



GOBIERNO DE LA
CIUDAD DE MÉXICO

SECTEI - UPIITA IPN

Manual Técnico

SECTEI / 226 / 2019

Sistema de gestión integral para la ayuda en la
toma de decisiones en el Metrobús

2021

Contenido

- | | |
|-------------------------|--------------------------|
| 01. Introducción | 05. Estado del arte |
| 02. Información general | 06. Marco teórico |
| 03. Objetivos | 07. Análisis del sistema |
| 04. Justificación | 08. Desarrollo |
-

Índice

INTRODUCCIÓN DEL SISTEMA

SECCIÓN 1

Introducción

| | |
|----------------------------|----|
| Metrobús | 09 |
| Unidades de Transporte | 10 |
| Estaciones y Mantenimiento | 11 |
| Operaciones | 13 |
| Planteamiento del problema | 14 |

SECCIÓN 2

Información general

| | |
|----------------------------|----|
| Formatos iniciales | 16 |
| Unidades de Transporte | 17 |
| Estaciones y Mantenimiento | 18 |
| Formatos finales | 19 |
| Unidades de Transporte | 19 |
| Estaciones y Mantenimiento | 20 |

SECCIÓN 3

Objetivos

| | |
|------------------------|----|
| Objetivo general | 22 |
| Objetivos particulares | 22 |

SECCIÓN 4

Justificación

Justificación

24

SECCIÓN 5

Estado del arte

| | |
|------------------------------------|----|
| Sistema de gestión docuental | 26 |
| Sistema de gestión de usuarios | 26 |
| Metrobús y sistemas de gestión | 27 |
| Sistema gestor de inventarios | 27 |
| Metrobús y sistemas de inventarios | 28 |
| Sistemas CRUD | 28 |

26
26
27
27
28
28

SECCIÓN 6

Marco teórico

| | |
|--------------------------------------|----|
| Sistema | 30 |
| Sistema de información | 30 |
| Aplicación web | 31 |
| Base de datos | 31 |
| Sistema de gestión de bases de datos | 32 |
| Usuarios de bases de datos | 32 |
| Bases de datos MariaDB | 33 |
| Arquitectura cliente servidor | 33 |

30
30
31
31
32
32
33
33

Análisis del sistema

MANUAL DEL SISTEMA

| | |
|---------------------------|----|
| Lenguajes de programación | 34 |
| Javascrip | 34 |
| HTML5 / CSS | 36 |
| Framework | 38 |
| Node.js | 38 |
| React | 38 |
| Express JS | 39 |

| | |
|----------------------------------|----|
| Metodologías | 41 |
| Métricas y modelo de estimación | 42 |
| Modelo de estimación cocomo | 44 |
| Análisis de requerimientos | 45 |
| Requerimientos funcionales | 45 |
| Requerimientos no funcionales | 46 |
| Actores del sistema | 46 |
| Reglas del negocio | 46 |
| Análisis de factibilidad | 46 |
| Análisis de factibilidad técnica | 47 |

| | |
|---|----|
| Análisis de factibilidad operativa | 47 |
| Análisis de factibilidad económica | 47 |
| Análisis de riesgos | 49 |
| Tipos de riesgos | 49 |
| Planes de contingencia | 50 |
| Análisis de software | 53 |
| Análisis de aplicaciones web | 53 |
| Frameworks para aplicaciones web | 53 |
| Base de datos | 56 |
| Seguridad | 58 |
| Sistema de autenticación | 59 |
| Control de cuentas | 59 |
| Codificación de imágenes | 59 |
| Control del acceso de red | 60 |
| Gestión de credenciales | 60 |
| Gestión del sistema operativo | 60 |
| Encriptación de passwords por medio de bcrypt | 61 |

SECCIÓN 8

Desarrollo

| | |
|---|----|
| Diseño técnico del sistema de información | |
| Selección de software | 00 |

| | |
|---|----|
| Modelo de datos | |
| Unidades de transporte | |
| Diagrama Entidad-Relación | 00 |
| Casos de Uso | 00 |
| Diagrama de Actividades | 00 |
| Estaciones y mantenimiento | |
| Diagrama Entidad-Relación | 00 |
| Casos de Uso | 00 |
| Diagrama de Actividades | 00 |
| Funcionalidad | 00 |
| Servicios ofrecidos | 00 |
| Despliegue y configuración de los componentes | |
| Instalación | 00 |
| Configuración del servidor | 00 |
| Despliegue de la información | 00 |
| Requisitos de los sistemas | 00 |
| Errores técnicos comunes | 00 |

SECCIÓN 9

Anexos

| | |
|-----------------------------|----|
| Unidades de Transporte | |
| Base de datos | |
| Descripción por tabla | 00 |
| Casos de uso | |
| Descripción de casos de uso | 00 |
| Estaciones y Mantenimiento | |
| Base de datos | |
| Descripción por tabla | 00 |

SECCIÓN 1

INTRODUCCIÓN

- 09 Metrobús
- 10 Unidades de Transporte
- 11 Estaciones y Mantenimiento
- 13 Operaciones
- 14 Planteamiento del proyecto

Metrobús

Parte central del desarrollo de una ciudad es un sistema efectivo de transporte público. Para la mayoría de la población de las ciudades, el transporte público es el único medio para acceder a su empleo, educación y servicios públicos.

En un esfuerzo por ofrecer a la población un modo efectivo de transporte público, muchas ciudades implementan medios caros que, por lo tanto, no permiten expandirse lo suficiente para cubrir las necesidades de la población.

Metrobús es un sistema de transporte, basado en autobuses de capacidad y tecnología de punta, que brinda movilidad urbana de manera rápida y segura por medio de la integración de una infraestructura preferente, operaciones rápidas y frecuentes, sistema de pago automatizado y excelencia en calidad en el servicio.

La Misión de Metrobús es planear, administrar y controlar el Sistema de Corredores de Transporte Público de Pasajeros de la Ciudad de México, promoviendo un servicio de calidad mundial. Además, se busca contribuir a la mejora de la calidad del aire de la Zona Metropolitana del Valle de México con acciones que también mitiguen los efectos de los gases de efecto invernadero en el cambio climático global.

La visión de METROBÚS buscamos ser una unidad de la administración pública del Gobierno de la Ciudad de la Ciudad de México de excelencia en la administración, control y vigilancia del servicio público de transporte de pasajeros.

Metrobús tendrá por objeto: La planeación, administración y control del Sistema de Corredores de Transporte Público de Pasajeros de la Ciudad de México.

Metrobús es un modo de transporte BRT (Bus Rapid Transit) que combina estaciones, vehículos, servicios y alta tecnología en un sistema integral con una identidad positiva.

Unidades de Transporte

Responsabilidades

Funciones vinculadas al Objetivo 1:

- Revisión de la flota a través de la revisión periódica anual del estado de los autobuses.
- Seguimiento al programa de mantenimiento de los autobuses que realizan las empresas operadoras.
- Seguimiento de la medición de las emisiones contaminantes.
- Revisión de desempeño mecánico eléctrico operativo de las unidades en servicio.

Funciones vinculadas al Objetivo 2:

- Estudiar la oferta técnica disponible en el país.
- Revisión de las normas vigentes aplicables al parque vehicular.
- Establecimiento de necesidades técnicas de nuevos proyectos.

Funciones vinculadas al Objetivo 3:

- Seguimiento a la operación realizada por cada unidad.
- Análisis de las fallas presentadas por el parque vehicular.
- Inspección del funcionamiento durante la operación del servicio.

Misión

Determinar los lineamientos técnicos del parque vehicular que se integrará a la operación del sistema y durante la operación dar seguimiento al adecuado funcionamiento del mismo.

Objetivos

- Objetivo 1: Supervisar el buen estado físico operativo de la flota vehicular
- Objetivo 2: Establecer especificaciones técnicas para los vehículos del sistema, a través de la definición de lineamientos técnicos para la adquisición de nuevos autobuses.
- Objetivo 3: Monitorear el comportamiento en operación del parque vehicular a través de la revisión mecánica y funcional.

