pidiendo ingresar la información específica, en el apartado de distribuidor y empaquetado se seleccionaran desplegando un menú de opciones en cada recuadro correspondiente eligiendo la información de cada uno, si uno por error entra en la parte final del formulario se señalara de color amarillo el hipervínculo para ser dirigido a la página de productos.

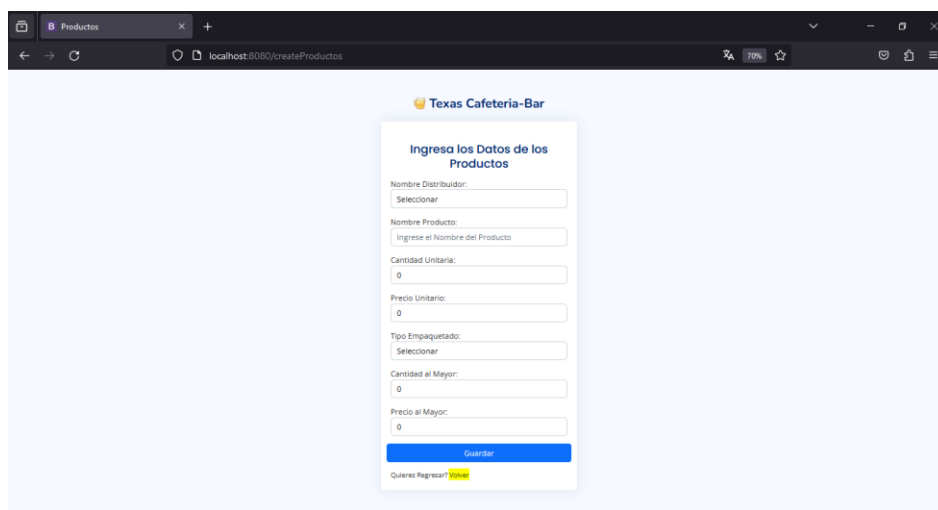


Figura 111: Crear Productos  
Fuente: Autor

En la página de productos podemos interactuar con los botones de acción uno nombrado editar y otro borrar, el botón de editar es para modificar o cambiar los datos ya registrados del producto existente dentro del sistema, dándole clic se nos redirigirá a otra página para reemplazar la información ya digitada y el botón de borrar elimina los datos permanentes seleccionados del sistema.

**Texas Cafeteria-Bar**

**Ingresa los Datos de los Productos**

Nombre Distribuidor:

Nombre Producto:

Cantidad Unitaria:

Precio Unitario:

Tipo Empaquetado:

Cantidad al Mayor:

Precio al Mayor:

[Guardar](#)

[Quieres Regresar? Volver](#)

*Figura 104: Editar Productos*  
*Fuente: Autor*

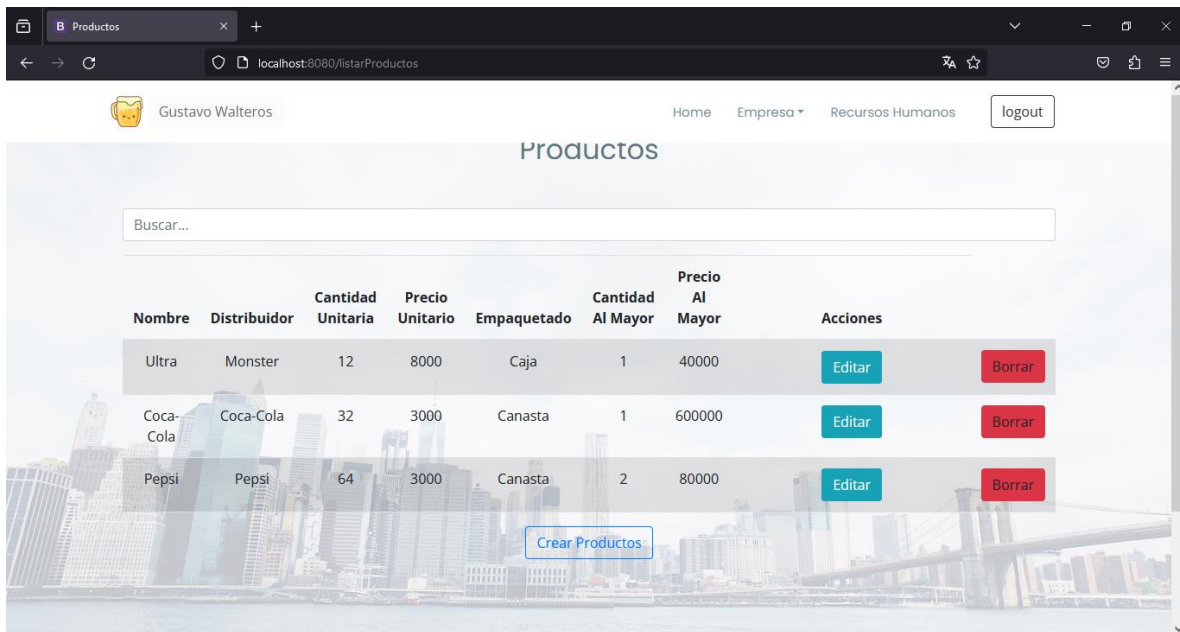


Figura 112: Borrar Productos  
Fuente: Autor