MANUAL TECNICO

SOFTWARE DE ADMINISTRACIÓN PARA LA GESTIÓN DE BENEFICIOS, POSTULACIONES Y ASISTENCIAS PARA BIENESTAR

BRAYAN DAVID RICO CERQUERA

JUAN DAVID RICO CERQUERA

SAIRA XIOMARA GUEVARA

SANTIAGO VALDEZ

CARLOS MAURICIO POLANCO REYES

CENTRO DE FORMACION SENA LA ANGOSTURA / ANALISIS Y DESARROLLO DE SOFTWARE FICHA 2500871

CAMPOALEGRE/HUILA

CONTENIDO

1. INTRODUCCION………………………………………………………………………………………………
   1. REQUERIMIENTOS TECNICOS………………………………………………………………..
   2. HERRAMIENTAS UTILIZADAS EN EL DESARROLLO…………………………………….
      1. Control de Versiones…………………………………………………………………………………….
      2. Desarrollo del lado del servidor……………………………………………………………………….
      3. Diseño de la interfaz de Usuario………………………………………………………………………….
2. INSTALACION DEL WEB SERVER…………………………………………………………
3. CONFIGURACION DEL SOFTWARE……………………………………………………..
   * 1. Control de Versiones………………………………………………………………………………….
     2. Desarrollo del lado del servidor…………………………………………………………………….
     3. Diseño de la interfaz de Usuario…………………………………………………………………….
4. DIAGRAMAS………………………………..……………………………………………………..
   * 1. Casos de Uso………………………………………………………………………………
     2. Clases…………………………………………………………………………………….
     3. Secuencias…………………………………………………………………………………….
5. DIAGRAMAS………………………………..……………………………………………………..
6. DICCIONARIO DE DATOS……………………………………………………………………..
7. PROTOTIPOS DE PANTALLAS DEL APLICATIVO

**INTRODUCCIÓN**

El módulo BIENESTAR se integra a SICEFA y está destinado a la entidad denominada "Bienestar al Aprendiz". Sus objetivos principales incluyen la gestión de convocatorias, postulaciones de los aprendices, administración de beneficios, gestión de rutas de transporte, así como la toma, consulta y registro de asistencias.

**Objetivos generales:**

Desarrollar un software para el centro de formación Sena Agroindustrial que permita escanear códigos de barras, almacenar información, gestionar la asistencia y registrar a los aprendices vinculados a los diferentes apoyos brindados por el centro de formación (transporte, alimentación, entre otros).

Este software también se encargará de gestionar convocatorias y postulaciones de aprendices a los apoyos ofrecidos por Bienestar, además de manejar información sobre aprendices, buses, rutas y conductores para mejorar el beneficio del servicio de transporte.

**Objetivos específicos:**

1. Diseñar un software que permita gestionar convocatorias, especificando fechas de duración y cupos disponibles para los beneficios.
2. Sistematizar las postulaciones de los aprendices a través de convocatorias, permitiéndoles seleccionar los apoyos que desean recibir.
3. Gestionar los diferentes apoyos, definiendo nombres y porcentajes asociados a cada beneficio.
4. Administrar y mantener información sobre conductores, buses y rutas de transporte.
5. Definir, asignar y gestionar los beneficios para los aprendices que han sido postulados.
6. Mantener un registro detallado de asistencias, así como especificaciones de postulaciones y beneficios otorgados.

Este módulo busca optimizar la eficiencia y transparencia en la gestión de apoyos y beneficios, brindando un sistema integral para el bienestar de los aprendices en el centro de formación Sena Agroindustrial.

* 1. REQUERIMIENTOS TÉCNICOS
* REQUERIMIENTOS MINIMOS DE HARDWARE
* Procesador :Core i3
* Memoria RAM :Mínimo 8 Gigabytes (GB)
* Disco Duro : 500Gb.
  1. HERRAMIENTAS UTILIZADAS EN EL DESARROLLO
     1. Control de Versiones

Durante el desarrollo del proyecto BIENESTAR en Laravel, utilizamos Git como sistema de control de versiones. El repositorio principal se encuentra alojado en GitHub bajo la rama "BIENESTAR". Este enfoque nos permitió gestionar de manera eficiente el código fuente, realizar seguimiento de cambios y colaborar de manera efectiva en el desarrollo del módulo.

* + 1. Desarrollo del Lado del Servidor

Para el desarrollo del lado del servidor, optamos por utilizar el framework Laravel. Laravel proporciona una arquitectura MVC robusta que facilita la organización del código y la implementación de funcionalidades esenciales. Además, aprovechamos las características incorporadas de Laravel, como Eloquent ORM para interactuar con la base de datos de manera elegante y eficiente.

* + 1. Diseño de la Interfaz de Usuario

En el proceso de diseño de la interfaz de usuario, empleamos Visual Studio Code como nuestro entorno de desarrollo principal. Utilizamos la plantilla de AdminLTE para construir la estructura base de la interfaz, aprovechando su diseño intuitivo y componentes listos para usar. Además, integramos Fontawesome para la gestión de iconos, mejorando la estética y la usabilidad de la interfaz.

La implementación de SweetAlert2 en la interfaz de usuario agregó una capa adicional de interactividad y experiencia de usuario. Utilizamos estas alertas personalizadas para notificar al usuario sobre acciones importantes de manera más amigable y visualmente atractiva.

La combinación de estas herramientas, junto con Laravel y Bootstrap, nos permitió lograr una interfaz de usuario coherente, receptiva y visualmente agradable para los usuarios finales.

1. INSTALACIÓN DEL WEB SERVER
   1. Se debe de instalar el entorno de desarrollo web llamado laragon se puede descargar a través del siguiente link <https://laragon.org/download/index.html>
   2. Se deben de instalar el gestor de dependencias denominado Composer que puede descargarse a través del siguiente link <https://getcomposer.org/download/>
   3. Se debe de instalar el sistema de control de versiones denominado git el cual se puede descargar a través del siguiente link <https://git-scm.com/downloads>
   4. Despues de integrado al repositorio de sicefa debe en una carpeta ejecutar los comandos para clonar el proyecto y tenerlo de manera local en su ordenador
2. CONFIGURACIÓN DEL SOFTWARE
   1. Despues de bajado el proyecto del repositorio debe ubicarse la carpeta del proyecto en la siguiente ruta partiendo de la carpeta laragon: \laragon\www\
   2. Se debe de eliminar la base de datos actual y crearla de nevo con el nombre de sicefa
   3. En la terminal apuntando al proyecto se debe ejutar un comando para correr las migraciones para la auditoria de datos: php artisan migrate --path=database/migrations. (ten en cuenta que la base de datos debe estar configurada en los parámetros DATABASE del archivo .env del proyecto laravel)
   4. Ejecutar las migraciones de todos los módulos con el comando php artisan migrations:run-modules
   5. En caso de no tener instalado SQLyog: descargar y descomprimir el siguiente archivo: https://drive.google.com/file/d/1d0gPxqI64L6CEEMnwFcfU0fhyV0pZSPw/view?usp =sharing Instala SQLyog y configura la conexión con el mismo puerto en que se ejecuta los servicios de mysql ya sea de XAMPP o laragon. Finalmente agrega la siguiente licencia para activar todas las características del producto: Nombre: Ultimate Clave: 60c1b896-7c22-4405-9f46-a6bce776ab36
   6. Descargamos el archivo .sql del siguiente enlace https://drive.google.com/drive/folders/1- QgV9p2Wla\_7SygabEGqc4gt\_zQshrCP?usp=sharing Abrimos la conexión del servidor de mysql en SQLyog y ubicamos la base de datos sicefa e importamos el archivo .sql descargado.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Una vez completada la línea en verde durante la ejecución del script cerramos la ventana de dialogo (click en Done) y la base de datos ha quedado lista.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

