

GESTIÓN DE CONTACTOS

Nombre del proyecto:	GESTIÓN DE CONTACTOS
Fecha:	10 DE NOVIEMBRE DEL 2024
Creado por:	Elkin Stiven Contreras, Danna Michelle Morales, Alejandro Ortiz, David Felipe Perdomo
Objetivo:	Llevar documentado el manejo del proyecto

Índice

- 1. **Introducción**
 - 1.1 Alcance del producto
 - 1.2 Valor del producto
 - 1.3 Público objetivo
- 2. **Requisitos funcionales**
 - 2.1 Restricciones
 - 2.2 Suposiciones y dependencias
- 3. **Metodología**
 - 3.1 Design Thinking
 - 3.2 MosCoW
- 4. **Requisitos no funcionales**
 - 4.1 Seguridad
 - 4.2 Usabilidad
 - 4.3 Infraestructura
- 5. **Anexos**
 - 5.1 Caso de uso
 - 5.2 Diagrama de clases
 - 5.3 Diagrama de actividades
 - 5.4 Diagrama de paquetes

1.Introducción

La gestión de contactos es una solución tecnológica diseñada para facilitar la organización, almacenamiento y manejo eficiente de la información personal o profesional de individuos o entidades. Este proyecto tiene como objetivo ofrecer una herramienta accesible y segura que permita a los usuarios realizar operaciones básicas como crear, leer, actualizar y eliminar (CRUD) contactos, optimizando así la forma en que gestionan sus relaciones e información.

En su fase inicial, la aplicación se centrará en proporcionar una experiencia intuitiva y enfocada en la privacidad, garantizando que todos los datos sean gestionados de manera local y segura. Los usuarios podrán registrar contactos con datos como nombres, apellidos y números de teléfono, lo que les permitirá acceder rápidamente a esta información desde cualquier dispositivo conectado a la web.

1.1 Alcance del producto

Beneficios

- 1. Organización eficiente: Facilita el manejo de contactos, agrupándolos y almacenándolos de manera estructurada.
- 2. Acceso rápido: Permite acceder fácilmente a la información de contacto desde cualquier dispositivo conectado.
- 3. Privacidad y seguridad: Garantiza que los datos personales sean gestionados localmente y protegidos contra accesos no autorizados.
- 4. Simplicidad en el manejo: Ofrece una interfaz intuitiva para usuarios de todos los niveles tecnológicos.

Objetivos

- 1. Proporcionar una herramienta confiable para almacenar y organizar información de contactos personales y profesionales.
- 2. Fomentar la digitalización y automatización en la gestión de contactos.
- 3. Asegurar una experiencia de usuario que combine facilidad de uso con seguridad de los datos.

Metas

- 1. Establecer un sistema que permita realizar operaciones CRUD (Crear, Leer, Actualizar, Eliminar) de manera eficiente.
- 2. Mejorar la experiencia de los usuarios en la organización y consulta de sus contactos.
- 3. Crear una solución escalable que se adapte a las necesidades futuras, como la integración con servicios adicionales o dispositivos móviles.

Referencias

- IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications.
Documentación de desarrollo de software.s

Visión general del documento:

Este documento está organizado en secciones que describen la visión general de la aplicación, los requisitos específicos y cualquier información adicional relevante.

1.2 Valor del producto

Número de personas: 4

Horas por día: 8

Días por mes: 22 (5 días laborales por semana)

Meses: 2

Calcular el costo total del trabajo:

Sueldo mínimo mensual: COP\$1,100,000

Número de meses: 2

Costo por persona = COP\$2,200,000

Costo total para 4 personas = COP\$8,800,000

Sueldo mínimo mensual COP\$1,100,000

Horas trabajadas por mes = 176 horas/mes

Total, de horas trabajadas: 1056 horas

Valor por hora: COP\$7,386.36/hora

Valor total del trabajo = COP\$8,800,000

El valor del trabajo realizado en la fase inicial del proyecto, es de **COP\$8,800,000**.

1.4 Público objetivo

El público objetivo de la aplicación de gestión de contactos incluye usuarios generales que buscan una solución práctica y segura para almacenar y organizar información personal o profesional. En su versión inicial, la aplicación está diseñada para individuos que requieren una herramienta sencilla para gestionar sus contactos de manera eficiente.

Características del usuario:

Usuarios generales que buscan agendar sus contactos de manera organizada.

Restricciones:

El entorno de esta aplicación solo será web.

