

Universidad de San Carlos de Guatemala.  
Centro Universitario de Occidente.  
División de Ciencias de la Ingeniería.  
Análisis y Diseño I.  
Ing. Daniel Gonzales.



## PROYECTO 1 - SIE

Estudiante	Carnet
201930643	Jeffrey Kenneth Menéndez Castillo
201931811	Diego José Maldonado Monterroso
202131284	José Roberto Bautista Rojas

Quetzaltenango, Guatemala.

Lunes 29 de septiembre de 2025.

## VERSIONES FRONTEND ANGULAR

Para la implementación de nuestro Frontend se decidió usar Angular con las siguientes versiones, ya que Angular es un Framework robusto que permite crear SPAs de manera ágil y sencilla.

*Angular Core: ^20.2.0 (versión más reciente/experimental)*

*Angular CLI: ^20.2.2*

*Angular Compiler: ^20.2.0*

*Angular Build: ^20.2.2*

*TypeScript: ~5.9.2*

*TailwindCSS: ^4.1.13 (versión v4, la más reciente)*

*TailwindCSS PostCSS: ^4.1.13*

*TailwindCSS PrimeUI: ^0.6.1*

*PrimeNG: ^20.1.1 (biblioteca de componentes UI)*

*PrimeIcons: ^7.0.0*

*@primeuix/themes: ^1.2.3*

*PostCSS: ^8.5.6*

*tslib: ^2.3.0 (TypeScript helper library)*

*@ng-icons/core: ^32.1.0*

*@ng-icons/heroicons: ^32.1.0*

*RxJS: ~7.8.0*

*Jasmine: ~5.9.0 (framework de testing)*

*Jasmine Types: ~5.1.0*

*Karma: ~6.4.0 (test runner)*

*Karma Chrome Launcher: ~3.2.0*

*Karma Coverage: ~2.2.0*

*Karma Jasmine: ~5.1.0*

*Karma Jasmine HTML Reporter: ~2.1.0*

## VERSIONES BACKEND SPRING BOOT

Elegimos Spring Boot ya que ofrece un entorno empresarial perfecto para aplicar arquitecturas limpias, desacomplando la lógica en capas.

*Spring Boot: 3.5.6*

*Java: 21 (OpenJDK)*

*Maven: 3.9.x (Build Tool)*

*Spring Security: Autenticación y autorización*

*Spring Data JPA: Persistencia de datos*

*Spring Boot Actuator: Monitoreo y métricas*

*Spring Boot Mail: Envío de correos electrónicos*

*Spring Boot Validation: Validación de datos*

*MariaDB: 11.x (Base de datos principal)*

*Driver JDBC: MariaDB Java Client*

*JWT (JSON Web Tokens): 0.12.3 (jjwt-api)*

*Spring Security: Control de acceso basado en roles*

*SpringDoc OpenAPI: 2.7.0 (Swagger UI)*

*Documentación interactiva disponible en: <http://localhost:8080/api/v1/swagger-ui/index.html>*

*iText PDF: 5.5.13.3 (Generación de PDFs)*

*Apache POI: 5.2.4 (Manejo de Excel)*

*Lombok: Reducción de boilerplate code*

*Jackson: Serialización JSON*

*Jakarta Annotations: Anotaciones estándar*

## DESPLIEGUE CI/CD

Para aplicar integración y despliegue continuo decidimos implementar una solución usando GitHub Actions, ya que es una herramienta gratuita al usar GitHub y tener nuestros repositorios públicos.

Así mismo, decidimos usar DockerHub como container registry, el cual permite almacenar todas las versiones de nuestras imágenes que construyamos en nuestros procesos de Build y Push.

No optamos por usar Jenkins, ya que necesita más recursos y esto implicaría más gastos para el equipo.

Para usar GitHub Actions, solo necesitamos crear nuestra carpeta `.github` que contiene la carpeta `workflows`, la cual contendrá nuestro pipeline en formato `yml`.




En resumen y a grandes rasgos, la logica del pipeline es la siguiente:

1. Construye un contenedor en uno de los runners de GitHub públicos. En este caso usamos `ubuntu-latest` como imagen del contenedor.
2. Tiene un único job que se llama `build`.
3. El cual hace `checkout` a la rama que disparó el pipeline.
4. Luego configura Docker en el contenedor.
5. Luego hacemos `build` de nuestra imagen.
6. Hacemos login a Docker Hub. `# solo si estamos en main`
7. Hacemos push a nuestro container registry. `# solo si estamos en main`
8. Desplegamos hacia el servidor. `# solo si estamos en main`
9. Luego limpiamos imágenes viejas en el servidor.

Estos pipelines se inician en cualquier rama. Y despliegan cuando hay cambios solo en `main`.

# AZURE VIRTUAL MACHINE

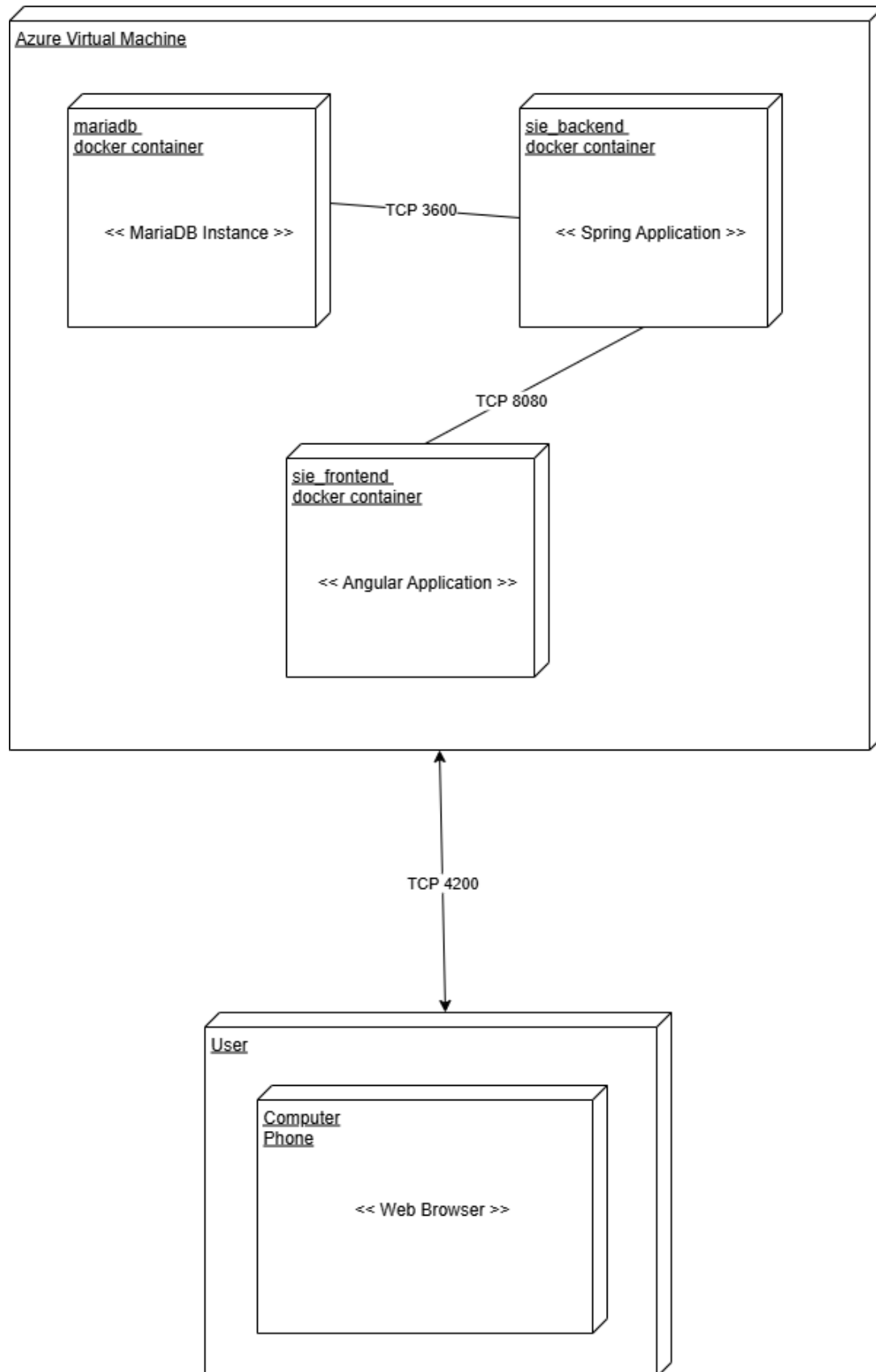
Esta es la configuración general de nuestra máquina virtual en Azure.