* 1. Ahora como último paso una vez configurada las migraciones y seeders necesarios del módulo lo podemos ejecutar con alguno de los siguientes comandos de acuerdo a su necesidad:

➢ Ejecutar solo migraciones del módulo X: php artisan module:seed X

➢ Ejecutar solo seeders del módulo X: php artisan module:seed X

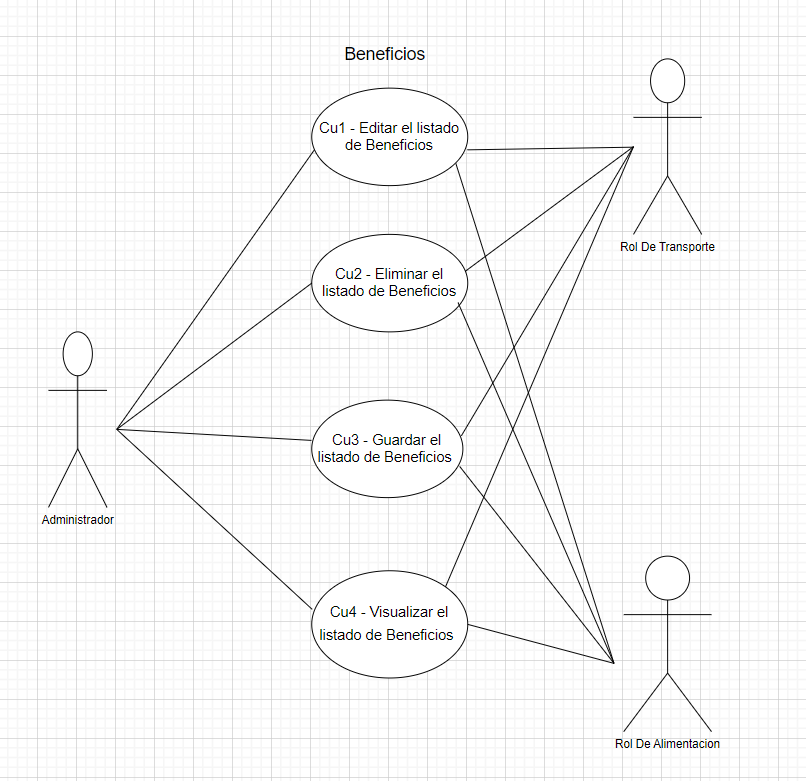
➢ Ejecutar solo migraciones y seeders del módulo X: php artisan

module:migrate --seed X

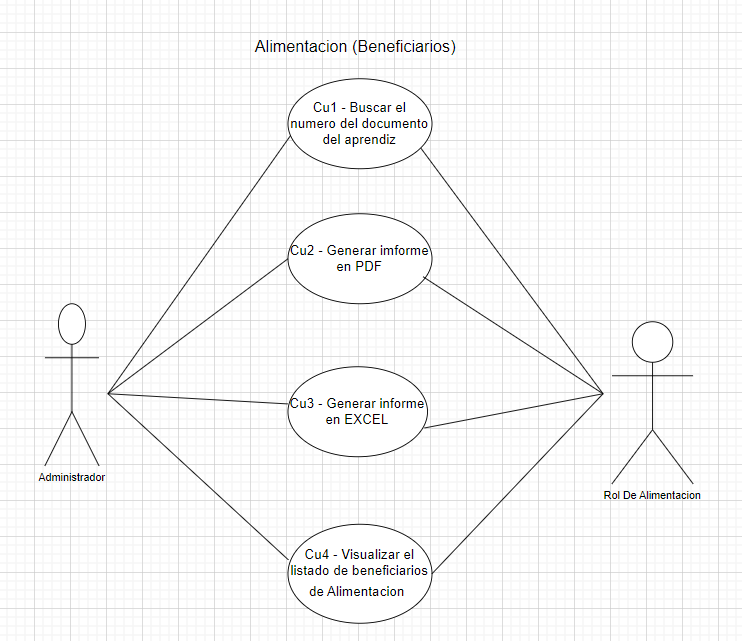
➢ Devolver migraciones ejecutadas del módulo X: php artisan module:migraterollback X

1. DIAGRAMAS
   * 1. Casos de Uso

Beneficios:



Beneficiario De Alimentación:

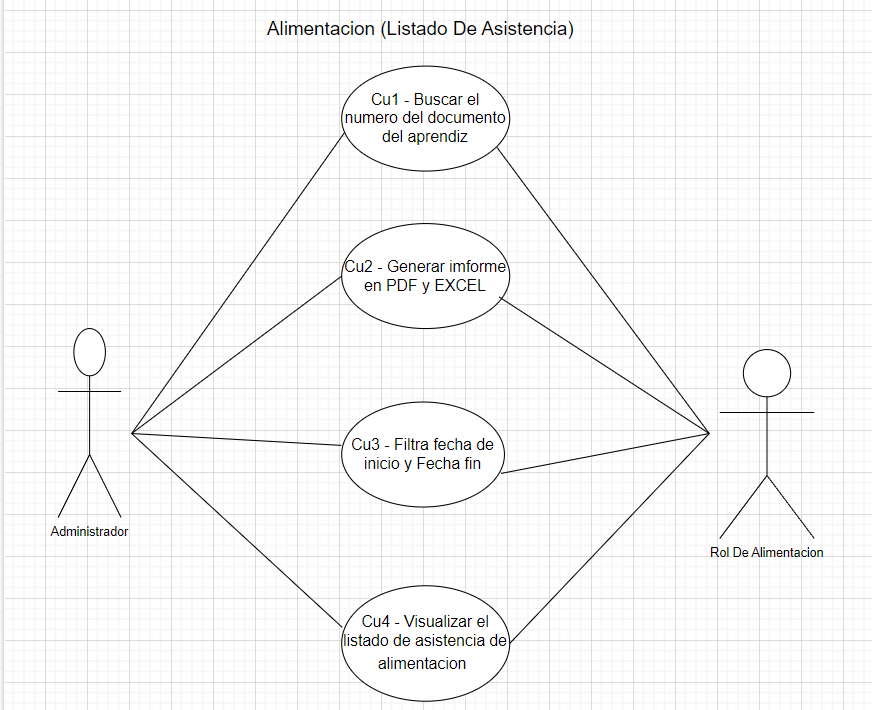


Tomar Asistencia Alimentación:

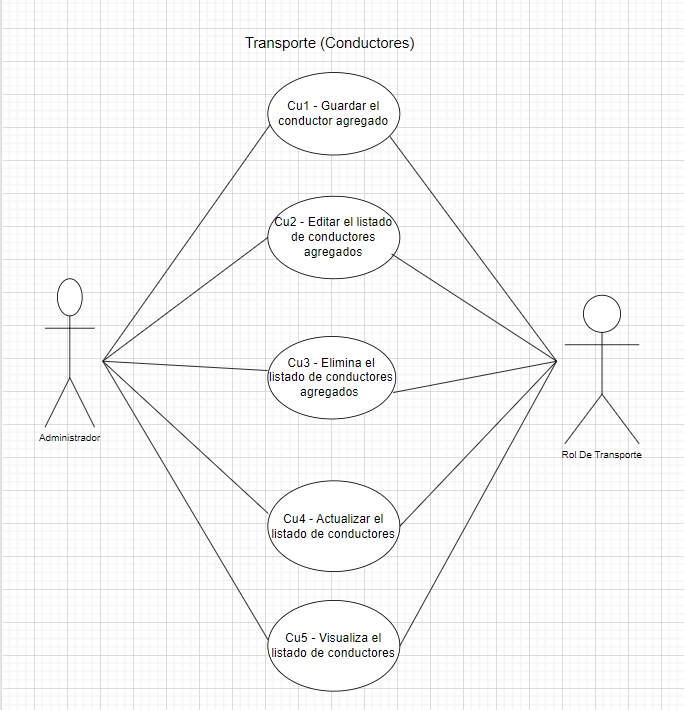
Diagrama

Descripción generada automáticamente

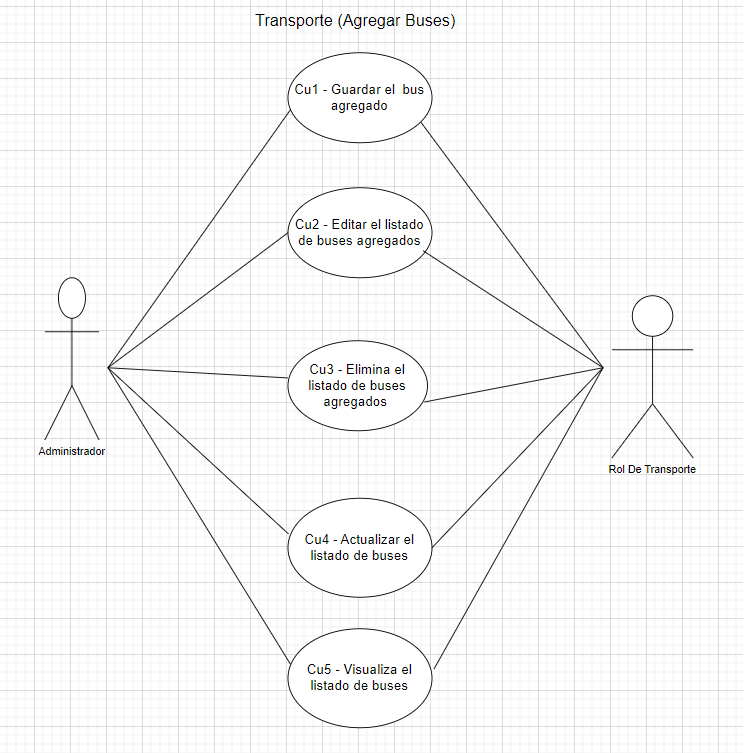
Listado De Asistencia Alimentación:

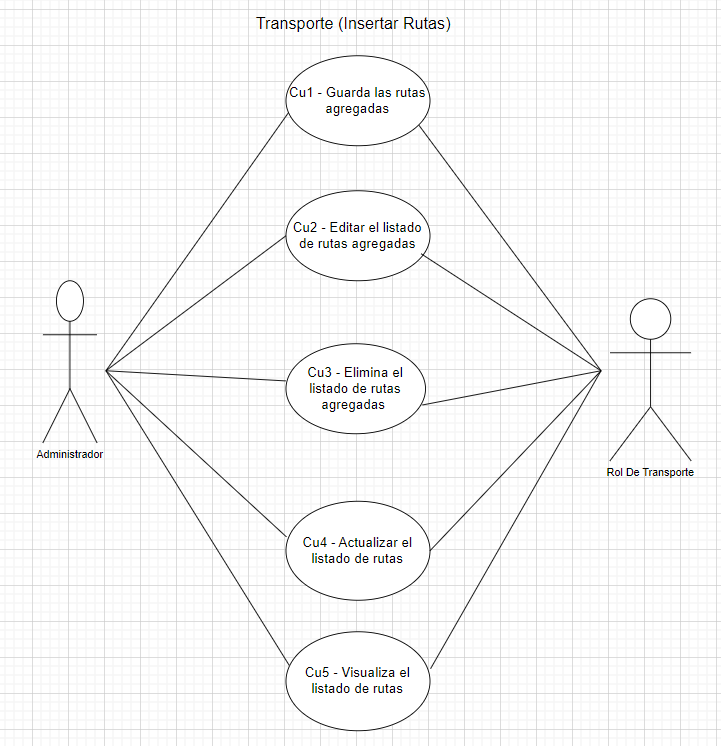


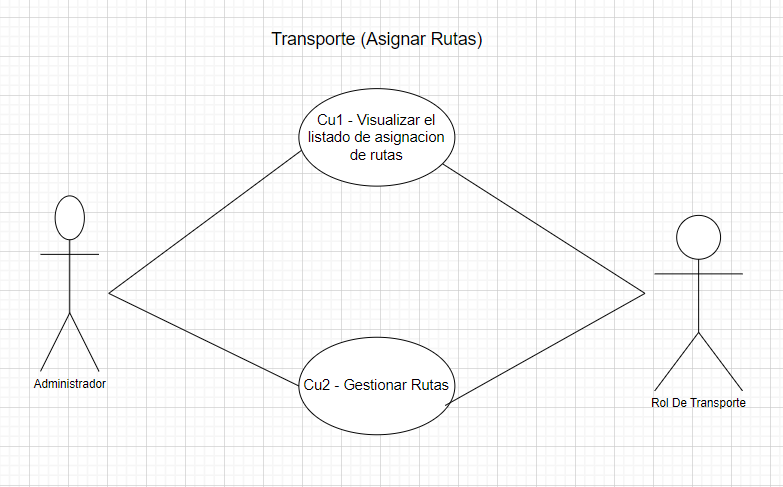
Conductores Transporte:

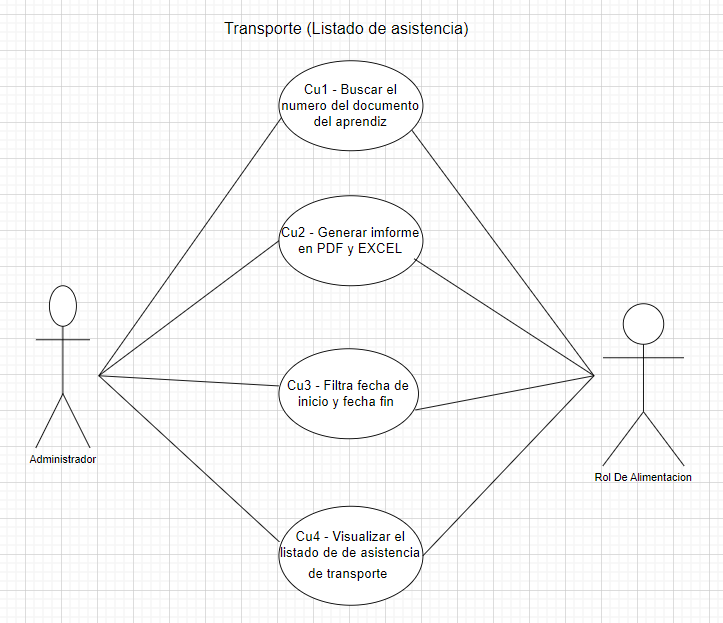


Buses:

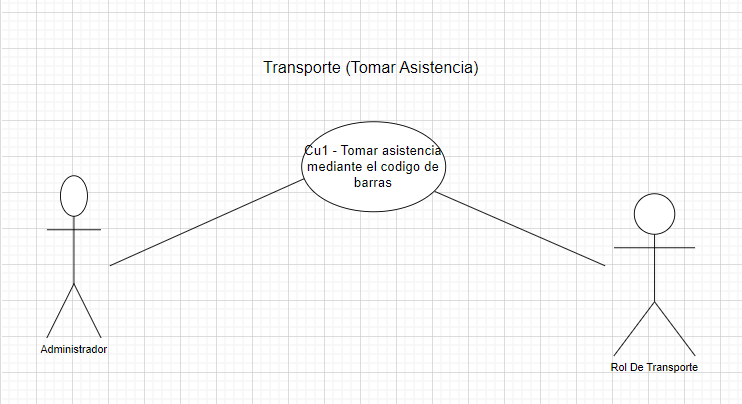


Rutas:

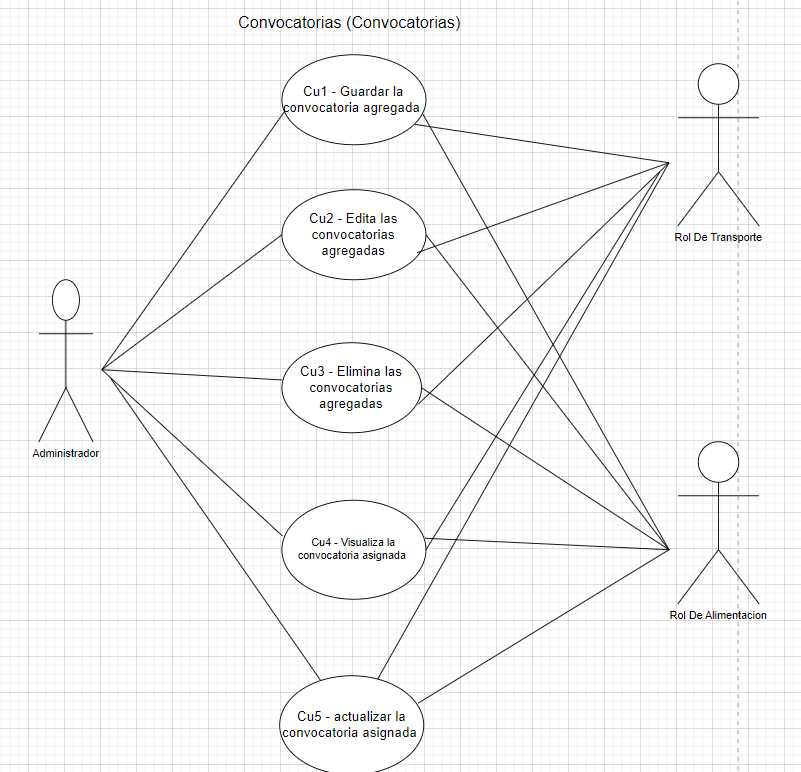
Asignar Rutas:

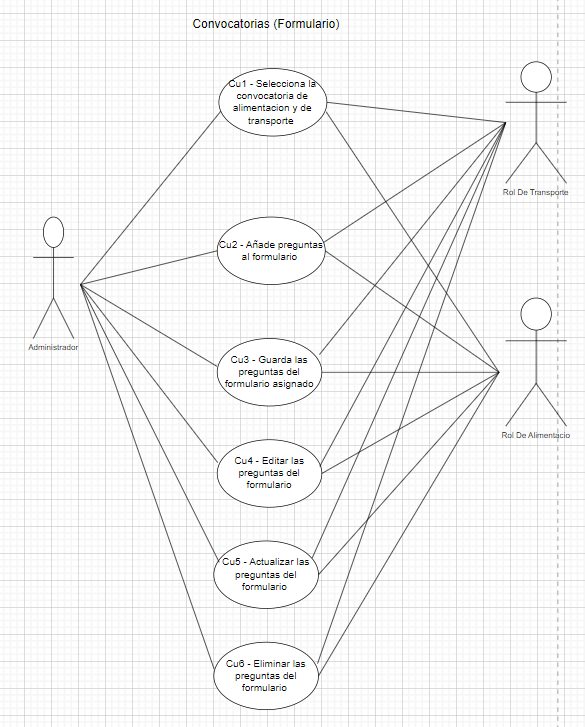
Tomar Asistencia:

Listados De Asistencia:

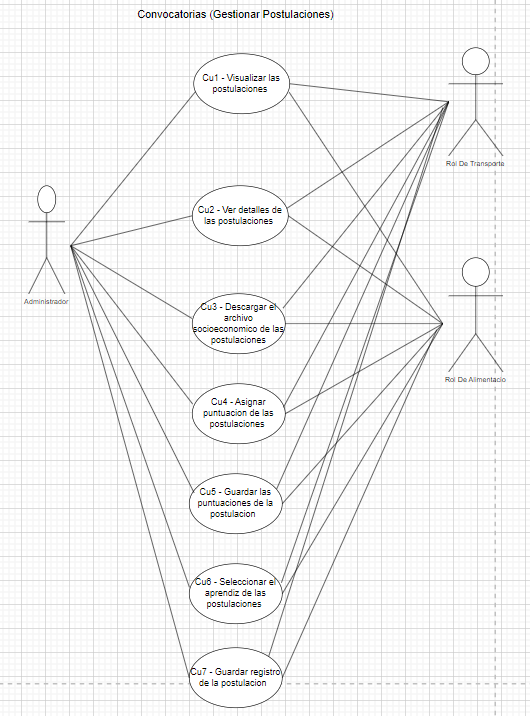


Convocatorias:



Formulario:

Gestionar Postulaciones:

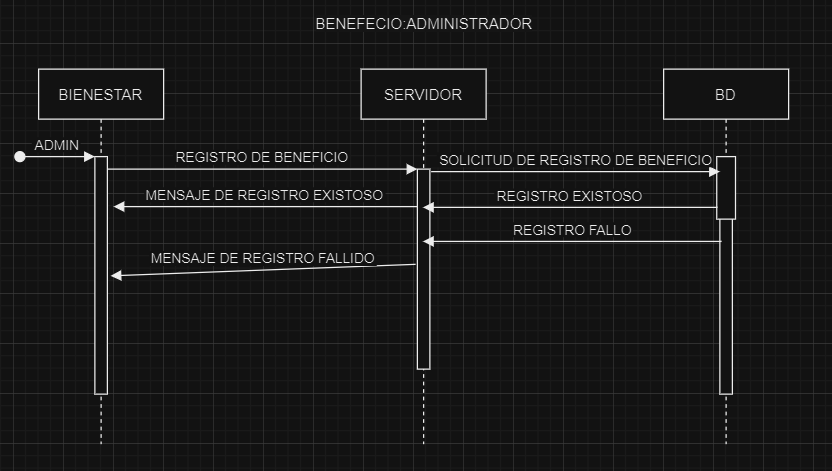


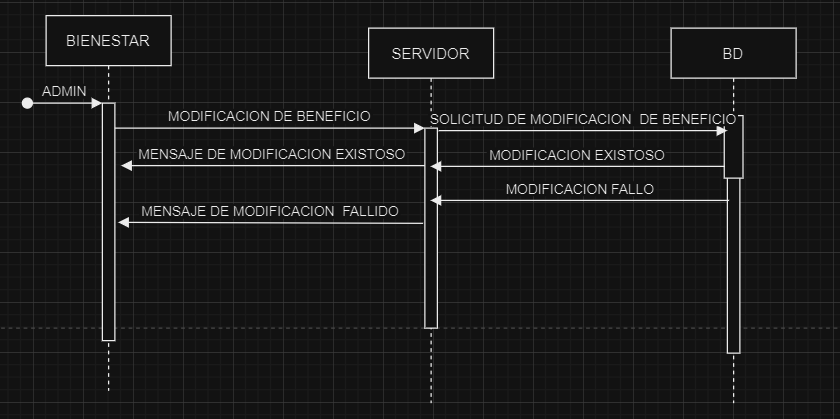
Presenta los casos de uso clave del módulo BIENESTAR.