Estaciones

Responsabilidades

Funciones vinculadas al Objetivo 1:

- Llevar a cabo la coordinación, programación, revisión de las necesidades del mantenimiento preventivo y correctivo de las Instalaciones Fijas en los Corredores.
- Gestionar y coordinar con el ente que corresponda para facilitar el mantenimiento de las Instalaciones Fijas.
- Coordinar la entrega-recepción de los nuevos corredores que cuenten con la calidad de obra requerida así como las condiciones requeridas para la operación.
- Revisar, analizar y canalizar los reportes, levantamientos, revisiones que sean elaborados por los supervisores del estado que guarda la infraestructura del sistema.

Funciones vinculadas al Objetivo 2:

- Coadyuvar para la elaboración del proyecto conceptual que sirve de referencia para el proyecto ejecutivo.
- Colaborar en la definición de proyectos para nuevos corredores (proporcionando parámetros de diseño).

Misión

Inspeccionar la Infraestructura, determinar las desviaciones en función de las instalaciones fijas respecto de su estado, gestionando ante las instancias según corresponda y verificar las correcciones realizadas.

Objetivos

- Objetivo 1: Recibir y conservar las instalaciones fijas, coordinando mediante programas y revisiones, además de establecer el control y gestión de los reportes que faciliten el mantenimiento preventivo y correctivo.
- Objetivo 2: Coadyuvar en los proyectos conceptuales, así como en la entrega-recepción de los nuevos corredores, definiendo parámetros de calidad y condiciones de funcionalidad en la operación.

Mantenimiento

Responsabilidades

Funciones vinculadas al Objetivo 1:

- Realizar el servicio de mantenimiento de energía eléctrica, bombas de agua, plantas de energía eléctrica, drenaje, pintura, y reparaciones en general.
- Atender las necesidades de mantenimiento, pintura, electricidad, que se requieran en el desempeño de las actividades del Organismo en las estaciones del Sistema.
- Verificar el cumplimiento de los programas de mantenimiento a instalaciones que conforman la infraestructura del Sistema.

Funciones vinculadas al Objetivo 2:

- Elaborar proyectos de remodelación, adaptación, reparación y mantenimiento de las oficinas generales del organismo optimizando los recursos con programas de mantenimientos preventivos.

Funciones vinculadas al Objetivo 3:

- Informar periódicamente el cumplimiento de las metas y programas de mantenimiento a la Dirección de Administración y Finanzas.
- Coordinar, y apoyar las actividades del personal adscrito al área competencia.

Misión

Coadyuvar en la formulación del plan de distribución anual del presupuesto de gasto corriente para su aprobación, contribuyendo en la definición de criterios y prioridades de asignación de recursos para el correcto desempeño de las labores administrativas y el oportuno suministro de los servicios generales.

Objetivos

- Objetivo 1: Elaborar el Programa Anual de Mantenimiento Preventivo y Correctivo de las estaciones del Corredor y de las oficinas generales del Organismo
- Objetivo 2: Definir y elaborar los términos de referencia con las especificaciones técnicas para la contratación de servicios de mantenimiento o reparaciones mayores para las estaciones y oficinas.
- Objetivo 3: Realizar las demás actividades que le sean encomendadas por la Dirección de Administración y Finanzas afines a las funciones y responsabilidades inherentes al cargo.

Operaciones

Responsabilidades

- I. Acordar con el Director Técnico Operativo el despacho de los asuntos encomendados a las Jefaturas de Unidad adscritas a su responsabilidad;
- II. Desempeñar las comisiones que el Director Técnico Operativo le enciende y mantenerlo informado sobre el desarrollo de sus actividades;
- III. Someter a la aprobación del Director Técnico Operativo los estudios y proyectos que se elaboren para la correcta prestación, supervisión e inspección del servicio de Transporte de Público de pasajeros bajo el marco de Corredor de Transporte por las jefaturas de unidad a su cargo;
- IV. Coordinar entre sí sus respectivas labores para obtener un mejor desarrollo de las mismas;
- V. Mantener la oferta de servicio de Transporte Público bajo la modalidad de Corredores de Transporte, en equilibrio con la demanda real para la factibilidad financiera del sistema;
- VI. Colaborar en la planeación, programación, organización y evaluación del desempeño de las actividades operativas y técnicas adscritas, conforme a los lineamientos que establece este Estatuto y el Director Técnico Operativo;
- VII. Dictaminar las medidas necesarias de mejoramiento técnico operativo de las áreas adscritas y proponer al Director Técnico Operativo, la delegación en funcionarios subalternos de facultades que tengan encomendadas en cuanto a verificación de mantenimiento de los autobuses, apego a los programas de servicio, control de los indicadores de servicio;

- VIII. Suscribir los documentos relativos al ejercicio de sus atribuciones y aquellos que le sean señalados por delegación o le corresponda por suplencia;
- IX. Proporcionar la información, los datos o la cooperación técnica que les sean requeridos por otras dependencias del Gobierno del Distrito Federal;
- X. Apoyar a la Dirección Técnico Operativa del Organismo en la promoción, conducción, coordinación, vigilancia y evaluación del desarrollo del Organismo en aspectos operativos;

Misión

Consolidar para los usuarios la oferta de transporte público del servicio en la modalidad establecida para Metrobús, acorde al comportamiento de la demanda, la operación, el seguimiento del servicio, análisis de los resultados y retroalimentación de la misma, que beneficie el esquema operativo y de negocio para el Sistema.

Visión

- Objetivo 1: Revisar los resultados de los estudios y el comportamiento de la demanda para la adecuada planeación y programación de la operación de cada corredor, que contribuya a una mejora en la oferta de servicio.
- Objetivo 2: Revisar las programaciones acorde al comportamiento de la demanda y optimizar los recursos disponibles sin omitir las directrices del esquema de servicio y negocio.
- Objetivo 3: Planear la implantación de la programación del servicio, su seguimiento, y registro de resultados de la operación.
- Objetivo 4: Analizar los resultados de la operación para emitir los reportes, mejoras e indicadores, que permitan la medición de los resultados de la operación del Sistema.

Planteamiento del problema

El Metrobús es considerado un Autobús de tránsito rápido (Bus Rapid Transit, BRT) debido a que incorpora estaciones, vehículos, servicios y alta tecnología en un sistema integral con una identidad positiva. Tiene diversos componentes distintivos que juntos conforman un sistema integral. La flota de Metrobús de la Ciudad de México es de 660 autobuses de vanguardia tecnológica, con altos estándares de calidad.

El Metrobús cuenta con una estructura orgánica con diversas áreas, entre ellas el departamento de Unidades de Transporte, la cual se encarga de la revisión de la flota a través de la evaluación periódica del estado de los autobuses, el seguimiento al programa de mantenimiento de los autobuses que realizan las empresas operadoras, el análisis de las fallas presentadas por el parque vehicular y la inspección del funcionamiento durante la operación del servicio.

Derivado de un análisis de procesos se ha observado que el Metrobús cuenta con un sistema de llenado manual en papel para llevar el control de las Unidades de Transporte para posteriormente llenarse de nuevo en un formato en Excel, implicando redundancia de datos y posibles errores por parte del equipo de Metrobús. Esto ha propiciado que se ingresen datos erróneos o que estos sean ilegibles y que al momento de llenar algún formato falte información o esté duplicada.

SECCIÓN 2

INFORMACIÓN GENERAL

- 16 Formatos iniciales
- 17 Unidades de Transporte
- 18 Estaciones y Mantenimiento
- 19 Formatos finales
- 19 Unidades de Transporte
- 20 Estaciones y Mantenimiento

Formatos Iniciales

Metrobús es un sistema de transporte, basado en autobuses de capacidad y tecnología de punta, que brinda movilidad urbana de manera rápida y segura por medio de la integración de una infraestructura preferente, operaciones rápidas y frecuentes, sistema de pago automatizado y excelencia en calidad en el servicio.

La Misión de Metrobús es planear, administrar y controlar el Sistema de Corredores de Transporte Público de Pasajeros de la Ciudad de México, promoviendo un servicio de calidad mundial. Además, se busca contribuir a la mejora de la calidad del aire de la Zona Metropolitana del Valle de México con acciones que también mitiguen los efectos de los gases de efecto invernadero en el cambio climático global.

Metrobús es un modo de transporte BRT (Bus Rapid Transit) que combina estaciones, vehículos, servicios y alta tecnología en un sistema integral con una identidad positiva.