Suposiciones y dependencias:

Se supone que los usuarios tienen acceso a internet a la hora de interactuar con la web.

2. Requisitos Funcionales

• Registro e Inicio de Sesión (RF1)

Descripción:

El sistema permitirá que los usuarios se registren y, si ya tienen una cuenta, inicien sesión para acceder a sus contactos.

Entrada de datos (Registro):

- Nombres

- Apellidos
- Nombre de usuario
- Contraseña
- Número de teléfono

Entrada de datos (Inicio de sesión):

- Nombre de usuario
- Contraseña

Restricciones:

- Los campos "Nombre", "Apellidos" y "Número de teléfono" son obligatorios.

Gestión de contactos (RF2)

Descripción:

El sistema permitirá realizar las operaciones CRUD (Crear, Leer, Actualizar y Eliminar) para gestionar contactos.

Entrada de datos (Para agregar/actualizar contactos):

- Nombre
- Apellidos
- Número de teléfono

Restricciones:

- Los campos "Nombres", "Apellidos" y "Número de teléfono" son obligatorios.

Flujo o Proceso

1. Acceso al sistema

El usuario accede al sistema y elige entre registrarse o iniciar sesión.

Registrar Usuario (RF1)

Entrada:

- Nombres, Apellidos, Nombre de usuario, Contraseña, Número de teléfono.

Proceso:

1. Validar la información ingresada en el formulario de registro.
2. Almacenar los datos del usuario en la base de datos.

Salida:

- El usuario queda registrado y puede iniciar sesión.

Inicio de Sesión (RF1)

Entrada:

- Nombre de usuario, Contraseña.

Proceso:

1. Validar las credenciales ingresadas.
2. Permitir el acceso a la aplicación si las credenciales son correctas.

Salida:

- El usuario accede a la interfaz principal de la aplicación.

Gestión de contactos (RF2)

1. Agregar contacto.

Entrada:

- Nombre, Apellidos, Número de teléfono.

Proceso:

2. Validar los datos ingresados.
3. Almacenar los datos del contacto en la base de datos.

Salida:

- Contacto agregado exitosamente.

2. Editar contacto:

Entrada:

- Identificador del contacto, Nombre, Apellidos, Número de teléfono.

Proceso:

2. Validar los datos ingresados.
3. Actualizar la información del contacto en la base de datos.

Salida:

- Contacto actualizado exitosamente.

3. Eliminar contacto:

Entrada:

- Identificador del contacto.

Proceso:

2. Buscar el contacto en la base de datos.
3. Eliminar el contacto seleccionado.

Salida:

- Contacto eliminado exitosamente.

4. Ver contactos:

Proceso:

1. Recuperar todos los contactos asociados al usuario desde la base de datos.
2. Mostrar los contactos en una lista organizada.

Salida:

- Lista de contactos visualizada en pantalla.

Cierre de sesión

Proceso:

1. El usuario selecciona la opción de cerrar sesión.
2. El sistema invalida la sesión activa y redirige a la página de inicio.

Salida:

- El usuario sale del sistema.

FIN.

2.1 Restricciones

El sistema de gestión de contactos será desarrollado utilizando un modelo cliente/servidor, garantizando eficiencia, escalabilidad y seguridad en el manejo de la información de los usuarios y sus contactos.

Lenguajes y Herramientas de Desarrollo:

- **Lenguaje de programación:**

- JavaScript
- TypeScript
- HTML
 - **Backend:**
- El desarrollo del backend se realizará en **Java** utilizando **Spring Boot** como framework principal.
 - **Base de datos:**
- Se utilizará **MySQL** como sistema de gestión de bases de datos relacional para almacenar de forma segura los datos de los usuarios y sus contactos.
 - **IDE utilizado:**
- El desarrollo del sistema se llevará a cabo en **IntelliJ IDEA**, que proporciona herramientas avanzadas para el desarrollo, depuración y despliegue.

Arquitectura del Sistema:

El sistema sigue un modelo cliente/servidor:

1. Servidor:

- El backend, implementado en Spring Boot, actúa como el servidor que almacena, gestiona y procesa la información de los usuarios y sus contactos.
- Este servidor estará encargado de manejar todas las operaciones de CRUD (Crear, Leer, Actualizar, Eliminar), garantizando la seguridad y la integridad de los datos.