Propiedades	Supervisión	Funcionalidades (7)	Recomendaciones (2)	Tutoriales
<div><div><div><div></div><div>Máquina virtual</div></div><div><div>Nombre del equipo</div><div>ayd</div></div><div><div>Sistema operativo</div><div>Linux (ubuntu 22.04)</div></div><div><div>Generación de VM</div><div>V2</div></div><div><div>Arquitectura de VM</div><div>x64</div></div><div><div>Estado del agente</div><div>Ready</div></div><div><div>Versión del agente</div><div>2.14.0.1</div></div><div><div>Hibernación</div><div>Deshabilitado</div></div><div><div>Grupo host</div><div>-</div></div><div><div>Host</div><div>-</div></div><div><div>Grupo con ubicación por proximidad</div><div>-</div></div><div><div>Estado de ubicación</div><div>N/D</div></div></div><div><div><div></div><div>Redes</div></div><div><div>Dirección IP pública ⓘ</div><div>20.55.81.100 ( Interfaz de red ayd36_z1 )</div></div><div><div></div><div>1 direcciones IP públicas asociadas</div></div><div><div>Dirección IP pública (IPv6)</div><div>-</div></div><div><div>Dirección IP privada</div><div>172.16.0.7</div></div><div><div>Dirección IP privada (IPv6)</div><div>-</div></div><div><div>Red virtual/subred</div><div>vnet-eastus/snet-eastus-1</div></div><div><div>Nombre DNS</div><div>Configurar</div></div></div><div><div><div></div><div>Tamaño</div></div><div><div>Tamaño</div><div>Standard D2s v3</div></div><div><div>vCPU</div><div>2</div></div><div><div>RAM</div><div>8 GiB</div></div></div></div>				

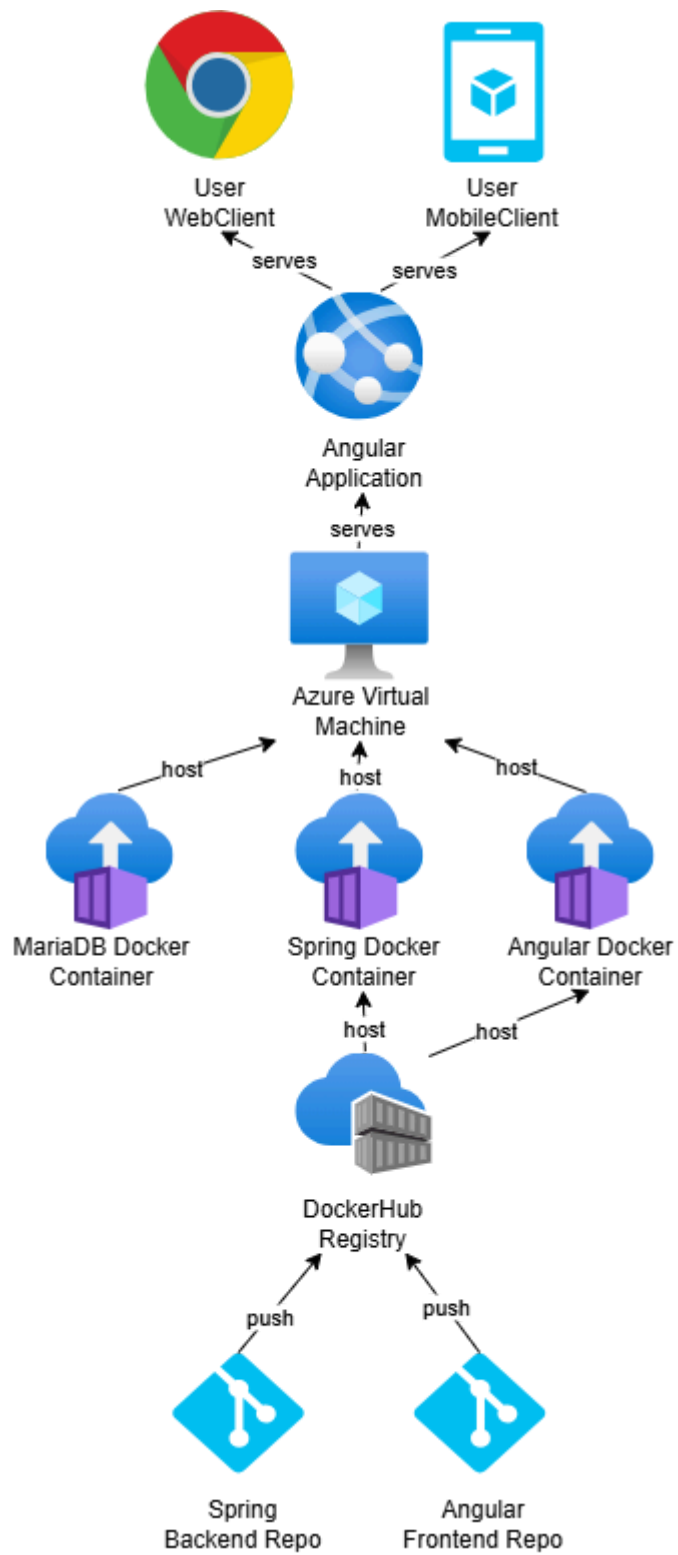
## DIAGRAMA ENTIDAD RELACION



## DIAGRAMA DE DESPLIEGUE

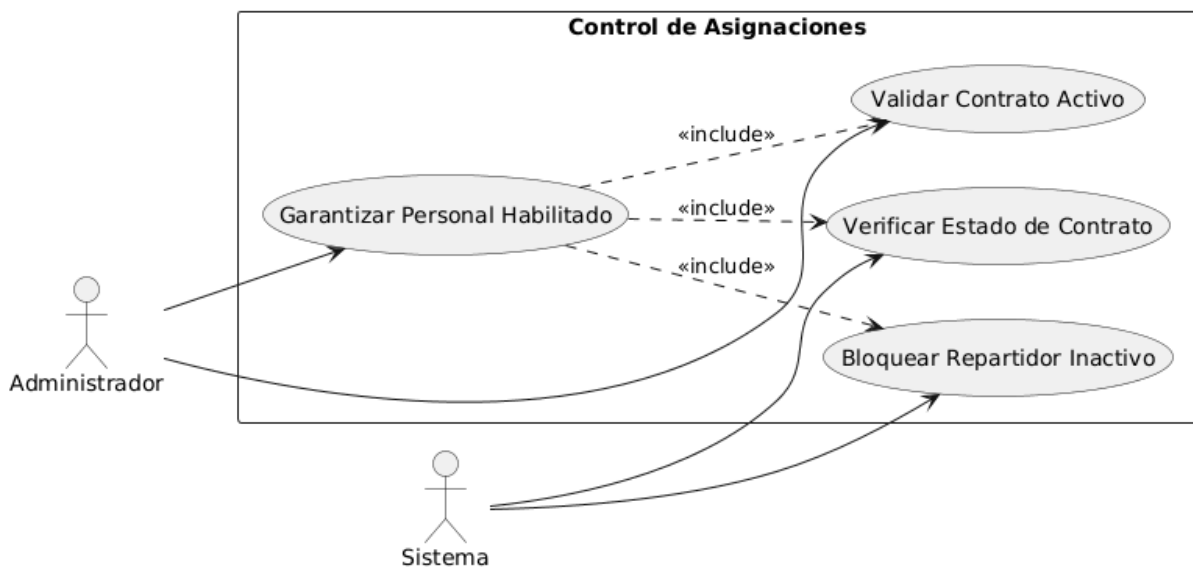
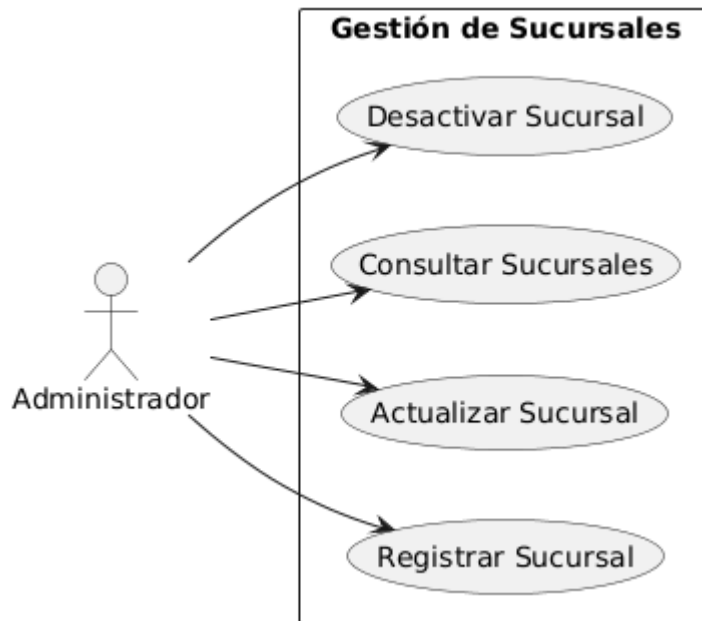