Para dar inicio con el proyecto fueron proporcionados los distintos documentos y reportes que son generados en el día a día dentro de Metrobús. Cada área tiene distintos métodos para generar sus reportes e informes. Con esta base se procedió a analizar cada uno de ellos por área y su relación entre cada proceso o flujo de trabajo.

Al término del análisis los procedimientos de minimizaron, lo que ayudó a disminuir la cantidad de reportes necesarios para la operación, así como un mejor flujo de trabajo entre procesos que aumentan la eficiencia de cada formato.

Muchos de estos formatos sufren modificaciones cada año, como consecuencia de la actualización constante entre los datos, información y procesos manuales que se realizan. Estos cambios se verán reflejados en el sistema una vez se realice la solicitud pertinente para su actualización o mejora del proceso.

Unidades de transporte

A lo largo del desarrollo del proyecto, se han presentado cambios en los formatos otorgados por parte de Unidades de Transporte – Metrobús, a continuación, se listan los primeros formatos otorgados:

- Bitácora de fallas 2019
- Consumo de combustible 2019
- Consumo y Mantenimiento empresas 2019
- Cédula de Revisión Técnica Inicial del Autobús 2019
- Cédula Técnica de Registro del Autobús 2019
- Estatus Patio 2019
- Formatos de levantamientos
- Kilometraje 2019
- Pruebas de Desempeño 2019
- REFFA 2019
- Rendimientos 2019
- Reporte del estado físico y del funcionamiento del autobús correspondiente al 2019 (31122018)
- Revisión del estado Físico y de funcionamiento del autobús 2019 (03082019)

Después de las primeras entrevistas realizadas se eliminaron algunos formatos a petición de Unidades de Transporte – Metrobús:

- Bitácora de fallas 2019
- Formatos de levantamientos

Se notificó que la mayoría de estos formatos ya no se utilizaban, ya que cada año se actualizaban, por lo que posteriormente se otorgaron las nuevas versiones, quedando los siguientes formatos a trabajar:

- Cédula de Revisión Técnica Inicial del Autobús 2020
- Cédula Técnica de Registro del Autobús 2020
- Consumo de combustible 2019
- Consumo Mantenimiento empresas 2020
- Estatus Patio 2019
- Kilometraje 2019
- Pruebas de Desempeño 2020
- Rendimientos 2018
- Reporte de Incidencias 2020
- Reporte del Estado Físico y del funcionamiento del autobús correspondiente al 2020 (04012020)
- Revisión del Estado Físico y de Funcionamiento del Autobús 2020 (05012020)

Estaciones y Mantenimiento

A continuación, se listan los primeros formatos otorgados:

- A Señalamiento Horizontal Y Vertical (A1, A2, A3, A4, A5, A6)
- Formato Alumbrado En Puentes Peatonales Y Rampas De Acceso (B1, B2 B3)
- Formato Superficie De Rodamiento Concreto Hidráulico (C1, C2, C3, C4, C5, C6)
- Formato Elementos De Confinamiento (D1, D2, D3. D4, D5, D6)
- Formato E3 Cortinas De Servicio Y Puertas De Acceso A Estacion (E1, E2, E3, E4, E5, E6)
- Formato Luminarias En Estacion (F1, F2, F3, F4, F5, F6)
- Formato Pintura En Muros Interior Y Exterior (G1, G2, G3, G4, G5, G6)
- Formato Pintura En Estructura (H1, H2, H3, H4, H5, H6)
- Formato Señaletica Interna Y Externa (I1, I2, I3, I4, I5, I6)
- Formato De Pisos Guia Táctil, Linea De Seguridad (J1, J2, J3, J4, J5, J6)
- Formato Luminarias De Los Módulos De Servicio (K1, K2, K3, K4, K5, K6)
- Formato Accesorios Y Muebles Sanitarios (L1, L2, L3, L4, L5, L6)
- Formato De Chapas Y Jaladeras (M1, M2, M3, M4, M5, M6)
- Formato Poda, Deshierbe Y Levantamiento De Fuste (N1, N2, N3, N4, N5, N6)
- Formato Rehabilitación De Publicidad 5m2 (O1, O2, O3, O4, O5, O6)
- Formato Dispositivos Para Discapacitados (P1, P2, P3, P4, P5, P6)
- Formato Levantamiento De Apagadores Y Contactos (Q1, Q2, Q3, Q4, Q5, Q6)
- Formato Apagadores Y Contactos (R1, R2, R3)
- Formato Estelas En Estación (T1, T2, T3, T4, T5, T6)
- Formato U4 Levantamiento De Las Terminales
- Formato V1 Sanitarios 5m2
- Formato Equipo De Recaudo (W1, W2, W3, W4, W5, W6)
- Formato Extintores (X1, X2, X3, X4, X5, X6)
- Formato Polycarbonato (Z1, Z2, Z3, Z4, Z5, Z6)
- Formato De Solicitud De Trabajo
- Formato Orden De Trabajo
- Herramienta Almacén

Formatos finales

Después del análisis de los formatos se llegó a un acuerdo final con cada una de las áreas de metrobus. Los flujos de trabajo, datos e información necesaria para cada proceso fue discutida y analizada por el equipo de desarrollo y los miembros de equipo de las diferentes áreas que hacen uso de estos formatos.

Al término de los distintos análisis y consensos que se llevarán a cabo se establecieron los siguientes formatos finales que formarán parte del sistema final y que de acuerdo a las necesidades de los equipos serán actualizados a lo largo de la vida del sistema.

Unidades de Transporte

Después de realizar el análisis de cada documento trabajado se pudo reducir a lo siguiente:

- Cédula de Revisión Técnica Inicial del Autobús 2020
- Cédula Técnica de Registro del Autobús 2020
- Consumo de combustible 2019
- Consumo Mantenimiento empresas 2020
- Estatus Patio 2019
- Kilometraje 2019
- Pruebas de Desempeño 2020
- Rendimientos 2018
- Reporte del Estado Físico y del funcionamiento del autobús correspondiente al 2020 (REFFA)

Estaciones y Mantenimiento

Después de analizar cada uno de estos formatos con el área de Estaciones en MB, se llegó a la conclusión de que muchos datos se repetían y que muchos eran algunas veces innecesarios por los procesos. Al final solo la información de los daños era plasmada en un formato de solicitud de trabajo, y era necesaria el formato de solicitud de almacén para cuando se reparara esa parte, utilizar el material del dicho almacén.

Llegando a un documento que se proporciona para obtener los conceptos y partes clasificados por categorías y revisando los formatos pudimos obtener las fallas, dicho documento es el siguiente:

- Catálogo General de Conceptos de las 7 líneas de MB

Desglosando dicho catálogo y corroborando con todos los formatos ya mencionados se llegó a una relación de partes y fallas clasificadas en categorías y conceptos. Haciendo más eficiente la captura de daños presentadas en las estaciones, así como también la solicitud que se deben hacer para darles mantenimiento. Después de todo el análisis que se hizo la información quedó plasmada en los siguientes formatos:

- Levantamiento
- Solicitud de trabajo
- Solicitud de almacén

S E C C I Ó N 3

O B J E T I V O S

Objetivos

Objetivo General

Implementar un modelo de sistema unificado aplicable a los sistemas de información del Metrobús que conjunte los procesos que se encuentran en papel, para la recolección y difusión de la información más sencilla y al alcance de todos los miembros del Metrobús, que permita la generación de conocimiento a partir del análisis obtenido de los datos para la mejora del transporte en cuestión.

Objetivos Particulares

- Diseñar un modelo que permita analizar cada uno de los campos de información necesarios en los sistemas disponibles del Metrobús.
- Diseñar el modelo unificado entre los sistemas de información del Metrobús.
- Analizar los campos disponibles en los sistemas de información del Metrobús.
- Interconectar la comunicación entre los sistemas de información disponibles del Metrobús.

SECCIÓN 4

JUSTIFICACIÓN

Justificación

Actualmente en las áreas de metrobús, el registro de datos se realiza de manera manual, por lo que puede llegar a generar pérdidas de información o errores en la inserción gracias a la interpretación personal de cada empleado, además de que es necesarios el uso de papel, por lo que el medio ambiente se ve sumamente afectado.