2. Clientes:

- Los dispositivos de los usuarios (computadoras, tabletas, smartphones) funcionarán como clientes.
- Los clientes se conectarán al servidor a través de la interfaz web, desarrollada en HTML y TypeScript, permitiendo el acceso a las funcionalidades del sistema.

Interfaz de Usuario:

- La interfaz gráfica será diseñada para ser intuitiva y fácil de manejar, de modo que cualquier usuario, sin necesidad de formación previa, pueda realizar las operaciones necesarias.
- La interfaz permitirá realizar las siguientes acciones:
 - Registro e inicio de sesión.
 - Gestión de contactos (crear, ver, actualizar y eliminar).
 - Visualización de mensajes de confirmación y advertencias relacionadas con los datos ingresados.

Ventajas del Modelo Cliente/Servidor:

1. Centralización de Datos:

- Toda la información se almacenará en el servidor, permitiendo mayor control y seguridad.

2. Acceso Multidispositivo:

- Los usuarios podrán acceder al sistema desde cualquier dispositivo con conexión a internet.

3. Escalabilidad:

- La arquitectura permite que el sistema crezca en funcionalidades y usuarios sin comprometer el rendimiento.

4. Seguridad:

- La información de los contactos y credenciales de los usuarios estará protegida mediante mecanismos de autenticación y autorización implementados en el servidor.

Esta arquitectura y tecnologías garantizan que el sistema de gestión de contactos sea robusto, seguro y fácil de usar.

2.2 Suposiciones y dependencias

Aumento de Usuarios Activos:

El sistema debe estar preparado para manejar un incremento significativo en la cantidad de usuarios y contactos almacenados.

Usabilidad:

La interfaz debe ser intuitiva y fácil de usar, permitiendo a los usuarios gestionar contactos de manera eficiente sin necesidad de capacitación previa.

Capacidad de Respuesta:

El sistema debe garantizar la disponibilidad y estabilidad ante aumentos repentinos de tráfico, evitando caídas en el servidor.

Infraestructura Escalable:

El sistema dependerá de un servidor configurado con Spring Boot y una base de datos MySQL capaz de manejar un alto volumen de datos y consultas.

Seguridad de Datos:

Se priorizará la privacidad y protección de la información proporcionada por los usuarios, asegurando un control de acceso adecuado.

3. Metodología

3.1 Design Thinking

1. Empatizar:

Los usuarios necesitan una aplicación sencilla e intuitiva que les permita registrar, organizar y administrar sus contactos de manera eficiente. Buscan funcionalidades claras para crear, actualizar y eliminar contactos, asegurando que la información esté siempre accesible y segura. Además, valoran una interfaz que facilite la navegación sin necesidad de conocimientos técnicos avanzados.

2. Definir:

- Los usuarios enfrentan dificultades para gestionar grandes cantidades de contactos de forma estructurada.
- Existe una necesidad de garantizar la seguridad y privacidad de los datos personales.
- La ausencia de una plataforma centralizada complica la organización y el acceso a los contactos almacenados.
- Se requiere una experiencia fluida y sin errores para maximizar la productividad del usuario.

3. Idear:

El sistema debe incluir:

- Un registro de usuarios con validación de datos obligatorios.
- Funcionalidades para crear, actualizar, eliminar y visualizar contactos fácilmente.
- Una interfaz de usuario intuitiva y amigable, accesible desde dispositivos móviles y computadoras.
- Mecanismos de seguridad robustos, como autenticación segura y cifrado de datos sensibles.
- Escalabilidad para manejar un aumento en el volumen de contactos y usuarios.

4. Prototipar:

- Los usuarios podrán registrarse, iniciar sesión y cambiar su contraseña.

- Se permitirá gestionar contactos mediante un sistema CRUD (Crear, Leer, Actualizar, Eliminar).
 - El backend será desarrollado en Spring Boot, con la base de datos MySQL para almacenar la información.
 - La interfaz estará diseñada con HTML y JavaScript, asegurando un diseño limpio y funcional.
5. **Evaluar:**
- Se implementará un período de prueba para recopilar retroalimentación de los usuarios y realizar ajustes necesarios, asegurando que el sistema cumpla con las expectativas de funcionalidad, seguridad y usabilidad.