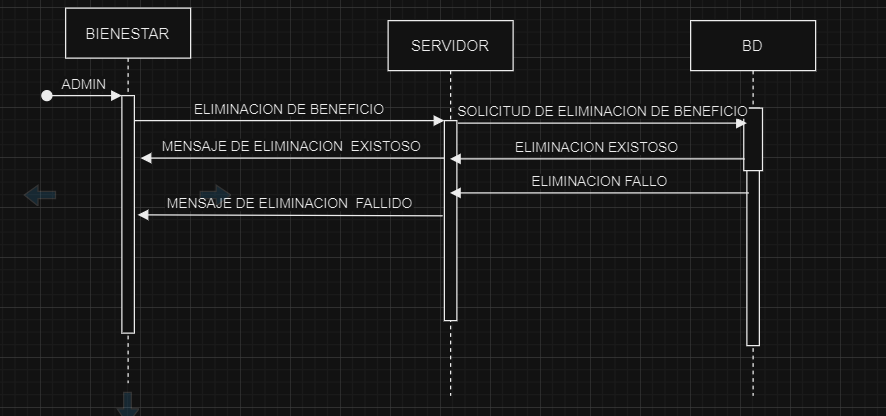
* + 1. Clases

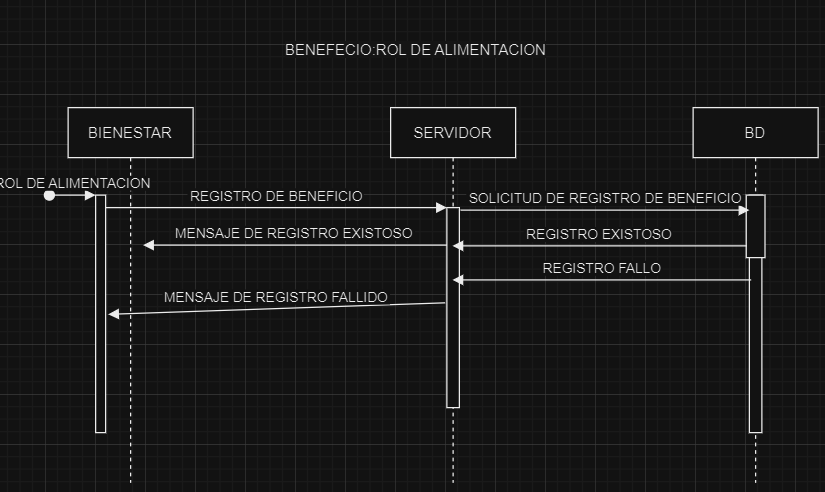
Proporciona diagramas de clases para mostrar la estructura del código.