A partir de esto, surge la necesidad de implementar una solución a esta problemática, utilizando un sistema web de administración de información que se encargue de agilizar el manejo de los datos necesarios para crear los reportes que utiliza Metrobús de la Ciudad de México en las tareas técnicas para el control de sus unidades, estaciones y métricas de mantenimiento.

Con esto, omitimos un proceso redundante al volver a capturar la información de los formatos, reduciendo el uso de papel y evitamos problemas de legibilidad en los mismos, manteniendo los procesos base que el personal de Metrobús utiliza.

SECCIÓN 5

ESTADO DEL ARTE

- 26 Sistema de Gestión Documental.
- 26 Sistema de Gestión de Usuarios.
- 27 Metrobus, SUG, y SDG.
- 27 Ssitema Gestión de inventarios.
- 28 Metrobus y Sistema Gestión de Invetnario.
- 00 Sistemas - CRUD

Sistema Gestión Documental (SGD).

La gestión documental consiste en controlar de un modo eficiente y sistemático la creación, la utilización y la disposición de los documentos. Sus objetivos primordiales son:

- Coordinar y controlar las actividades específicas que afectan a la creación, recepción, la ubicación, el acceso y la preservación de los documentos.
- Custodiar y recuperar la documentación que genera de un modo eficaz.
- Garantizar el buen funcionamiento y el cumplimiento de la legislación vigente.

Motivos por los cuales es bueno tener un gestor documental

- Evitar la pérdida de la documentación, ya que todos los documentos están integrados e identificados en un único sistema.
- Evita la perdida de tiempo en la recuperación en la documentación.
- Permite el control de accesos y de seguridad de la información, e impide que terceras personas accedan a los contenidos.
- Establece un ciclo de vida de la documentación y posibilita su destrucción de un modo seguro y certificado.
- Establece la colaboración entre los diferentes órganos de un modo seguro y certificado de la institución para una óptima gestión del SGD.

Sistema de gestión de usuarios (SUG)

Tener un sistema de gestión de usuarios es de mucha ayuda cuando se trata de controlar el acceso al sistema. Se pueden declarar diferentes tipos de usuarios con su respectivo rol dentro de la empresa. Estos tendrán diferentes tipos de privilegios dentro de una sesión en el sistema con ellos tendremos mayor seguridad en el acceso de los archivos.

Ventajas

- Seguridad del sistema.
- Control del flujo de la información.
- Administración del personal trabajador.



Metrobus , SUG y SDG

Dentro del mercado existen varias empresas que ofrecen sus servicios de gestión documental. El precio puede variar dependiendo de las características deseadas. Al ser un sistema especialmente desarrollado para el MetroBus no existe un programa comercial que sea parecido en la gestión de documentos y administración de usuarios pero si debemos mencionar algunos son los siguientes:

- Quipu : Precio mensual: 770 MXN + IVA.
- Factorial: Precio mensual: 96 MXN + IVA.

Cabe resaltar que estos documentos son para gestionarlos ambos sistemas comerciales cuentan con un sistema de gestor de usuarios para su acceso a los documentos. En el sistema del Metrobús se crean y además se gestiona la documentación para el uso en un futuro, además de la interacción con el documento para la firma digitalizada.

Sistema gestor de Inventarios

Dentro del sistema cuenta con un gestor de inventario para la correcta administración de cada uno de los materiales. Con ella se tendrá un control completo de cada uno de los materiales y poder solicitar más si así se desea.

- Control de materiales.
- Manejo de devoluciones.
- Asegurar que no falte un material.
- Saber siempre su estado actual del material.

Metrobus y Sistema de inventario

Para la gestión de inventarios tenemos varias opciones, tanto de paga como gratuitas de paga tenemos:

- Cin7 : Precio 300USD al mes aprox 6,235.41 MXN.
- Veeqo: Precio aprox 4,813.66 MXN.

Gratis

- ABC Inventory.
- Delivr'd.
- Odoo Inventory.

En estos software se puede realizar una gestión de inventario. El sistema desarrollado para Metrobus cuenta con la posibilidad de gestionar el inventario y revisar el uso de las herramientas utilizadas. Es un sistema de inventarios especialmente para metrobus contando con todas las ventajas de tener un sistema de inventario de paga o gratuito

Sistemas - CRUD

Dentro del sistema contamos con distintos módulos los cuales cuentan con CRUD (Create, Read, Update, Delete). Estos sistemas en el mercado varían de precio según sus funcionalidades, pero son muy comunes encontrarlos en sistemas administrativos, ya que se basan dar de alta, modificar, ver información, actualizarla y si se desea eliminarla. En el mercado podemos encontrar distintos CRUD dependiendo el módulo que se desea y el precio normalmente varía dependiendo la tecnología utilizada, al final siempre se cumple con los mismos objetivos.

El sistema del metrobus está desarrollado con varios módulos. los cuales cuentan con un CRUD, para la administración de diferentes áreas.

SECCIÓN 6

MARCO TEÓRICO

- 30 Sistema
- 30 Sistema de información
- 31 Aplicación web
- 31 Base de datos
 - 32 Sistema de gestión de bases de datos
 - 32 Usuarios de bases de datos
 - 33 Bases de datos MariaDB
- 33 Arquitectura cliente servidor
- 34 Lenguajes de programación
 - 34 Javascript
 - 36 HTML5 / CSS
- 38 Framework
 - 38 Node.js
 - 38 React
 - 39 Express JS

Sistema

Es un conjunto de funciones que operan en armonía o con un mismo propósito, y que puede ser ideal o real. Un sistema por su propia naturaleza posee determinadas reglas o normas que regulan su funcionamiento por lo que puede ser entendido, aprendido y enseñado, unos ejemplos de sistemas: el sistema respiratorio, cardiovascular, el sistema económico de un país, el ciclo de la vida, entre otros.

Cualquier sistema tiene su complejidad, pero debe poseer una coherencia discreta acerca de sus propiedades y operaciones. Por lo general, los elementos o módulos de un sistema interactúan y se interrelacionan entre sí. En algunas ocasiones llegan a existir subsistemas.

Sistema de información

Un sistema de información es un conjunto de elementos o componentes interrelacionados que recaban, manipulan, almacenan y distribuyen datos e información y proporciona una reacción correctiva si no se ha logrado cumplir un objetivo.

Un sistema de información basado en computadora es un conjunto único de hardware, software, bases de datos, personas y procedimientos configurado para recolectar, almacenar y procesar datos para convertirlos en información.

Aplicación Web

Es un software desarrollado con tecnologías web, sin la necesidad de ninguna instalación ya que está totalmente cargada desde un servidor. También es posible verlo como una página web con características de una aplicación, entre más tenga el sitio, se vuelve apropiado llamarlo una aplicación web. En ellas se pueden encontrar más elementos interactivos de la interfaz de usuario como por ejemplo barras de herramientas, elementos arrastrables, editores de texto y barras de movimiento. El espectro dentro del cual puede estar una aplicación web llega desde herramientas de gráficos o juegos que se ejecutan desde el navegador.

Entre las ventajas de este tipo aplicaciones también se encuentra un mejor control sobre los fallos de seguridad, ya que, a diferencia de las aplicaciones nativas, que se solucionan descargando una nueva versión o descargando una actualización, en las aplicaciones web se implementa directamente en el software.

Base de datos

Una base de datos es un conjunto de datos estructurados y definidos a través de un proceso específico, que busca evitar la redundancia, y que se almacenará en algún medio de almacenamiento masivo, como un disco. Es una serie de datos organizados y relacionados entre sí, los cuales son recolectados y explotados por los sistemas de información de una empresa o negocio en particular.

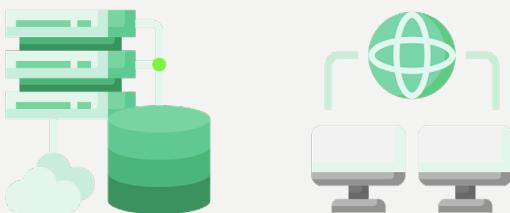
Entre las principales características de los sistemas de base de datos podemos mencionar:

- Independencia lógica y física de los datos.
- Redundancia mínima.
- Acceso concurrente por parte de múltiples usuarios.
- Integridad de los datos.
- Consultas complejas optimizadas.
- Seguridad de acceso y auditoría.
- Respaldo y recuperación.
- Acceso a través de lenguajes de programación estándar

Sistema gestor de base de datos

El SGBD puede implementar instrucciones dadas por los diferentes usuarios, que se agrupan mínimamente en: DDL (Lenguaje de Definición de Datos) y DML (Lenguaje de Manipulación de Datos), aunque también suele reconocerse al DCL (Lenguaje de Control de Datos).