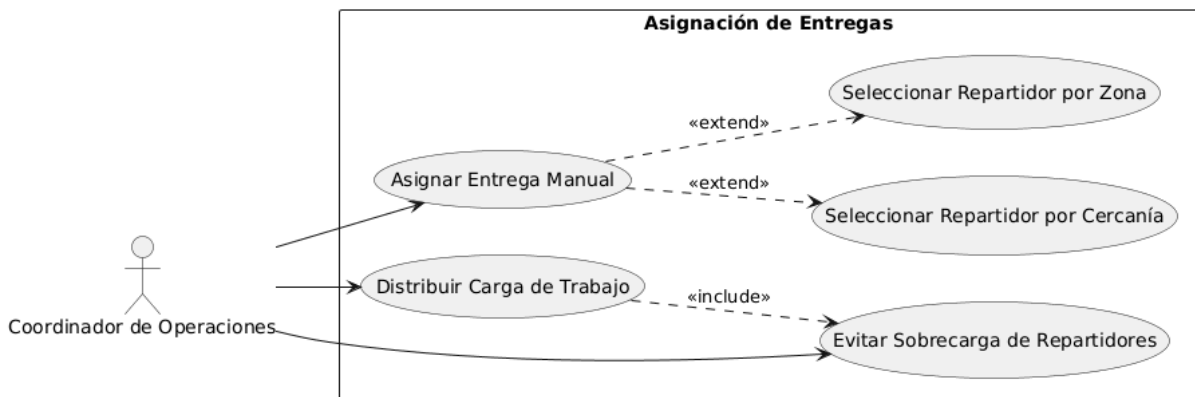
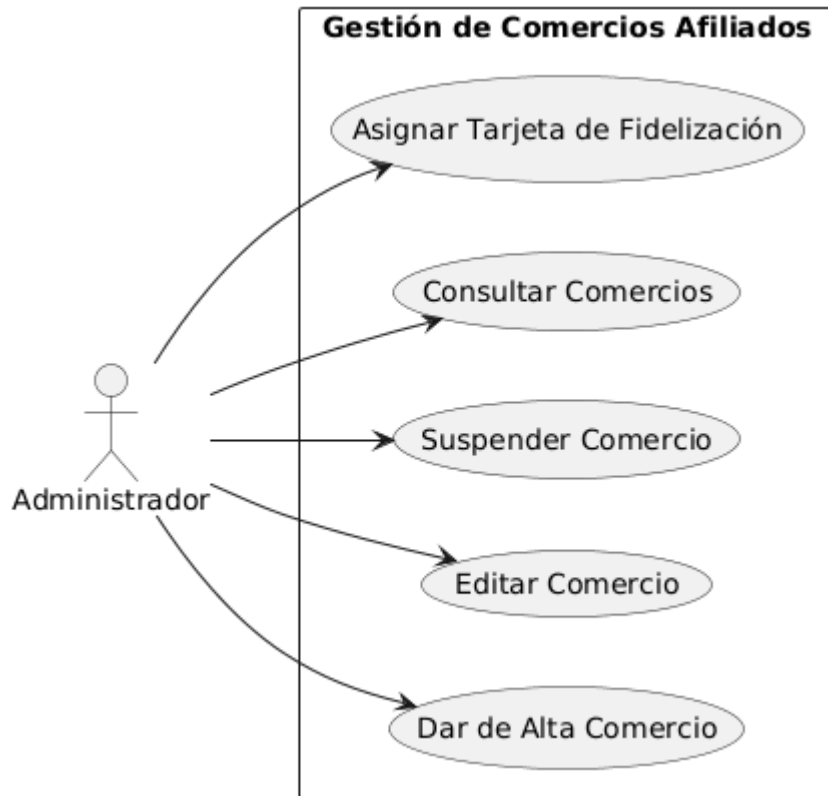
## DIAGRAMA DE ARQUITECTURA

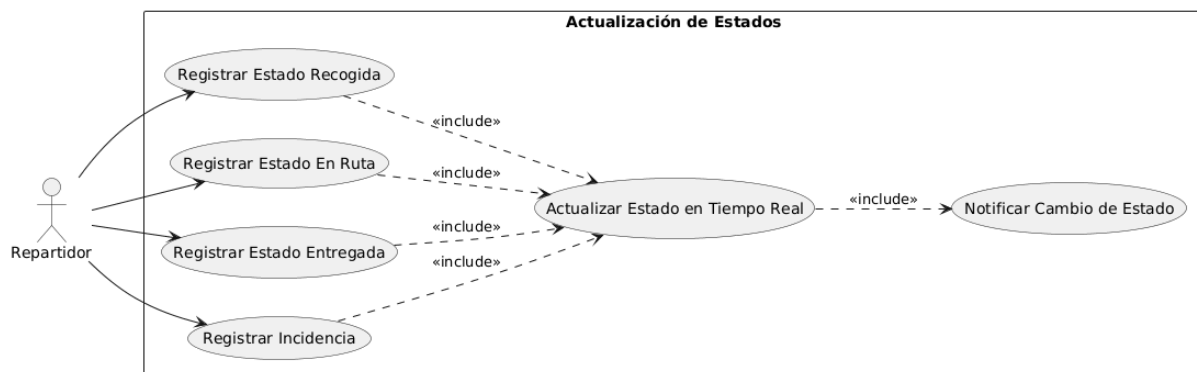
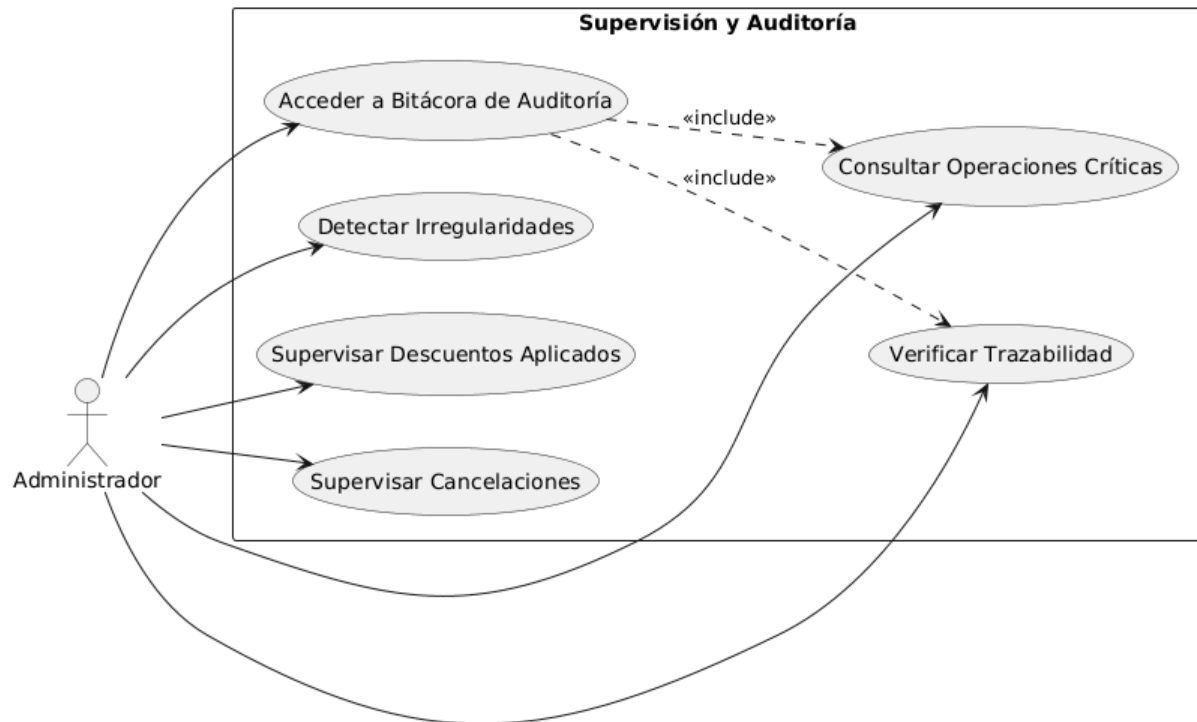