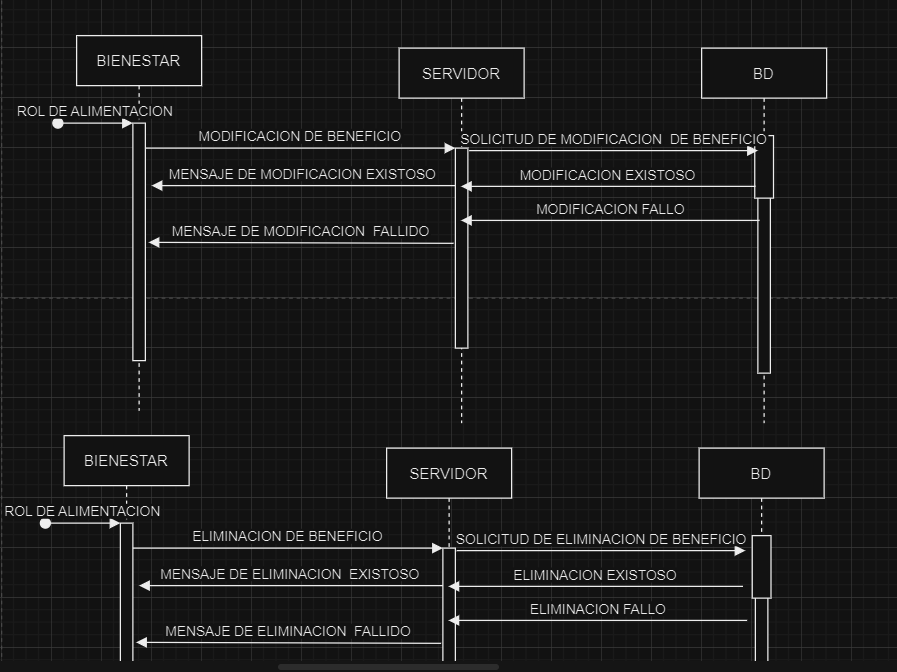
* + 1. Secuencias

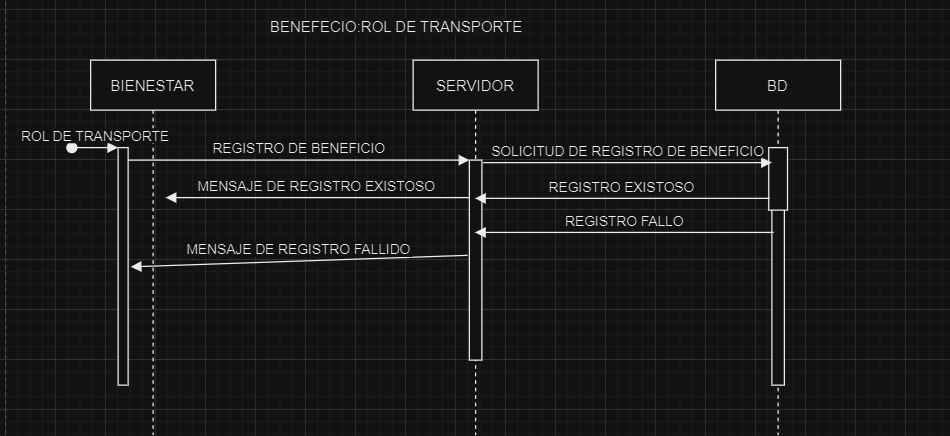


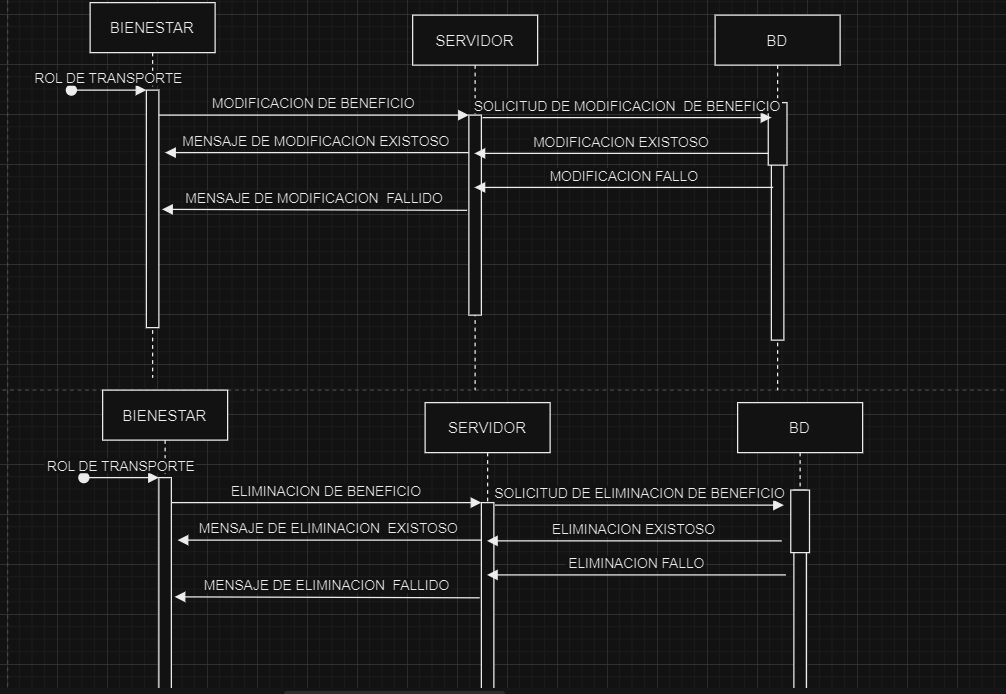


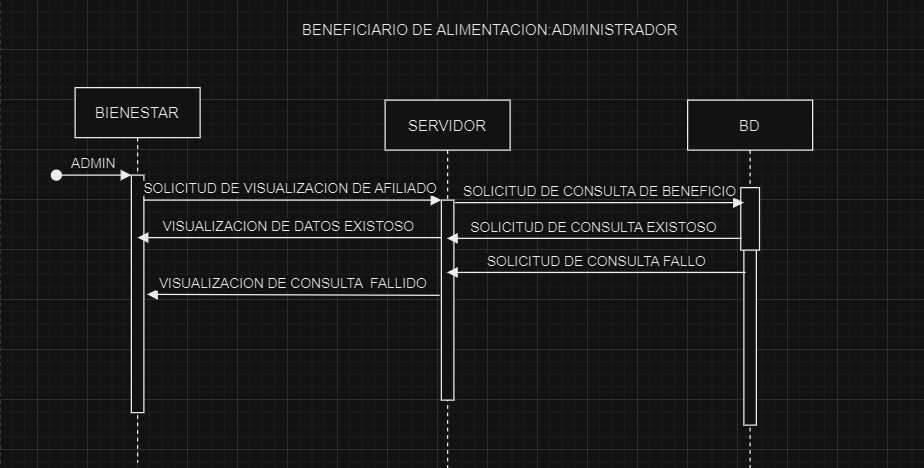


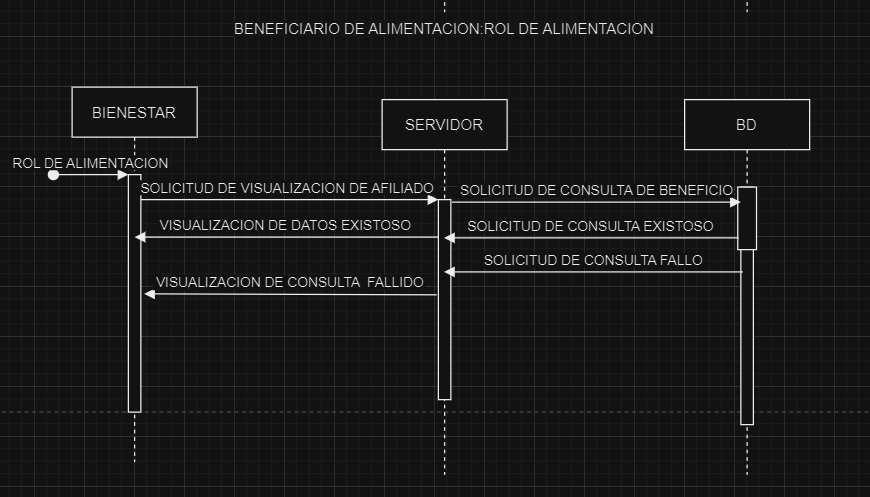


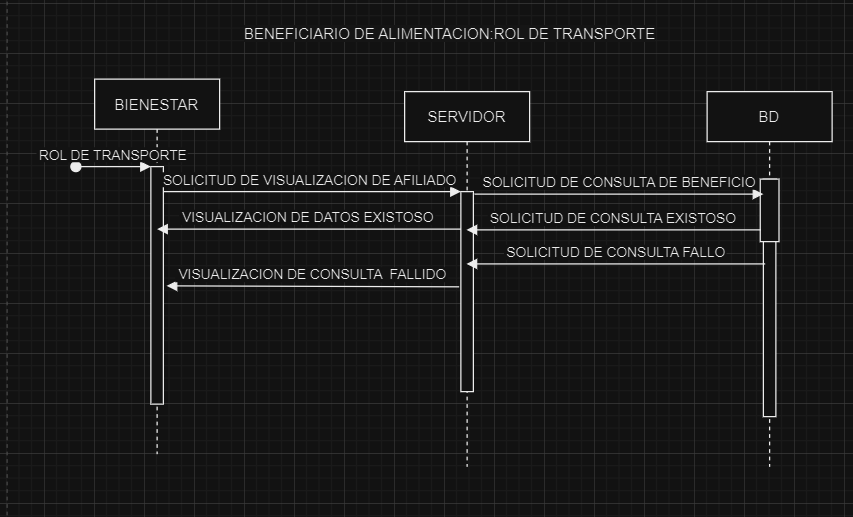


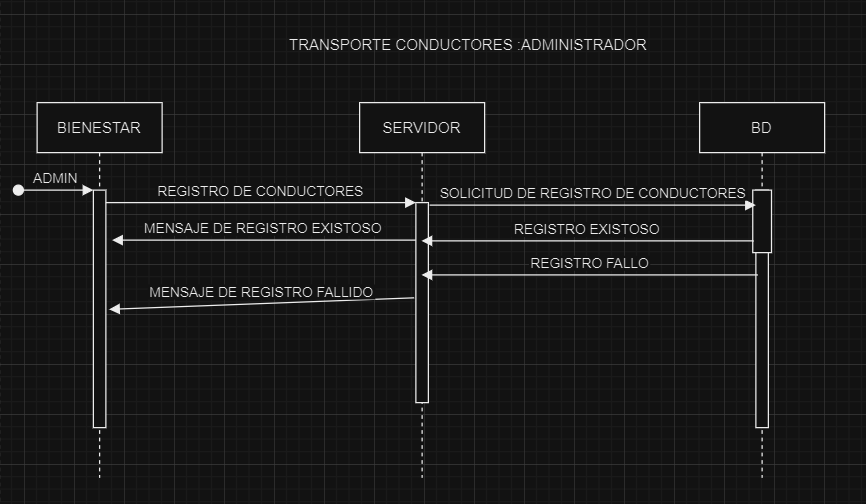


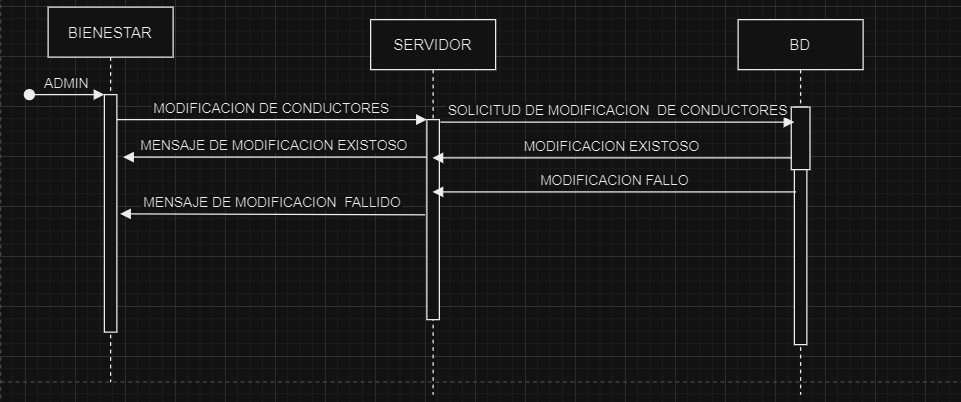


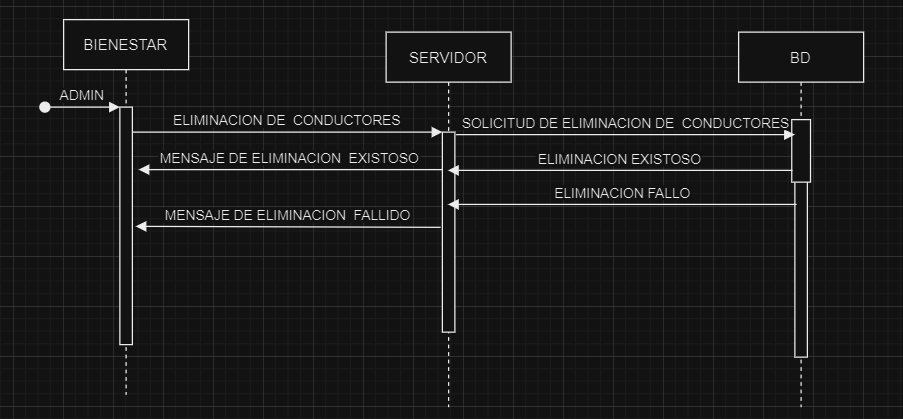


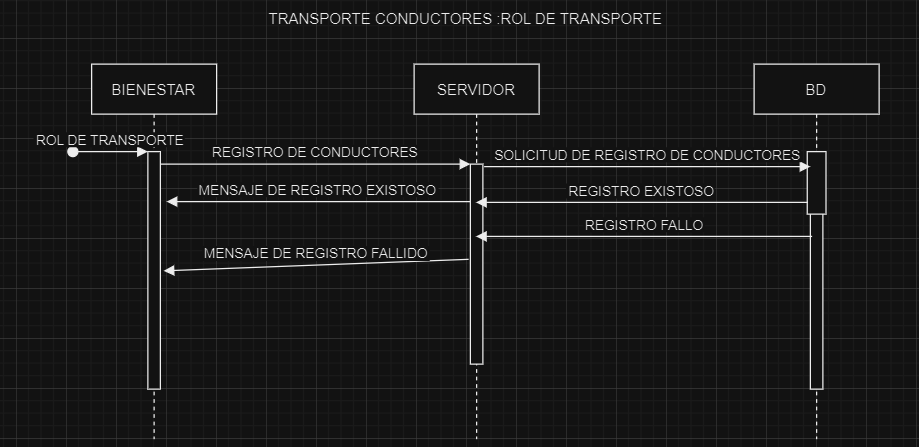