- DDL: Es el conjunto de órdenes que permiten definir la estructura de una base de datos.
- DML: Las instrucciones que conforman este grupo son las que están incluidas en las aplicaciones y se usan para alterar el contenido de un archivo de datos.
- DCL: Son órdenes que se utilizan para implementar seguridad en la base de datos



Usuarios de bases de datos

Después de reconocer las entidades y relaciones entre ellas, se pueden detectar usuarios que interactúan con la base de datos de distinta manera y con distintos objetivos.

A continuación, se muestra una clasificación muy general:

- Administrador de la base de datos: comúnmente es el profesional – ingeniero o analista – con perfil técnico que, en el ambiente informático, se denomina DBA (Data Base Administrator). Este profesional recibe las especificaciones del Equipo de Análisis y Diseño para su implementación en un Sistema de Gestión de Base de Datos.
- Programador de aplicaciones: conoce los casos que se desarrollarán, los prototipos de interfaces y las estructuras de los almacenamientos que se manipularon. El programador genera las aplicaciones necesarias en el sistema para la obtención de las salidas, que se plantearon en la propuesta de solución.
- Usuario general: es el personal que interactúa con las aplicaciones programadas

MariaDB

MariaDB es un sistema gestor de base de datos (SGBD), surgido de cambios hechos al código fuente original de MySQL y generando un derivado con mejoras y cambios denominado MariaDB, manteniendo la licencia GPL (General Public License). Cuenta con dos motores de almacenamiento nuevos, llamados Aria y XtraDB, sustituyendo InnoDB y MyISAM.

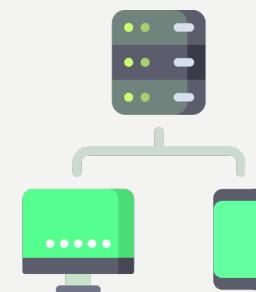
- Soporta diferentes sentencias SQL, estructuras y reglas, funciones y procedimientos, funciones definidas por el usuario, variables de servidor e incluso particionamiento de tablas.
- MariaDB viene con nuevas opciones, características, extensiones, motores de almacenamiento y corrección de errores que no están presentes en MySQL. Es también compatible con bases de datos de Oracle.
- MariaDB y MySQL comparten las mismas características de seguridad

Arquitectura Cliente-Servidor

La arquitectura cliente- servidor es un modelo de aplicación distribuida en el que las tareas se reparten entre los proveedores de recursos o servicios, llamados servidores, y los demandantes, llamados clientes. Un cliente realiza peticiones a otro programa, el servidor, quien le da respuesta.

En esta arquitectura la capacidad de proceso está repartida entre los clientes y los servidores, aunque son más importantes las ventajas de tipo organizativo debidas a la centralización de la gestión de la información y la separación de responsabilidades, lo que facilita y clarifica el diseño del sistema.

La separación entre cliente y servidor es una separación de tipo lógico, donde el servidor no se ejecuta necesariamente sobre una sola máquina ni es necesariamente un sólo programa



Lenguajes de programación

Un lenguaje de programación es un lenguaje de computadora que los programadores utilizan para comunicarse y para desarrollar programas de software, aplicaciones, páginas webs, scripts u otros conjuntos de instrucciones para que sean ejecutadas por los ordenadores.

Así como los idiomas que utilizan los humanos para comunicarse, los ordenadores tienen sus propios lenguajes de programación. Cada lenguaje de programación tiene un conjunto único de palabras clave (palabras que entiende) y una sintaxis especial para organizar las instrucciones del programa específico de programación.

Javascript

Es un lenguaje de programación que permite hacer funciones complejas en las páginas web, creando contenido de actualización dinámica, controlar multimedia, animar imágenes, etc.

Entre sus características podemos encontrar:

- Almacenar valores útiles dentro de variables.
- Operaciones sobre fragmentos de texto (conocidas como “cadenas” (strings) en programación).
- Ejecuta código en respuesta a ciertos eventos que ocurren en una página web. por ejemplo, al hacer click en un botón o al estar escribiendo en una caja de texto, etc.

Otra ventaja del uso de JavaScript son las denominadas interfaces de programación de aplicaciones(API).

Las API's son conjuntos de bloques de construcción de códigos listos para usar, que permiten a un desarrollador implementar programas que de otro modo serían difíciles. En otras palabras, serían como los kits de muebles prefabricados para la construcción de viviendas. Se dividen en dos categorías:

API DEL NAVEGADOR

Estas se encuentran integradas en tu navegador web y pueden exponer datos del entorno informático circundante o realizar tareas complejas y útiles. Por ejemplo:

- La API del DOM (Document Object Model): Te permite manipular HTML y CSS, crear, eliminar y cambiar el HTML, aplicar dinámicamente nuevos estilos a tu página, cada vez que ves aparecer una ventana emergente en una página, o se muestra algún nuevo contenido, etc.
- La API de Geolocalización recupera información geográfica. Como “Google Maps” puede encontrar tu ubicación y trazar en un mapa.
- Las APIs Canvas y WebGL te permiten crear gráficos animados en 2D y 3D.
- Las APIs de audio y video como HTMLMediaElement y WebRTC te permiten interactuar con multimedia, como reproducir audio y video directamente en una página web, o tomar video de tu cámara web y mostrarlo en la computadora de otra persona.

API DE TERCEROS

No están integradas en el navegador de forma predeterminada y, por lo general, debes obtener su código e información de algún lugar de la Web.

Por ejemplo:

- La API de Twitter te permite hacer cosas como mostrar tus últimos tweets en tu sitio web.
- La API de Google Maps y la API de OpenStreetMap te permiten insertar mapas personalizados en tu sitio web y otras funciones similares
- ejemplo:
 - La API del DOM (Document Object Model): Te permite manipular HTML y CSS, crear, eliminar y cambiar el HTML, aplicar dinámicamente nuevos estilos a tu página, cada vez que ves aparecer una ventana emergente en una página, o se muestra algún nuevo contenido, etc.
 - La API de Geolocalización recupera información geográfica. Como “Google Maps” puede encontrar tu ubicación y trazar en un mapa.
 - Las APIs Canvas y WebGL te permiten crear gráficos animados en 2D y 3D.
- Las APIs de audio y video como HTMLMediaElement y WebRTC te permiten interactuar con multimedia, como reproducir audio y video directamente en una página web, o tomar video de tu cámara web y mostrarlo en la computadora de otra persona.

HTML5

HTML, que significa Lenguaje de Marcado de Hipertextos (Hyper Text Markup Language), es la pieza más básica para la construcción de la web y se usa para definir el sentido y estructura del contenido en una página web. Otras tecnologías además de HTML son usadas generalmente para describir la apariencia/presentación de una página web (CSS) o su funcionalidad (JavaScript). La palabra “Hipertexto” se refiere a los enlaces que conectan las páginas web entre sí, ya sea dentro de un mismo sitio web o entre diferentes sitios web. Los vínculos son un aspecto fundamental de la web.

HTML5 es la última versión de HTML. El término representa dos conceptos diferentes:

- Se trata de una nueva versión de HTML, con nuevos elementos, atributos y comportamientos.
- Contiene un conjunto más amplio de tecnologías que permite a los sitios Web y a las aplicaciones ser más diversas y de gran alcance. A este conjunto se le llama HTML5.

Diseñado para ser utilizable por todos los desarrolladores de Open Web:

- **Semántica:** Permite describir con mayor precisión cuál es su contenido.
- **Conectividad:** Permite comunicarse con el servidor de formas nuevas e innovadoras.
- **Sin conexión y almacenamiento:** Permite a las páginas web almacenar datos localmente en el lado del cliente y operar sin conexión de manera más eficiente.