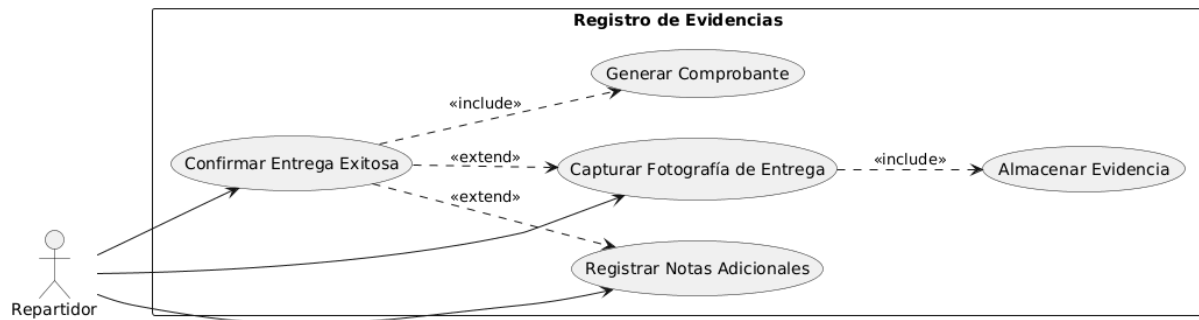
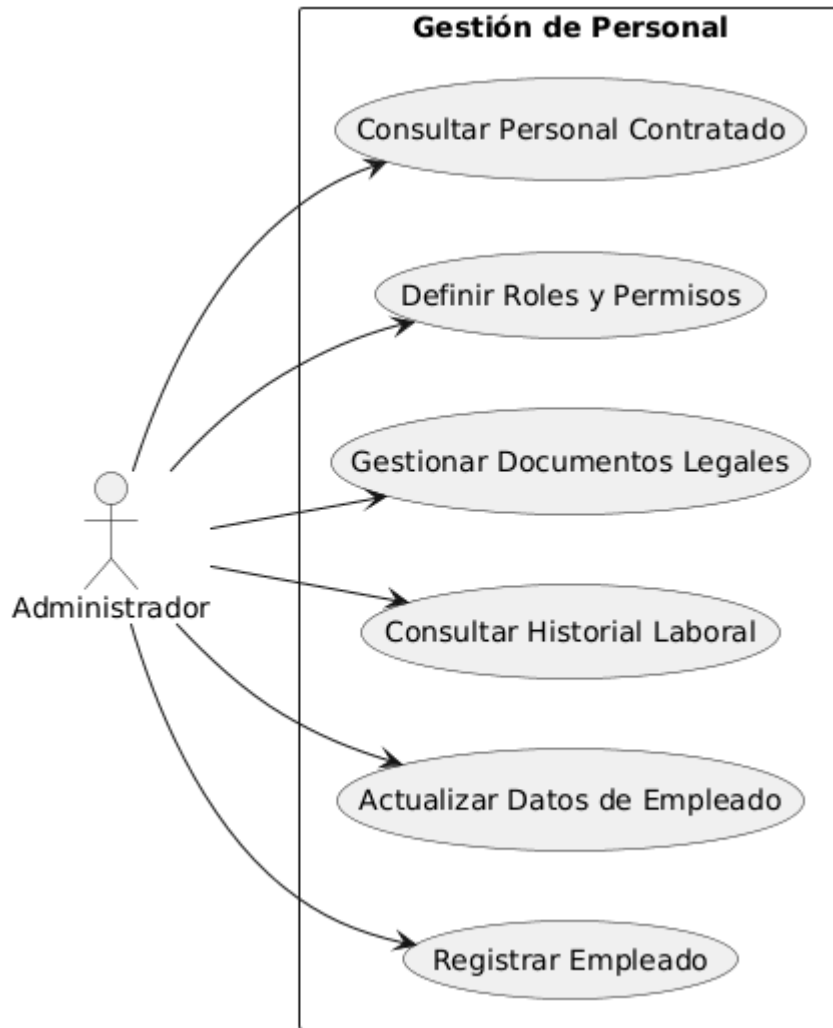


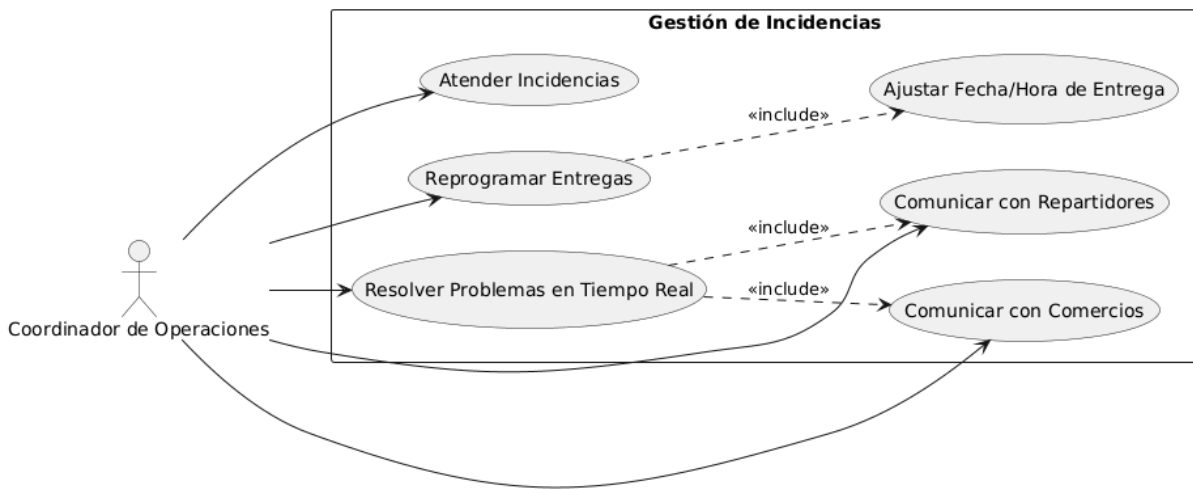
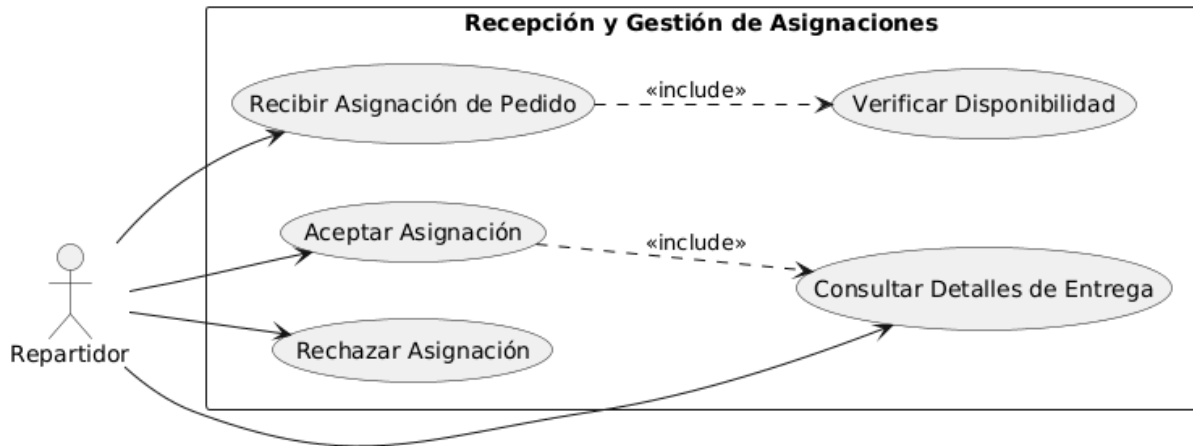
## CASOS DE USO

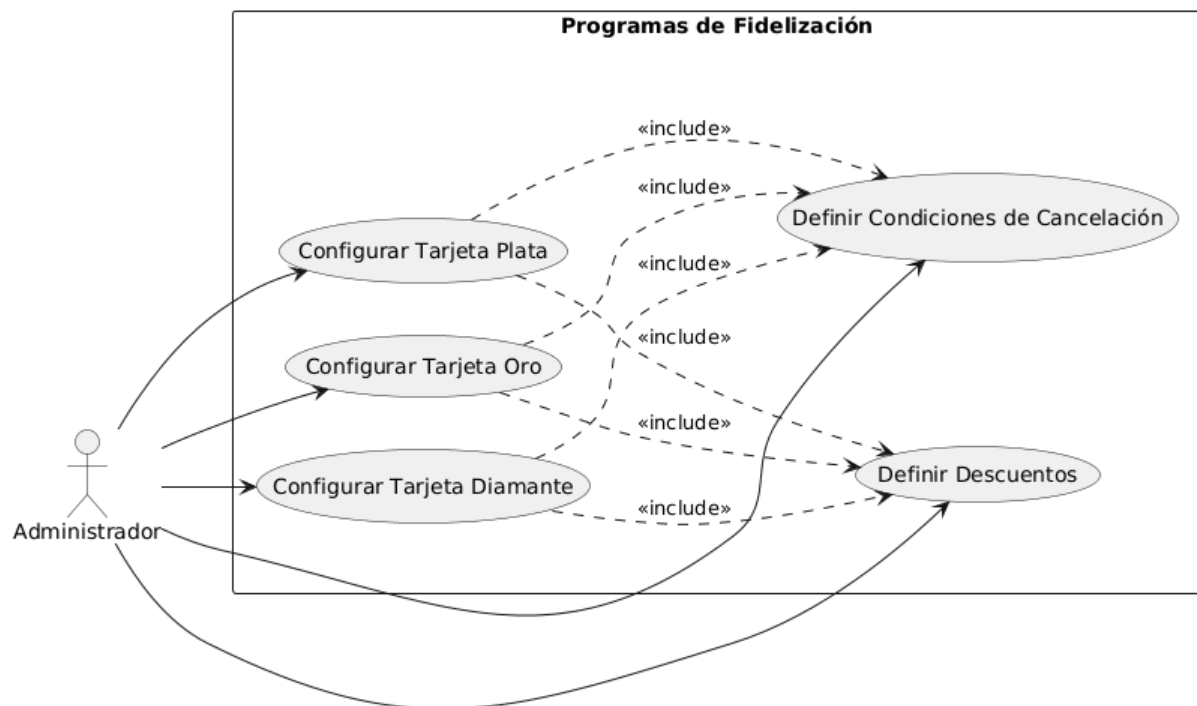
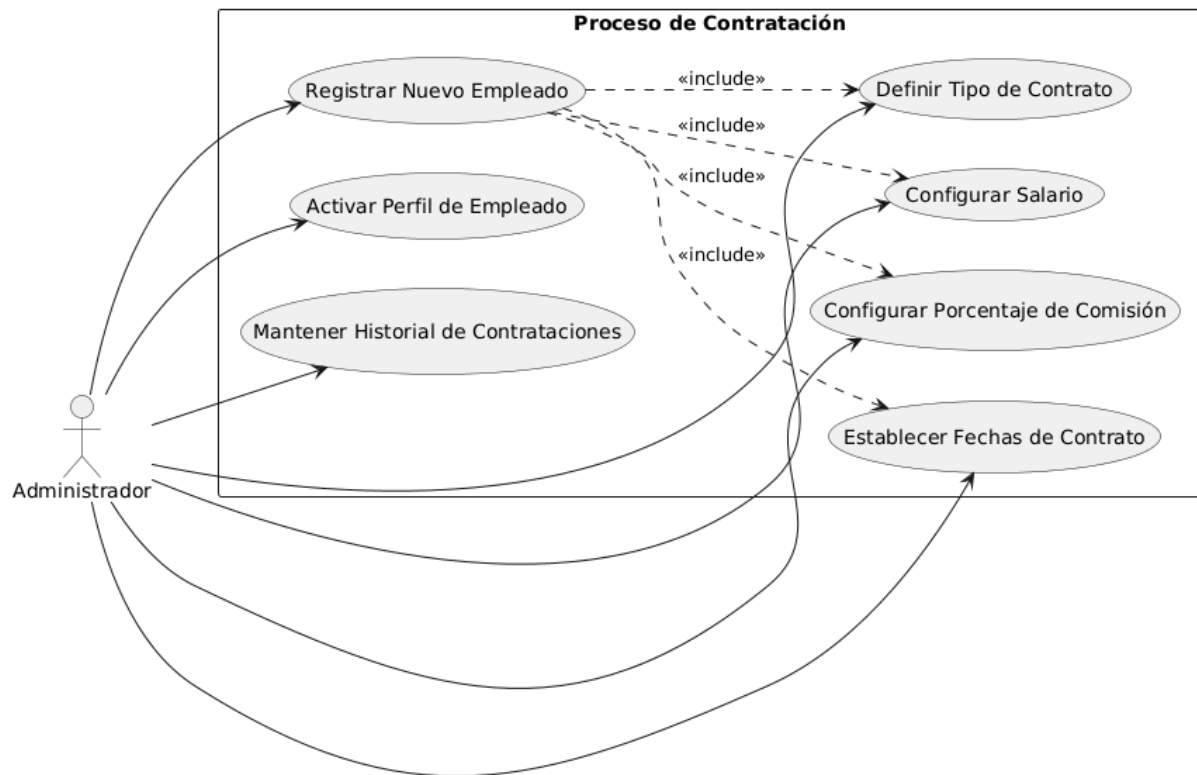