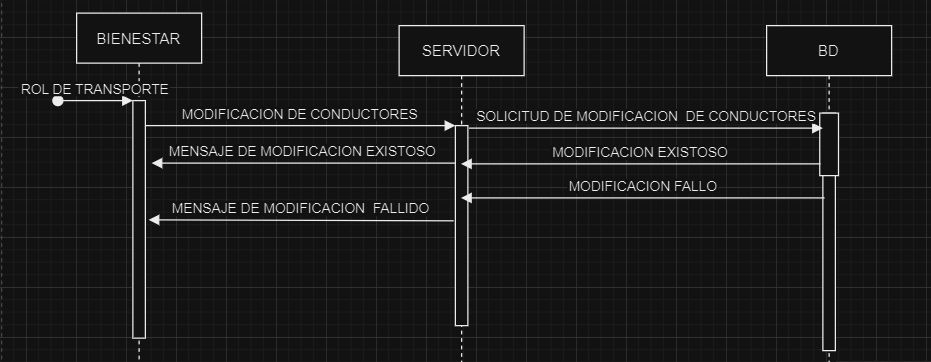


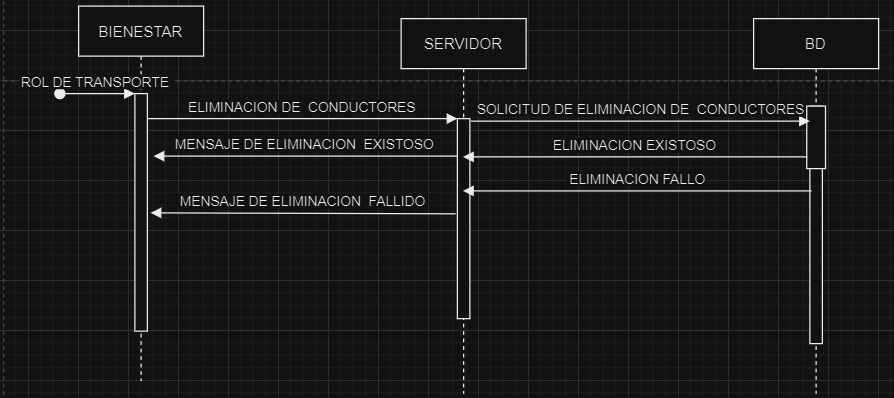


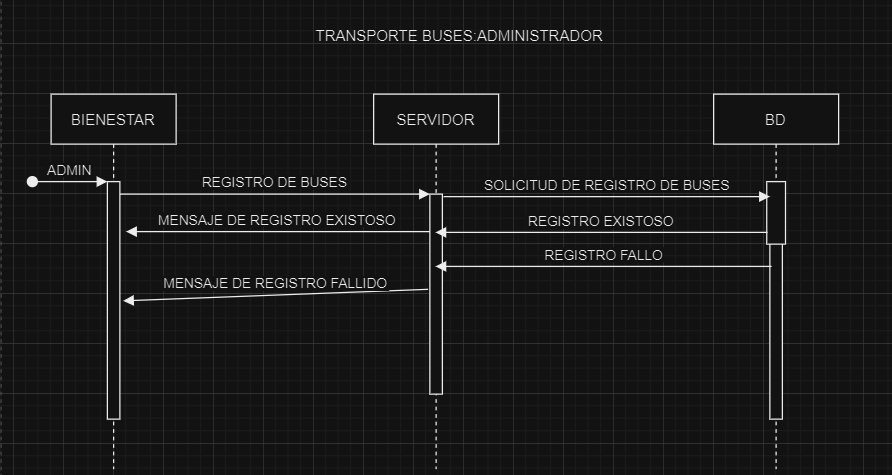


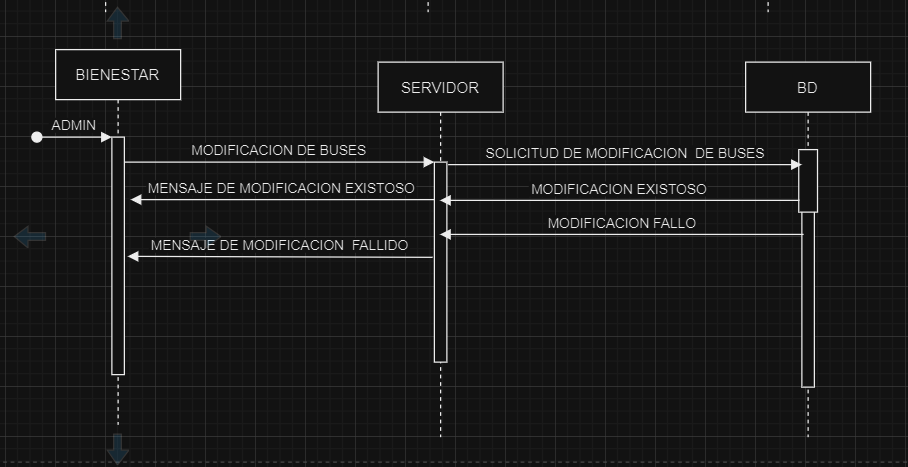


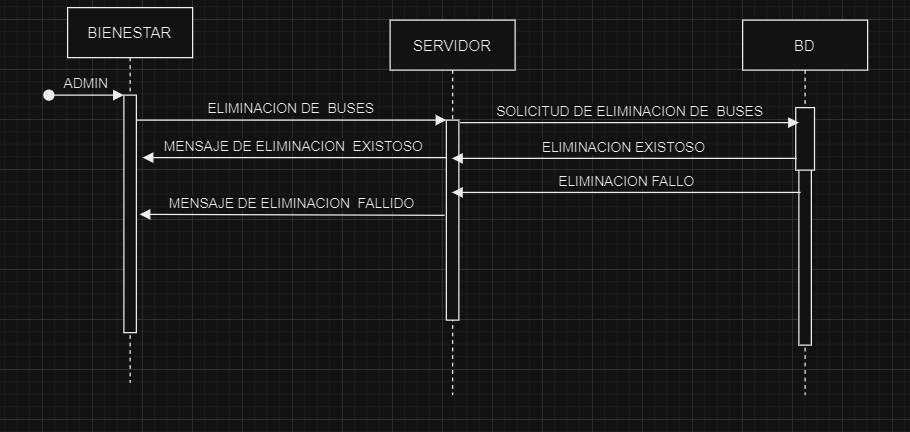


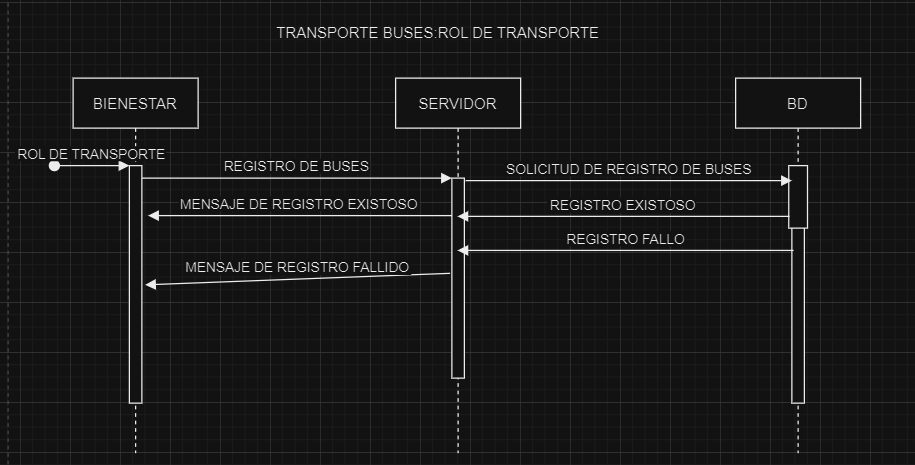


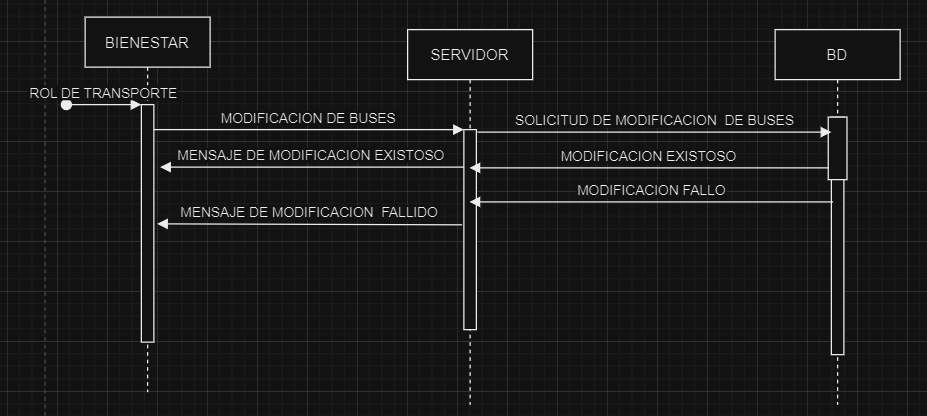


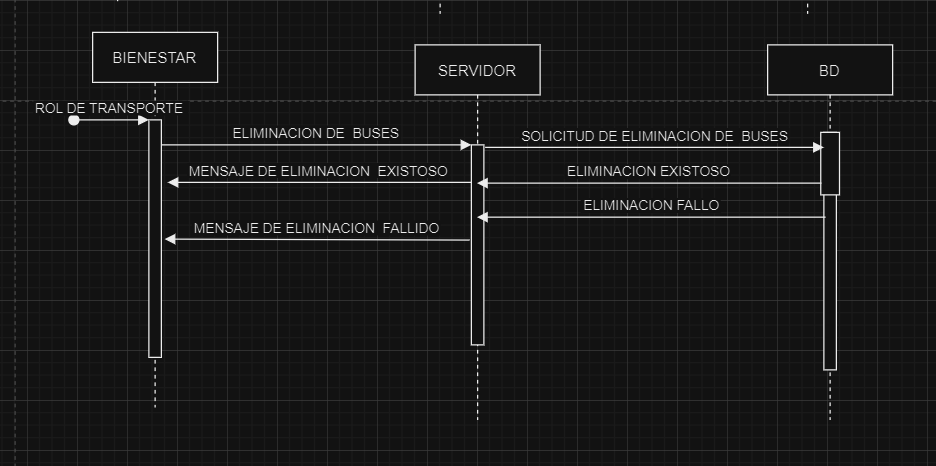












Muestra secuencias clave que representan la interacción entre componentes.

1. OTROS DIAGRAMAS

Puedes repetir la estructura de la sección 4 para incluir otros tipos de diagramas relevantes.

1. DICCIONARIO DE DATOS
2. PROTOTIPOS DE PANTALLAS DEL APLICATIVO

Incluye prototipos visuales de las pantallas del módulo. Puedes utilizar capturas de pantalla o herramientas de diseño para representar la apariencia final.

Recuerda adaptar cada sección según las peculiaridades de tu proyecto y las necesidades específicas de tu audiencia. ¡Buena suerte con tu manual técnico!