- **Multimedia:** Nos otorga un excelente soporte para utilizar contenido multimedia como lo son audio y video nativamente.
- **Gráficos y efectos 2D/3D:** Proporciona una amplia gama de nuevas características que se ocupan de los gráficos en la web como lo son canvas 2D, WebGL, SVG, etc.
- **Rendimiento e Integración:** Proporciona una mayor optimización de la velocidad y un mejor uso del hardware.
- **Acceso al dispositivo:** Proporciona APIs para el uso de varios componentes internos de entrada y salida de nuestro dispositivo.

CSS

Se denomina lenguaje de hojas de estilo en cascada y se usa para estilizar elementos escritos en un lenguaje de marcado como HTML. CSS separa el contenido de la representación visual del sitio. CSS fue desarrollado por W3C en 1996 por una razón muy sencilla. HTML no fue diseñado para tener etiquetas que ayuden a formatear la página. Está hecho solo para escribir el marcado para el sitio. CSS utiliza una sintaxis simple basada en el inglés con un conjunto de reglas que la gobiernan. HTML no fue hecho con la intención de utilizar elementos de estilo, sino solo para el marcado de la página. Fue creado simplemente para describir el contenido.

Cada declaración incluye un nombre de propiedad CSS y un valor, separados por dos puntos. Una declaración CSS siempre termina con un punto y coma, y los bloques de declaración están rodeados por llaves.

Existen tres estilos de implementación de CSS:

- **Estilo Interno:** Se cargan cada vez que se actualiza un sitio web, lo que puede aumentar el tiempo de carga. Además, no se podrá usar el mismo estilo CSS en varias páginas, ya que está contenido en una sola página. Sin embargo, esto también tiene sus beneficios. Tener todo en una página facilita compartir la plantilla para una vista previa.

- **Estilo Externo:** Todo se hace externamente en un archivo .css. Esto significa que puedes hacer todo el estilo en un archivo separado y aplicar el CSS a cualquier página que quieras. El estilo Externo también puede mejorar los tiempos de carga.
- **Estilo Inline:** Inline trabaja con elementos específicos que tienen la etiqueta <style>. Cada componente tiene que ser estilizado, por lo que podría no ser la mejor forma, ni la más rápida para manejar CSS. Pero puede de ser útil, si quieres cambiar un solo elemento, tener una vista previa rápida de los cambios o tal vez no tengas acceso a los archivos CSS

Frameworks

Un framework es un marco o esquema de trabajo generalmente utilizado por programadores para realizar el desarrollo de software. Utilizar un framework permite agilizar los procesos de desarrollo ya que evita tener que escribir código de forma repetitiva, asegura unas buenas prácticas y la consistencia del código.

Un framework es por tanto un conjunto de herramientas y módulos que pueden ser reutilizados para varios proyectos. Uno de los frameworks más conocidos y utilizados es el .NET Framework de Microsoft para webs.

NODEJS

Es en un entorno - sistema de tiempo de ejecución para Javascript, multiplataforma y de código abierto. Con esto es posible ejecutar código Javascript fuera del explorador. Es capaz de correr código de manera asíncrona y su arquitectura está orientada a eventos.

Este modelo de entrada y salida sin bloqueo controlado por eventos es lo que nos permite una ejecución ligera y eficiente. Node soporta protocolos TCP, DNS Y HTTP, y entre los puntos fuertes que tiene es la capacidad de mantener un gran número de conexiones abiertas

REACT

React es una librería de Javascript para Front End, utilizada para construir interfaces de usuario rápidas e interactivas para la web y aplicaciones móviles. Es de código abierto, basada en componentes y responsable únicamente de la capa de visualización de la aplicación.

Entre sus características más resaltantes tenemos:

- Fácil creación de aplicaciones dinámicas, debido a que requiere menos codificación que una implementación con Javascript base.
- Incremento de desempeño en la aplicación, debido a que se utiliza Virtual DOM, cambiando componentes conforme se ocupan y no todos los componentes al mismo tiempo.
- Componentes reutilizables, que permiten un flujo de codificación reducido en comparación a las aplicaciones convencionales.
- Mismo desarrollo tanto para aplicaciones web como para móviles, utilizando el derivado React Native.

EXPRESS JS

Es un framework de aplicaciones web de tipo Node.js minimalista y flexible que proporciona un conjunto sólido de características para aplicaciones web y móviles. Este se utiliza para poder apoyar a Node en tareas que no están directamente soportadas, como lo es el manejo de los diferentes verbos HTTP, como peticiones GET, POST, DELETE, etc., y se encarga de gestionar de forma separada las peticiones por medio de diferentes direcciones URL o rutas.

Entre las características que ofrece Express tenemos:

- Escritura de manejadores de peticiones HTTP en diferentes caminos.
- Integración de motores con renderización de vistas para generar respuestas con el uso de plantillas.
- Establecimiento de ajustes de aplicaciones web, tales como el puerto para conectar o la localización de las plantillas de respuesta.
- Procesamiento de peticiones de tipo middleware, en cualquier punto, dentro de la tubería de manejo de la petición.

SECCIÓN 7

ANÁLISIS DEL SISTEMA

| | | | |
|----|------------------------------------|----|---|
| 41 | Metodologías | 53 | Análisis de software |
| 42 | Métricas y modelo de estimación | 53 | Análisis de aplicaciones web |
| 44 | Modelo de estimación cocomo | 53 | Frameworks para aplicaciones web |
| 45 | Análisis de requerimientos | 56 | Base de datos |
| 45 | Requerimientos funcionales | 58 | Seguridad |
| 46 | Requerimientos no funcionales | 59 | Sistema de autenticación |
| 46 | Actores del sistema | 59 | Control de cuentas |
| 46 | Reglas del negocio | 59 | Codificación de imágenes |
| 46 | Análisis de factibilidad | 60 | Control de acceso de red |
| 47 | Análisis de factibilidad técnica | 60 | Gestión de credenciales |
| 47 | Análisis de factibilidad operativa | 60 | Gestión del sistema operativo |
| 47 | Análisis de factibilidad económica | 61 | Encriptación de passwords por medio de bcrypt |
| 49 | Análisis de riesgos | | |
| 49 | Tipos de riesgos | | |
| 50 | Planes de contingencia | | |

Metodologías

El método elegido para el desarrollo de este Trabajo Terminal es el modelo de prototipos, que permite comprender y aclarar fácilmente los aspectos que garantizan que desarrolladores y usuarios estén de acuerdo en lo que necesitan y las soluciones que necesitan. En este caso, se tiene un cliente específico, por lo que, con base en la encuesta a realizar, se define el objetivo general del sistema a desarrollar y no se definen en detalle los requisitos de entrada, procesamiento y salida, los cuales se definen claramente a lo largo del desarrollo del proyecto, según las investigaciones que se realicen. Este método permite entender el resultado de la construcción y mejorar el resultado en cada iteración.

El paradigma de construcción de prototipos tiene seis pasos:

- Recolección y refinamiento de requisitos.
- Modelado, diseño rápido.
- Construcción de cada prototipo.
- Desarrollo, evaluación del prototipo por el cliente.
- Refinamiento del prototipo.
- Producto de Ingeniería.

Las ventajas del paradigma son las siguientes:

- No modifica el flujo del ciclo de vida.
- Reduce el riesgo de construir productos que no satisfacen las necesidades de los usuarios.
- Reduce el costo y aumenta la probabilidad de éxito.
- Exige disponer de las herramientas adecuadas.
- Este modelo es útil al conocer los objetivos generales para el sistema, pero no identifica los requisitos detallados de entrada, procesamiento o salida.
- También ofrece un mejor enfoque cuando el responsable del desarrollo del software está inseguro de la eficacia de un algoritmo, de la adaptabilidad de un sistema operativo o de la forma que debería tomar la interacción humano-máquina.

Aquí va una imagen

Métricas y modelo de estimación

Al planificar un proyecto, las estimaciones de costos y mano de obra necesarias deben obtenerse a través de métricas de software, que se utilizan para recopilar datos cualitativos sobre el software y sus procesos para mejorar su calidad.

En la mayoría de los desafíos técnicos, las métricas nos ayudan a entender tanto el proceso técnico que se utiliza para desarrollar un producto, como el propio producto. El proceso para intentar mejorarlo, el producto se mide para intentar aumentar su calidad.