3.2 MosCoW

Must Have	Should Have	Could Have	Won't Have
<div><div>- Registrarse con validación de campos.</div><div>- Iniciar sesión con validación de credenciales</div><div>- Crear contacto con validación.</div><div>- Actualizar contacto con validación.</div><div>- Eliminar contacto.</div><div>- Ver lista de contactos.</div><div>- Validación de campos obligatorios: Nombre, Apellidos, Número de teléfono.</div></div>	<div><div>- Mensajes de confirmación en CRUD:</div><div>- "Contacto agregado exitosamente".</div><div>- "Contacto actualizado correctamente".</div><div>- "Contacto eliminado exitosamente".</div></div>	<div><div>- Posibilidad de ver un historial de cambios en los contactos</div><div>- Funcionalidad de búsqueda avanzada de contactos por nombre, apellido o número de teléfono</div><div>-Opción para agregar contactos a grupos (por ejemplo, familia, amigos, trabajo) para facilitar la organización.</div></div>	<div><div>- Funcionalidad de búsqueda de contactos.</div><div>- Validación de formatos de correo y teléfono al agregar o actualizar contactos.</div></div>

4. Requisitos no funcionales

- 4.1 Seguridad
- El sistema debe garantizar que los usuarios solo tengan acceso a los recursos que les corresponden.
- 4.2 Usabilidad
- La interfaz de usuario será intuitiva, permitiendo que cualquier usuario utilice las funciones principales de la aplicación sin necesidad de formación previa.
- 4.3 Infraestructura

- Backend: La aplicación será desarrollada utilizando Spring Boot como framework para la API en Java.
- Frontend: El frontend estará construido con HTML nativo, sin el uso de frameworks o bibliotecas adicionales como Angular o React.
- Servidor: El backend se ejecutará en un servidor Tomcat.
- Base de datos: Se utilizará una base de datos relacional, específicamente MySQL.