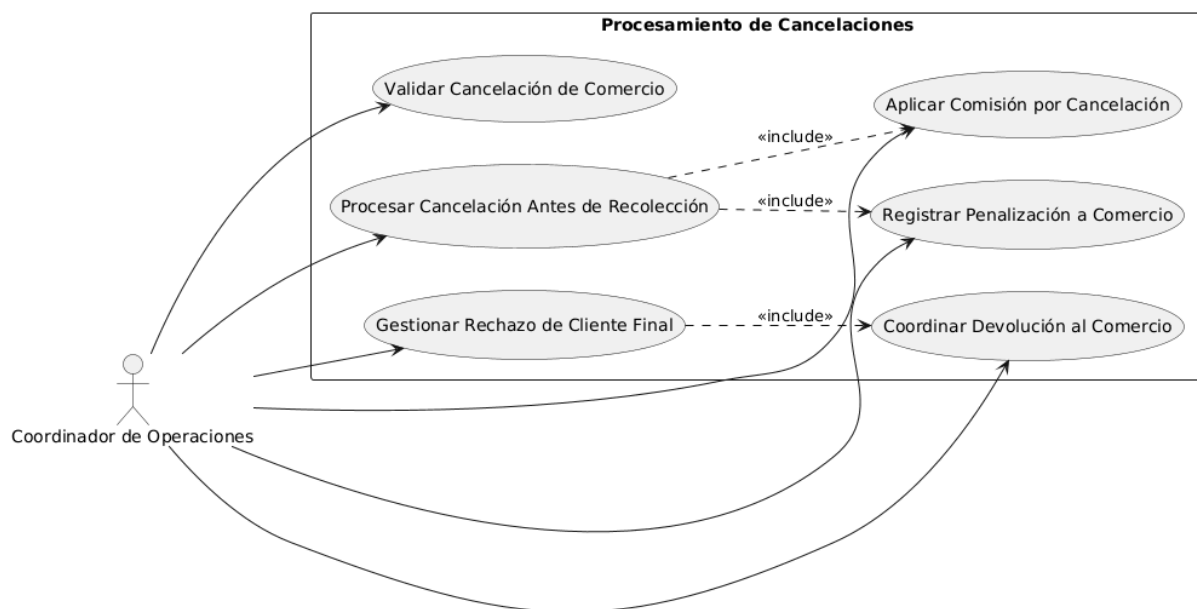
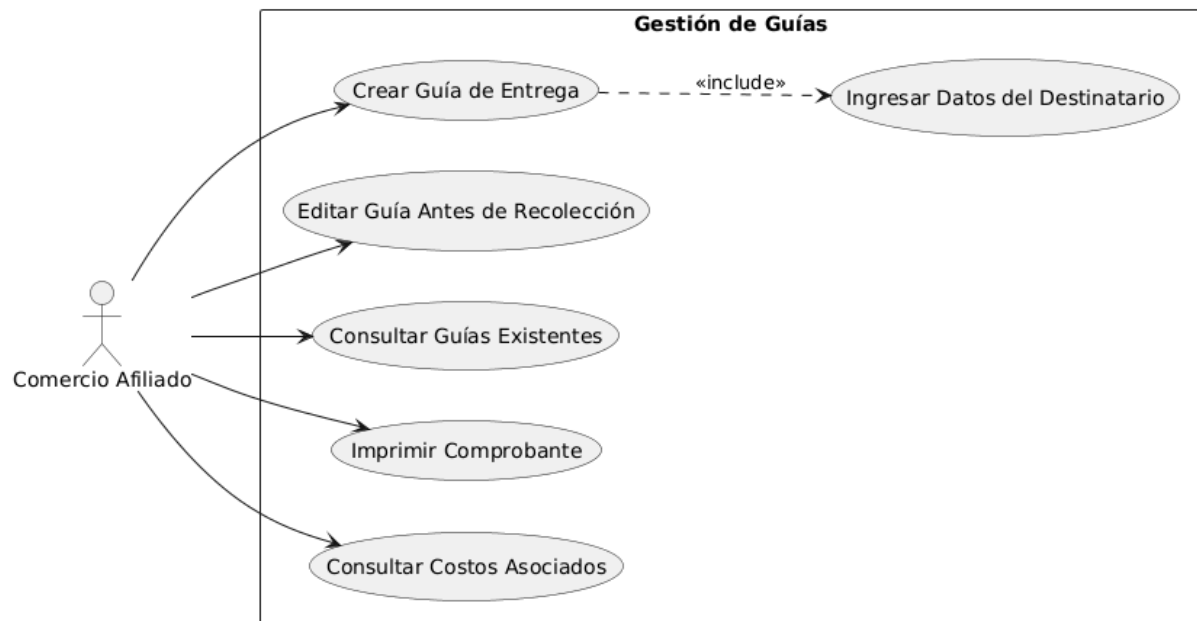