Aunque las métricas de producto para el software de computadora son imperfectas, pueden proporcionar una forma sistemática de valorar la calidad con base en un conjunto de reglas claramente definidas.

Para el desarrollo del proyecto decidimos ocupar la métrica basada en funciones, la cual se basa en la métrica de punto de función (PF), de modo que puede usarse de manera efectiva como medio para medir la funcionalidad de un sistema. Al usar datos históricos, la métrica PF puede entonces usarse para:

- Estimar el costo o esfuerzo requerido para diseñar, codificar y probar el software.
- Predecir el número de errores que se encontrarán durante las pruebas.
- Prever el número de componentes y/o de líneas fuente proyectadas en el sistema implementado.

Los puntos de función se derivan usando una relación empírica basada en medidas contables (directas) del dominio de información del software y en valoraciones cualitativas de la complejidad del software.

Para calcular los Puntos de Función (PF), se usa la siguiente relación:

$$PF = \text{conteototal} \times [0.65 + 0.01 \times \sum(F_i)]$$

Donde conteo total es la suma de todas las entradas PF obtenidas de la Tabla V, los F_i ($i = 1$ al 14) son Factores de Ajuste de Valor (FAV) con base en respuestas a las siguientes preguntas:

| 0 | 1 | 2 | 3 | 4 | 5 |
|---|------------|----------|-------|---------------|----------|
| Sin influencia | Incidental | Moderado | Medio | Significativo | Esencial |
| 1. ¿El sistema requiere de respaldo y recuperación confiables? | | | | | 2 |
| 2. ¿Se requieren comunicaciones de datos especializadas para transferir información hacia o desde la aplicación? | | | | | 5 |
| 3. ¿Existen funciones de procesamiento distribuidas? | | | | | 0 |
| 4. ¿El desempeño es crucial? | | | | | 2 |
| 5. ¿El sistema correrá en un entorno operativo existente y enormemente utilizado? | | | | | 5 |
| 6. ¿El sistema requiere entrada de datos en línea? | | | | | 5 |
| 7. ¿La entrada de datos en línea requiere que la transacción de entrada se construya sobre múltiples pantallas u operaciones? | | | | | 5 |
| 8. ¿Los ALI (archivo configuración sistema) se actualizan en línea? | | | | | 2 |

| | |
|---|-----------|
| 9. ¿Las entradas, salidas, archivos o consultas son complejos? | 1 |
| 10. ¿El procesamiento interno es complejo? | 5 |
| 11. ¿El código se diseña para ser reutilizable? | 3 |
| 12. ¿La conversión y la instalación se incluyen en el diseño? | 3 |
| 13. ¿El sistema se diseña para instalaciones múltiples en diferentes organizaciones? | 0 |
| 14. ¿La aplicación se diseña para facilitar el cambio y su uso por parte del usuario? | 5 |
| Cuenta Total ($\Sigma(F_i)$) | 43 |

Las métricas orientadas a la función son medidas indirectas del software y del proceso por el cual se desarrolla. En lugar de calcular las LDC, las métricas orientadas a la función se centran en la funcionalidad o utilidad del programa. Las métricas orientadas a la función fueron el principio propuestas por Albercht quien sugirió un acercamiento a la medida de la productividad denominado método del punto de función. Los puntos de función que obtienen utilizando una función empírica basado en medidas cuantitativas del dominio de información del software y valoraciones subjetivas de la complejidad del software.

Los puntos de función se calculan llenando la tabla como se muestra a continuación:

| Valor de dominio de información | Factor ponderado | | | | | | |
|---------------------------------|------------------|--------|----------|----------|---------------|----------|----|
| | Conteo | Simple | Promedio | Complejo | Factor tomado | Subtotal | |
| Entradas de usuario | 3 | x | 3 | 4 | 6 | 4 | 12 |
| Salidas de usuario | 1 | x | 4 | 5 | 7 | 4 | 4 |
| Peticiones del usuario | 4 | x | 3 | 4 | 6 | 6 | 24 |
| Archivos | 1 | x | 7 | 10 | 15 | 7 | 7 |
| Interfaces externas | 1 | x | 5 | 7 | 10 | 10 | 10 |
| Conteo total | 57 | | | | | | |

Entradas de usuario:

1. Datos de los formatos
2. Imágenes
3. Credenciales

Salidas de usuario:

1. Mensaje de aviso

Peticiones del usuario:

1. Creación de PDF
2. Búsqueda de PDF
3. Ingreso o Login
4. Registro de empleado

Archivos:

1. PDF construidos

Interfaces externas:

1. Aplicación Web

Al obtener los datos, se sustituyen en la siguiente fórmula:

$$PF = \text{conteototal} \times [0.65 + 0.01 \times \sum(Fi)] \quad (a)$$

$$PF = 57 \times (0.65 + 0.01 \times 43) = 61.56(b)$$

La cantidad de funciones obtenidas es 61.56. Tomando en cuenta que este proyecto se desarrollará con un equipo de 3 personas.

MODELO DE ESTIMACIÓN COCOMO

En el año 1981 Barry Boehm publica el Modelo Constructivo de Costos (por sus siglas en inglés COCOMO), acorde a las prácticas de desarrollo de software de aquel momento [Boehm 1981]. Durante la década de los ochenta, el modelo se continuó perfeccionando y consolidando, siendo el modelo de estimación de costos más ampliamente utilizado en el mundo. Al aparecer las computadoras personales y generalizarse su uso, surgieron algunas implementaciones. Varias empresas comenzaron a comercializar herramientas de estimación computarizadas.

En los años noventa, las técnicas de desarrollo de software cambiaron dramáticamente, surgieron la necesidad de reusar software existente, la construcción de sistemas usando librerías, etc. Estos cambios comenzaron a generar problemas en la aplicación del modelo COCOMO. La solución fue reinventar el modelo. Después de algunos años y de un esfuerzo combinado de USC-CSE (University of Southern California- Center For Software Engineering), IRUS at UC Irvine y organizaciones privadas, aparece COCOMO II. Las incorporaciones a este modelo lo reforzaron e hicieron apto para ser aplicado en proyectos vinculados a tecnologías como orientación a objetos, desarrollo incremental, composición de aplicación, y reingeniería. COCOMO II consta de tres modelos, cada uno de los cuales ofrece una precisión acorde a cada etapa de desarrollo del proyecto. Enunciados en orden creciente de fidelidad son, modelo de Com-

-posición de Aplicación, Diseño Temprano y Post Arquitectura.

Los modelos COCOMO están definidos para tres tipos de proyectos:

- Modo Orgánico: Proyectos de software relativamente pequeños y sencillos.
- Modo Semiacoplado: Proyectos de software intermedio, con variados niveles de experiencia, que deben satisfacer requisitos poco o medio rígidos.
- Modo Empotrado: proyectos de software que deben ser desarrollados en un conjunto de hardware, software y restricciones operativas muy restrictivas.

El modelo que se utilizará será COCOMO -"semiacoplado"-, considerando que el prototipo tiene el desarrollo de software. Como estamos en la primera fase de construcción utilizaremos COCOMO básico, el cual estima el esfuerzo y el tiempo empleado en el desarrollo de un proyecto de software usando dos variables predictivas denominadas factores de costo: el tamaño del software y el modo de desarrollo. Las ecuaciones básicas son:

1. Esfuerzo:

$$PM = PF / (1/\#Personas)$$

Donde:

- PM: Es el esfuerzo estimado. Representa las horas-hombre necesarias para ejecutar el proyecto.
- PF: Puntos de función.
- #Personas: Es el número de integrantes que forman el equipo encargado de desarrollar el proyecto.

2. Duración en horas:

$$D.H. = \text{Esfuerzo} / 0.125$$

3. Duración en meses:

$$D.M. = D.H. / (168(\text{hrs/Mes}))$$

Procedemos a calcular el esfuerzo (persona – mes), tiempo de desarrollo, personal requerido y la productividad:

- Esfuerzo (horas-hombre):

$$PM = 61.56 / (1/3) = 184.68 \text{ hrs/persona}$$
- Duración en horas:

$$D.H.=184.68/0.125=1477.44 \text{ hrs}$$
- Duración en meses:

$$D.M =1477.44/168=8.8 \approx 9 \text{ Meses}$$

De acuerdo con los datos obtenidos del modelo COCOMO obtuvimos que para este tipo de proyecto lo más recomendado sería una duración de 9 meses, aproximadamente, con un estimado de 6 horas diarias de desarrollo en conjunto.