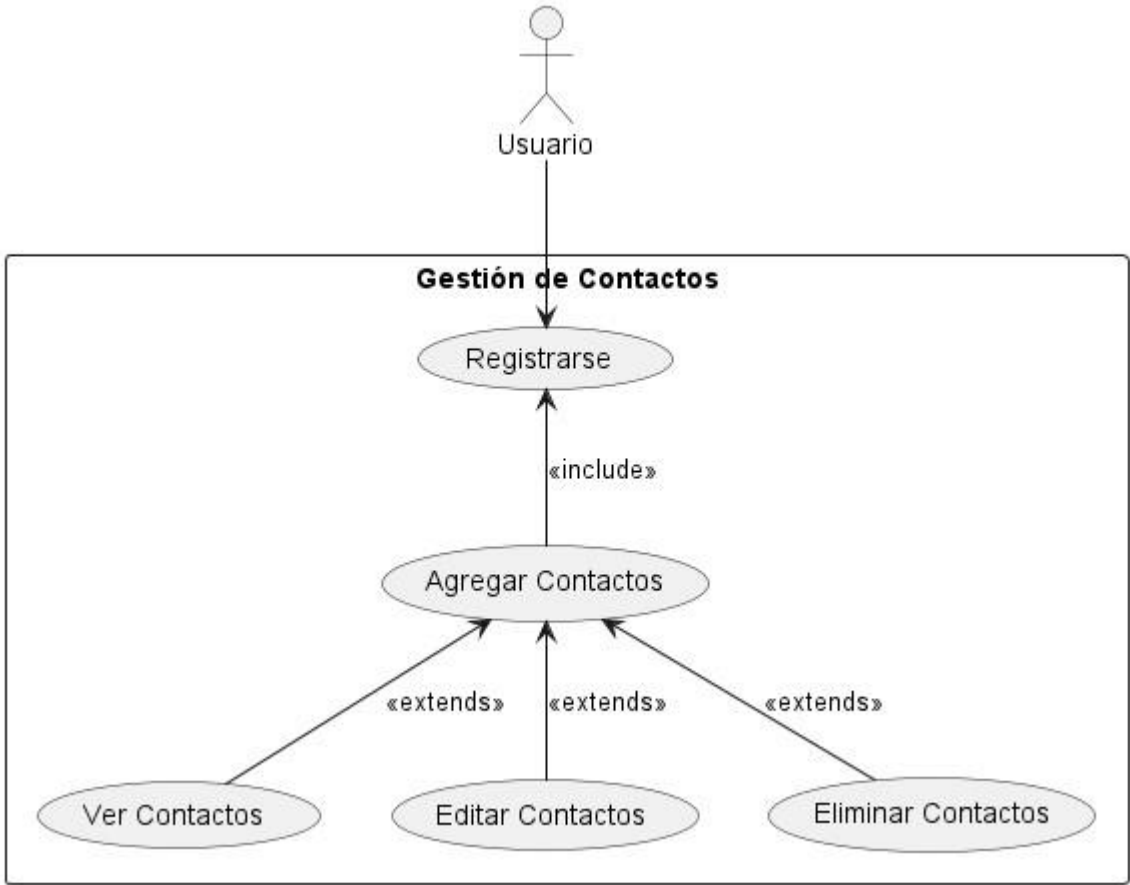
5. Anexos

5.1 Caso de Uso: Caso General

- **Actor:** Usuario.
- **Descripción:** La persona se registra y puede gestionar sus contactos.
- **Flujo Principal:** La persona se registra en el sistema y puede agregar, eliminar, editar y ver contactos.

Condiciones:

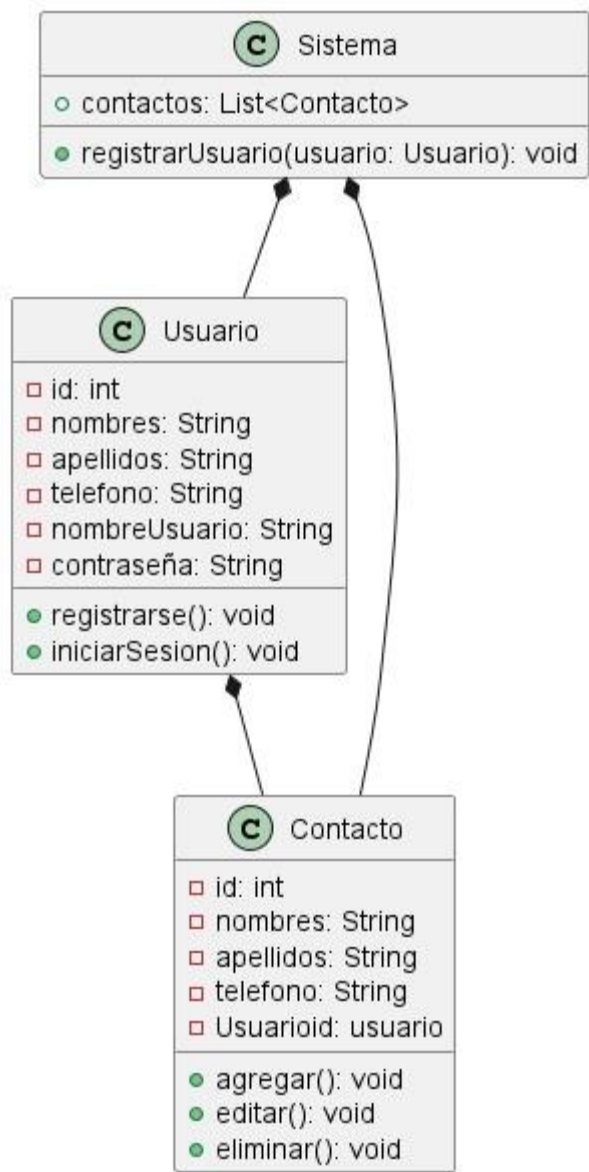
- La persona debe estar registrada para poder gestionar contactos.
- No puede ver, editar o eliminar contactos si no ha agregado al menos un contacto.



5.2 Diagrama de Clases

- Un usuario se registra en el sistema a través de la clase usuario.
- El usuario puede agregar, editar y eliminar contactos asociados a su cuenta, utilizando la clase contacto.

•



El sistema actúa como el núcleo que administra tanto a los usuarios como a los contactos.