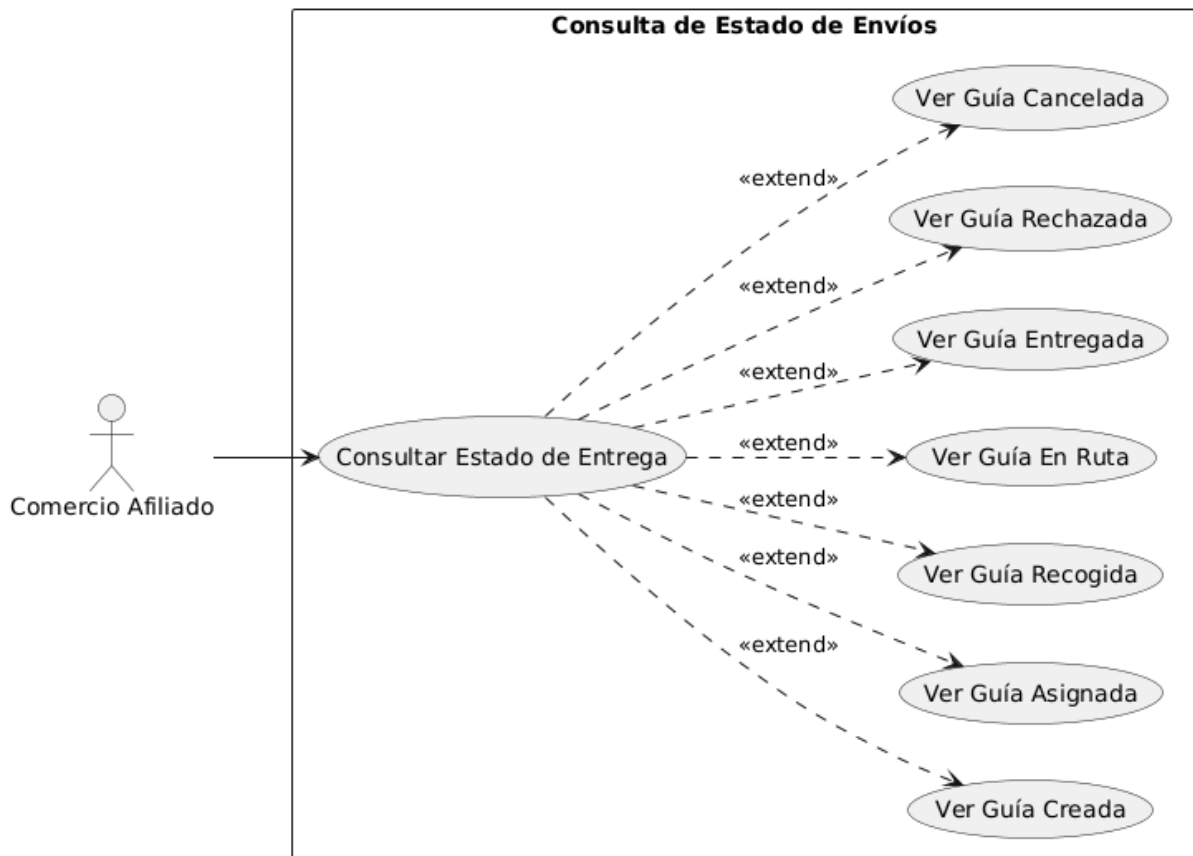
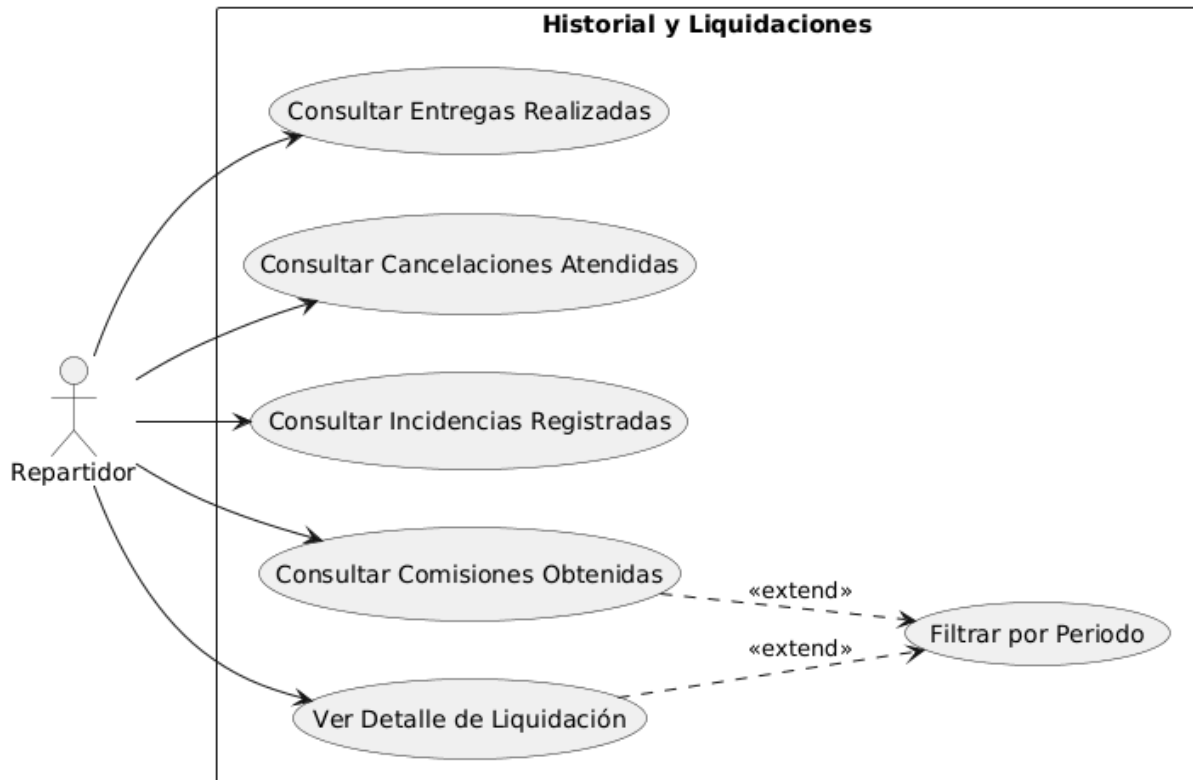




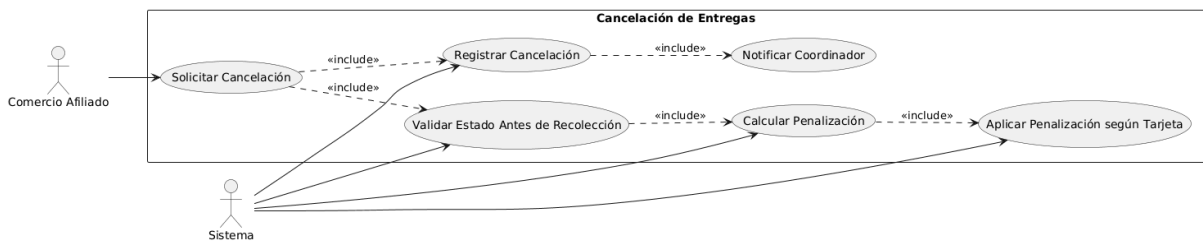
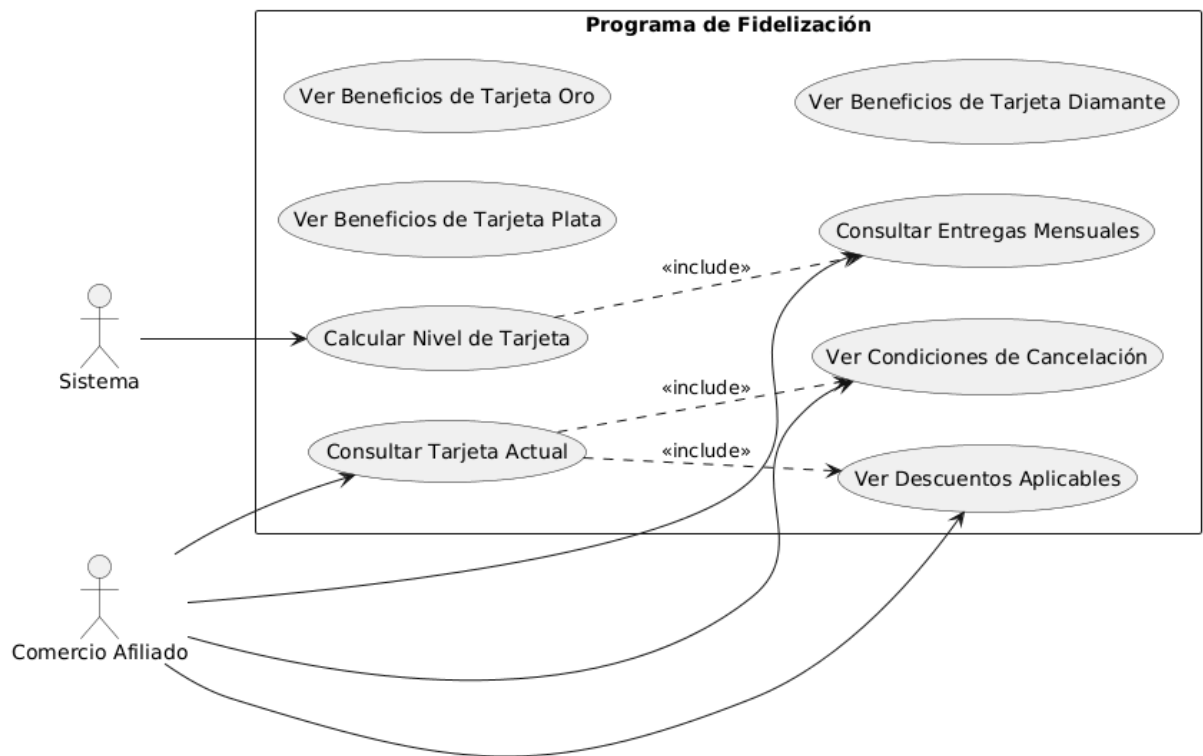
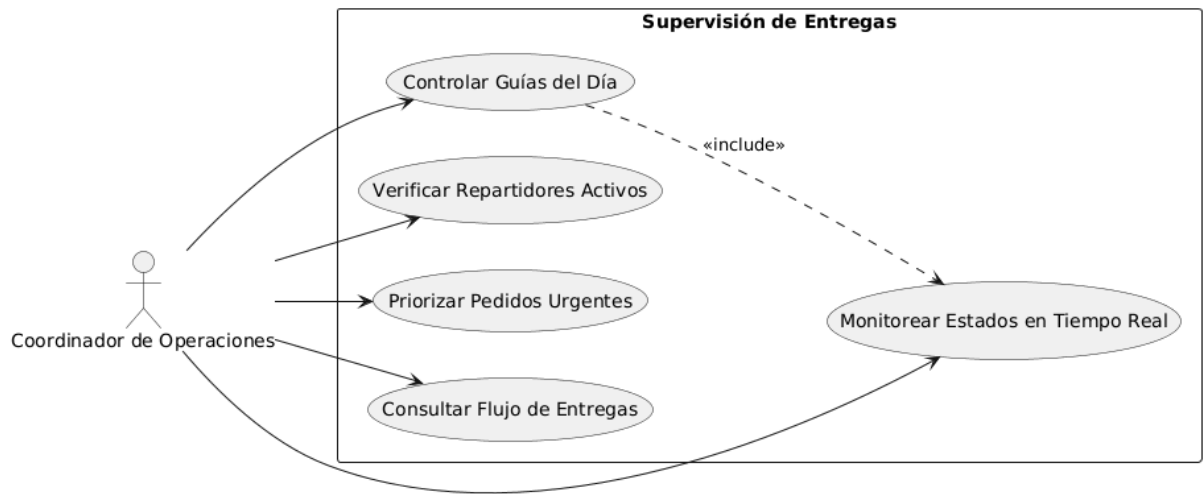


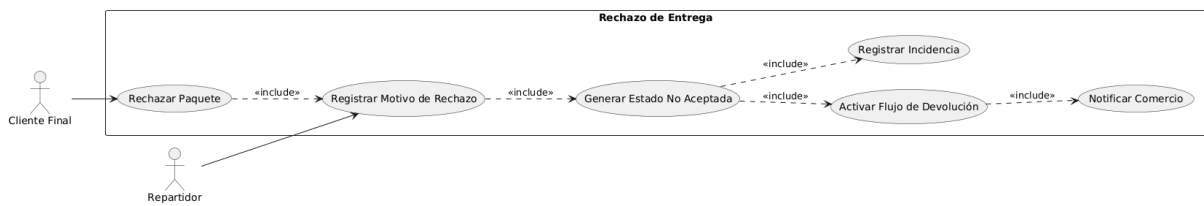
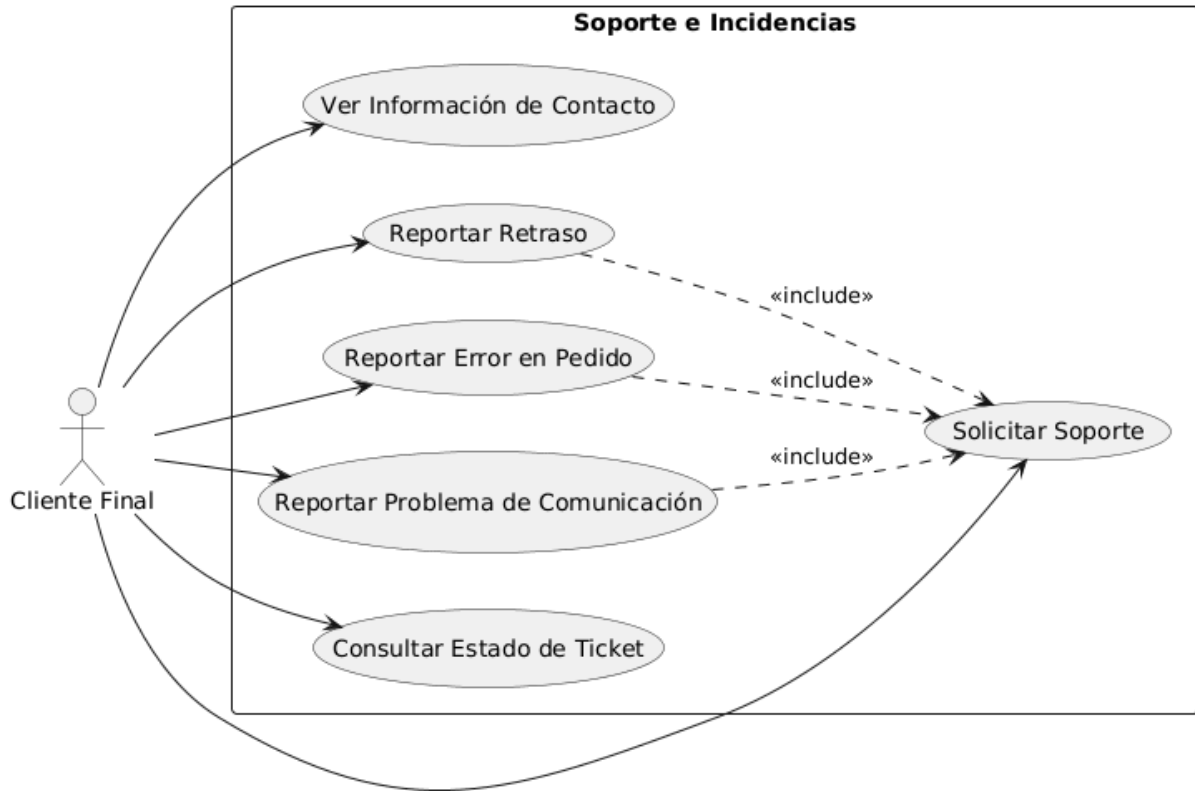
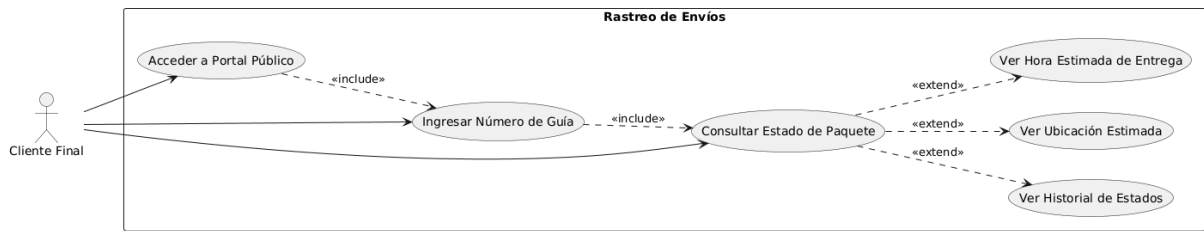


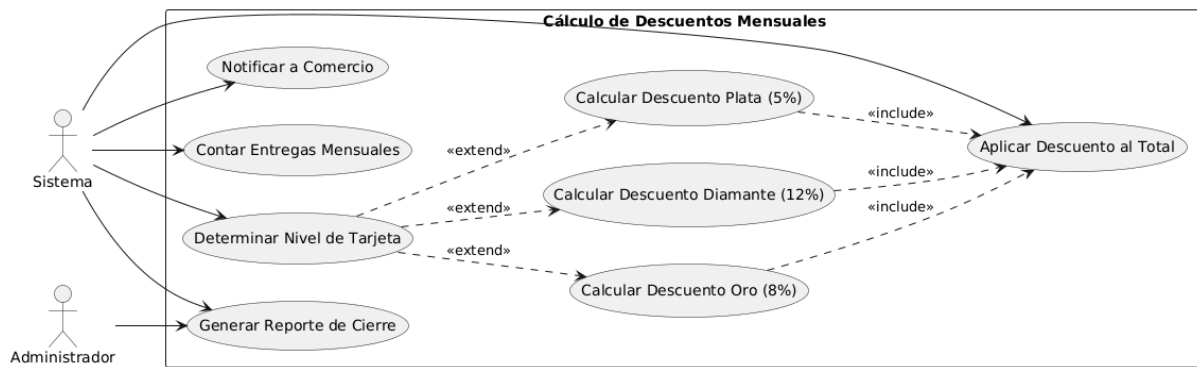
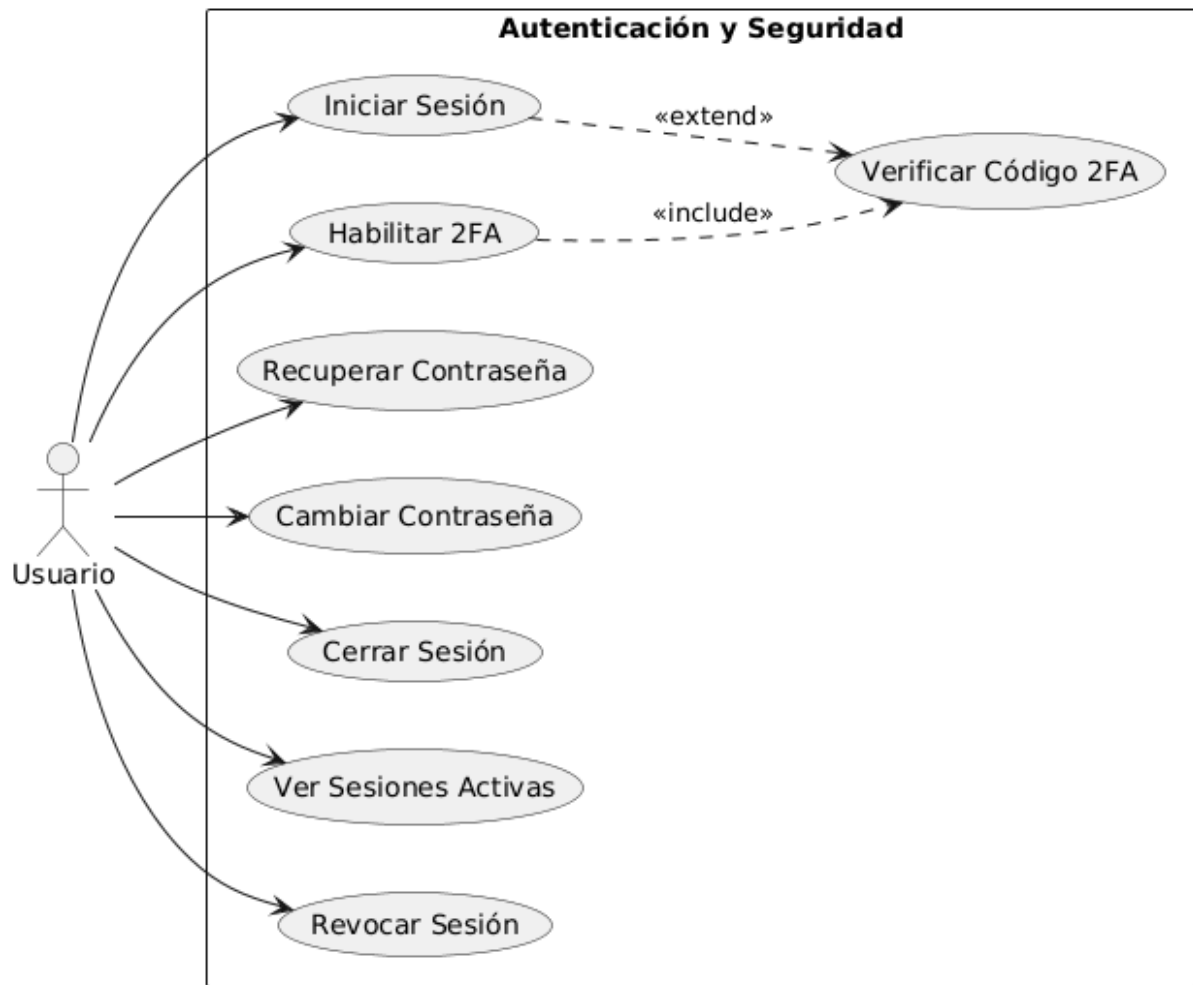