5.3 Diagrama de Actividades

1. Inicio del flujo:

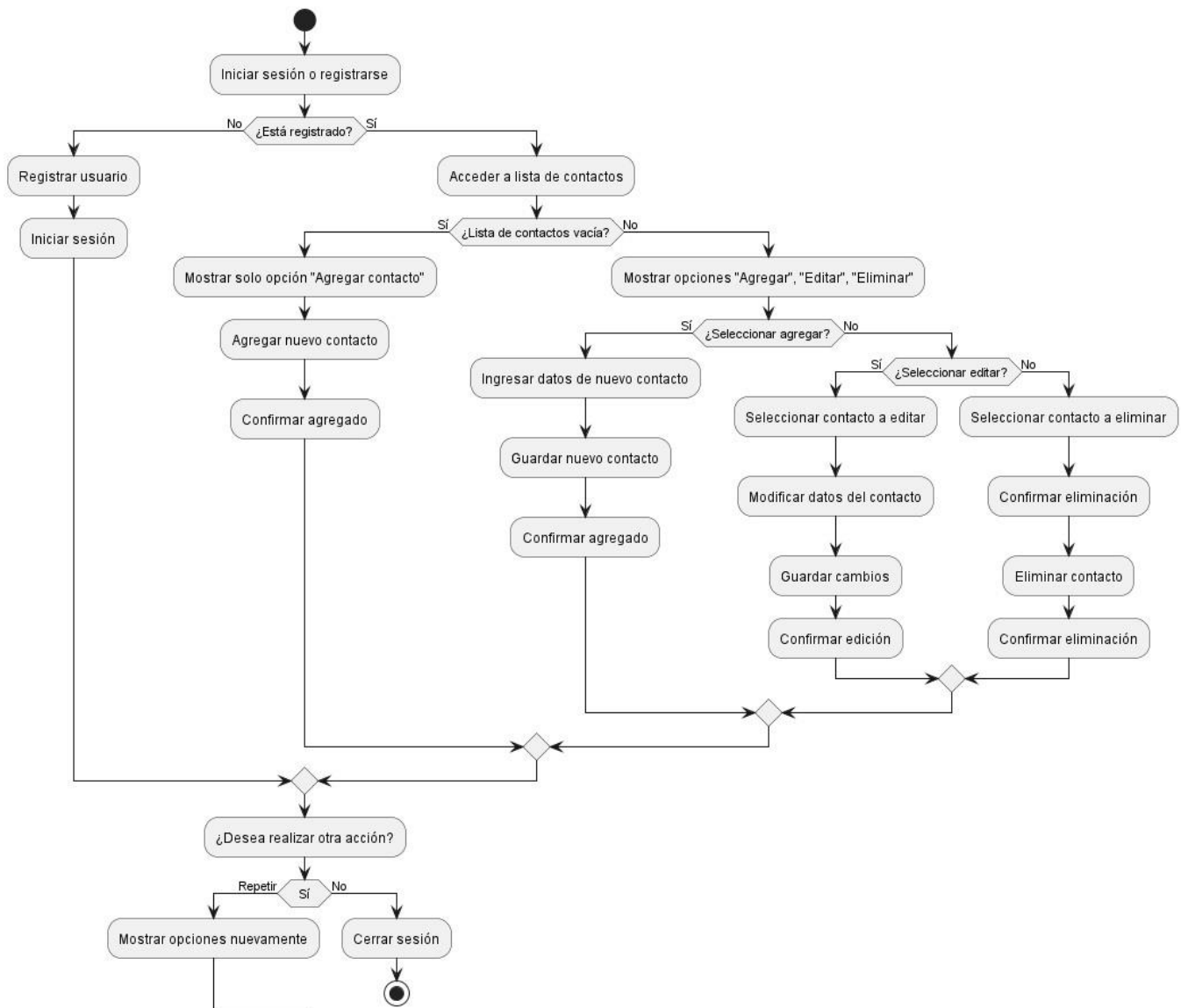
- El proceso comienza con el usuario registrándose o iniciando sesión.
- Si el usuario no está registrado, se le pide que lo haga y luego se le redirige a la pantalla de inicio de sesión.
- Si el usuario ya está registrado, accede directamente a la lista de contactos.

2. Acceso a la lista de contactos:

- Si la lista de contactos está vacía: El sistema muestra solo la opción de agregar un contacto.
- El usuario ingresa los datos del contacto, se guarda y se confirma el agregado.
- Si hay contactos en la lista: El sistema muestra tres opciones: Agregar, Editar, Eliminar.

3. Verificación de continuación:

- Después de realizar cualquiera de las acciones (agregar, editar o eliminar), el sistema pregunta si el usuario desea realizar otra acción.
- Si el usuario decide realizar otra acción, el flujo vuelve a mostrar las opciones disponibles.
- Si el usuario decide salir, el proceso termina con el cierre de sesión.



5.4 Diagrama de Paquetes

- El sistema permite a un usuario registrarse e iniciar sesión (paquete Autenticación).
- Una vez autenticado, el usuario puede acceder a la gestión de contactos (paquete Gestión de Contactos).
- La interfaz de usuario es responsable de mostrar las vistas correspondientes (paquete Interfaz de Usuario).
- La lógica de negocio se asegura de que las acciones sean válidas (paquete Lógica de Negocio).