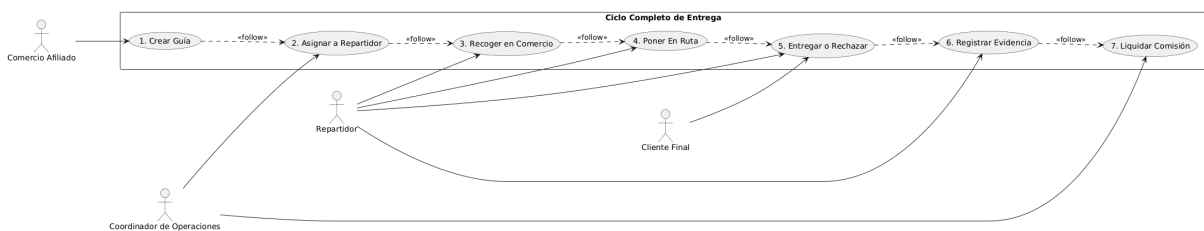
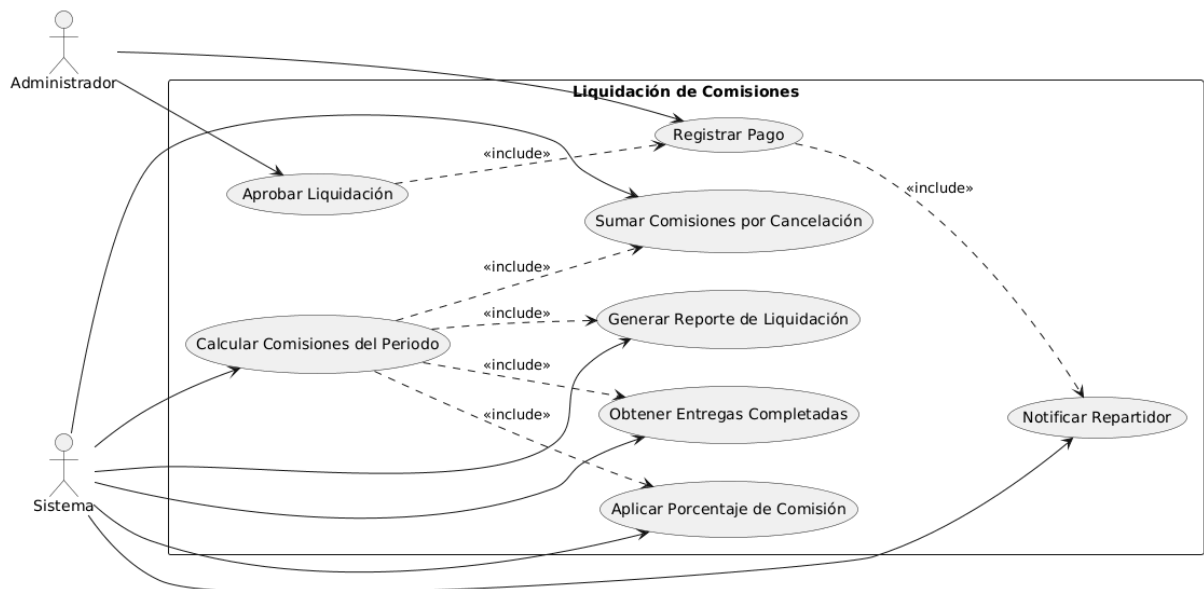
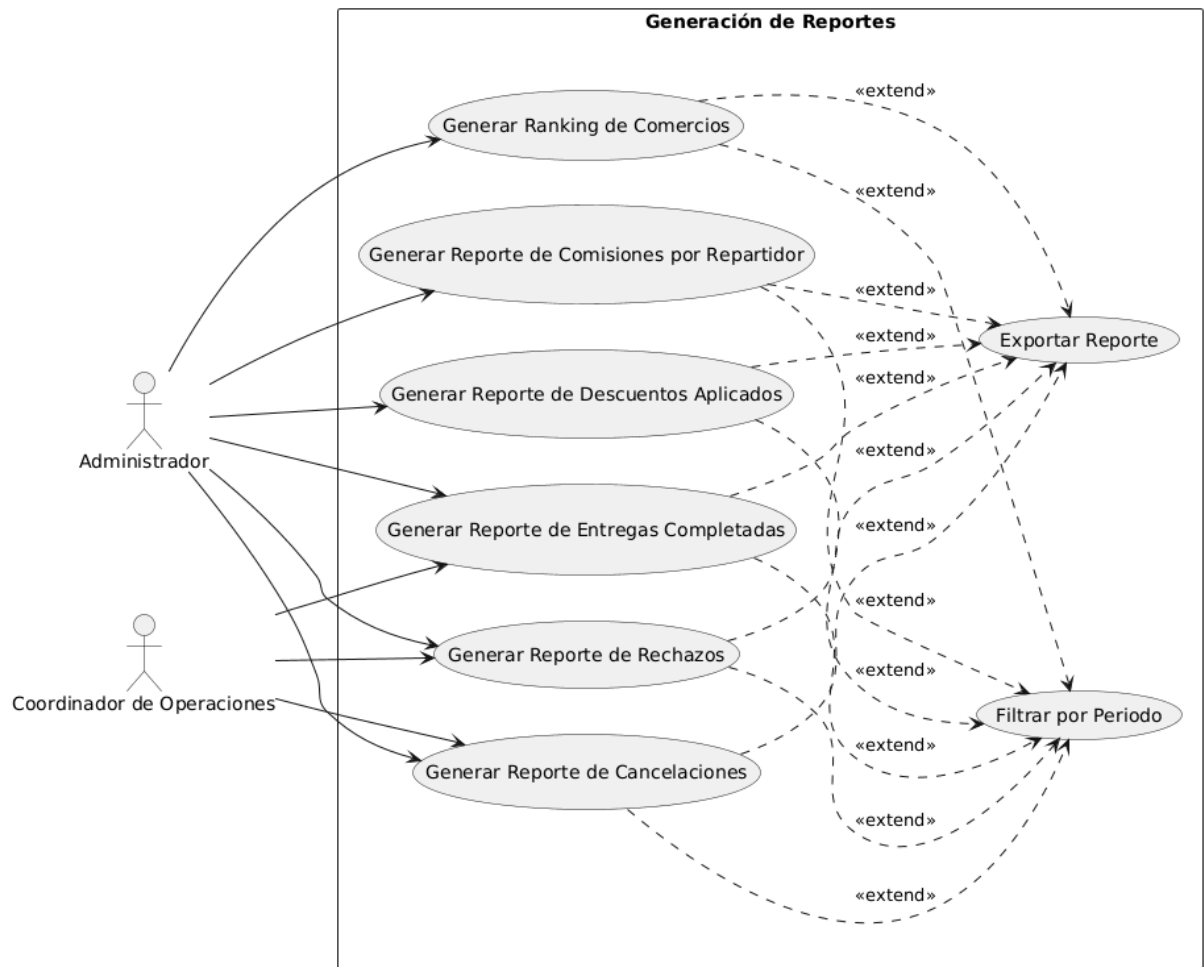












## ANEXOS

- Carpeta de Drive:  
<https://drive.google.com/drive/folders/1DNokQQUopcnqjJjDE7KTUaJowKf6JvqW?usp=sharing>