ANÁLISIS DE REQUERIMIENTOS

Requerimientos: Los requerimientos especifican qué es lo que el sistema debe hacer (sus funciones) y sus propiedades esenciales y deseables. La captura de los requerimientos tiene como objetivo principal la comprensión de lo que los clientes y los usuarios esperan que haga el sistema. Un requerimiento expresa el propósito del sistema sin considerar cómo se va a implementar. En otras palabras, los requerimientos identifican el qué del sistema, mientras que el diseño establece el cómo del sistema.

Análisis de requerimientos: Es el conjunto de técnicas y procedimientos, los cuales permiten conocer los elementos necesarios para definir un proyecto de software. Es una tarea de ingeniería del software que permite especificar las características operacionales del software, indicar la interfaz del software con otros elementos del sistema y establecer las restricciones que debe cumplir el software.

REQUERIMIENTOS FUNCIONALES

Los requerimientos funcionales describen una interacción entre el sistema y su ambiente, el cómo debe responder el sistema ante determinado estímulo. Son declaraciones de los servicios que debe proporcionar el sistema, de la manera en que éste debe reaccionar a entradas particulares y de cómo se debe comportar en situaciones particulares.

| Identificador | Nombre | Descripción |
|---------------|------------------------------|--|
| RF1 | Control de usuarios | Registro, ingreso, activación y estatus de las cuentas |
| RF2 | Lectura de datos del formato | Llenar formulario para la creación de un formato |
| RF3 | Almacenamiento de datos | Guardar registros en la Base de Datos |
| RF4 | Consulta de formatos | Visualizar formatos guardados en la aplicación |
| RF5 | Creación de PDF | Construcción de documento a partir de los registros guardados y las plantillas |
| RF6 | Carga de imágenes | Subir al sistema las imágenes solicitadas para el formato o página |
| RF7 | Descarga de PDF | Obtención del archivo PDF generado |

REQUERIMIENTOS NO FUNCIONALES

Los requerimientos no funcionales describen una restricción sobre el sistema que limita nuestras elecciones en la construcción de una solución al problema. Restringen los servicios o funciones ofrecidas por el sistema. Incluyen restricciones de tiempo, el tipo de proceso de desarrollo a utilizar, fiabilidad, tiempo de respuesta, capacidad de almacenamiento.

| Identificador | Nombre | Descripción |
|---------------|------------------------------------|--|
| RNF1 | Diseño intuitivo | Interfaz fácil de usar por parte del usuario |
| RNF2 | Costos bajos de uso | Ahorro de datos en el uso de la aplicación |
| RNF3 | Multiplataforma | Compatibilidad con la mayor parte de los exploradores Web |
| RNF4 | Identidad gráfica acorde con el MB | Diseño personalizado basado en la imagen de Metrobús |
| RNF5 | Filtro de consultas | Consultas especializadas de acuerdo con la solicitud del usuario |

ACTORES DEL SISTEMA

Usuario directo: Empleados del Metrobús que se encuentren en campo, realizando las revisiones pertinentes para cada formato.

Usuario indirecto: Supervisores y el titular de la Jefatura de Unidad Departamental de Unidades de Transporte de Metrobús, que se encargan de verificar tanto las cuentas como aprobar ciertos formatos.

REGLAS DEL NEGOCIO

El Grupo de Reglas de Negocio (BRG por sus siglas en inglés) del Grupo de Gestión de Objetos (OMG por sus siglas en inglés), considera que una Regla de Negocio es una “declaración que define o restringe algún aspecto del negocio” (BRG, 2000). Igualmente, desde el punto de vista tecnológico Martin y Odell (1998) y Russel (1995) proponen que una regla de negocio es una restricción que opera sobre el sistema [26].

Para lograr el objetivo de este proyecto se han identificado las siguientes reglas de negocio:

| Regla del Negocio | Descripción |
|-------------------|--|
| RN1 | La aplicación web estará disponible en cualquier momento. |
| RN2 | Se reducirá el gasto de datos móviles. |
| RN3 | Llenado de formatos de manera digital. |
| RN4 | Validación de campos para verificar su correcto ingreso. |
| RN5 | Generación automática de archivos en formato PDF. |
| RN6 | Mantener estilos individuales de cada formato establecido. |

ANÁLISIS DE FACTIBILIDAD

A continuación, se describen los estudios de factibilidad: técnica, operacional y económica para contemplar los recursos humanos y materiales necesarios para que dicho proyecto pueda desarrollarse e implantarse, además, se obtendrá una estimación del costo real del proyecto para que pueda realizarse durante el periodo que se ha establecido.

ANÁLISIS DE FACTIBILIDAD TÉCNICA

En este apartado abordaremos los recursos de hardware y software necesarios para que el proyecto pueda desarrollarse e implantarse, tomando en cuenta que dichos recursos son necesarios para que el proyecto pueda funcionar de manera correcta.

| Recursos | Especificaciones | Cantidad |
|----------------------------------|---|----------|
| Laptop/Computadora Portátil | Procesador Intel Core i5-6300, 8GB RAM y SSD de 256GB | 3 |
| Smartphone | Sistema Operativo Android o iOS. | 3 |
| Servidor Web/Amazon Web Services | Ubuntu Server 20.04 LTS, 1.07GB RAM, t2.micro CPU. | 1 |

| Recurso | Especificación | Cantidad |
|-------------------|---|----------|
| Sublime | Versión 3.2.2 y la extensión Babel. | 3 |
| Node.js | Versión 14.16.1 LTS. | 1 |
| React.js | Versión 17.0.2. | 1 |
| Express.js | Versión 5. | 1 |
| Microsoft Office | Versión 365. | 3 |
| Explorador Web | Microsoft Edge, Google Chrome, Mozilla Firefox, OSX Safari u Opera. | 3 |
| Sistema operativo | Windows 10 o cualquier distribución de Linux | 3 |

ANÁLISIS DE FACTIBILIDAD OPERATIVA

A continuación, se mencionarán los requisitos y habilidades que deben tener los integrantes del equipo que van a desarrollar el prototipo propuesto.

| Recurso | Habilidades y Características | Cantidad |
|------------------------|--|----------|
| Ingenieros en Sistemas | Que tengan el conocimiento de desarrollo de aplicaciones web, con lenguajes HTML, Javascript y CSS, además del uso de Bases de Datos y frameworks. | 3 |

ANÁLISIS DE FACTIBILIDAD ECONÓMICA

En este apartado se calcula el costo estimado de desarrollo del proyecto que será elaborado en un periodo de diez meses y se hará un presupuesto de costos y gastos del proyecto detallado en las siguientes tablas. Las estimaciones presentadas en las siguientes tablas están expresadas en pesos corrientes del año 2021.

| Costos y gastos de recursos humanos | | | | | |
|-------------------------------------|-------------|----------|----------------|--------------------|--------------|
| Nº | Puesto | Cantidad | Sueldo Mensual | Sueldo en 10 Meses | Total |
| 1 | Programador | 3 | \$20,000.00 | \$200,000.00 | \$600,000.00 |

| Costos y gastos de recursos materiales de hardware y software | | | | |
|--|-----------------------------|----------|--------------------|-------------|
| Nº | Concepto | Cantidad | Costo Unitario | Costo Total |
| 1 | Laptop/Computadora Portátil | 3 | \$11,000.00 | \$33,000.00 |
| 2 | Smartphone | 3 | \$5,000.00 | \$15,000.00 |
| 3 | Sublime | 3 | \$1,600.00 MXN | \$4,800.00 |
| 4 | Microsoft Office | 3 | \$173.00 | \$519.00 |
| Total | | | \$53,744.00 | |

| Gastos generales del proyecto | | | |
|--------------------------------------|------------------------|-------------|---------------------|
| Nº | Concepto | Valor mes | Valor en 10 meses |
| 1 | Renta para oficina | \$12,000.00 | \$120,000.00 |
| 2 | Electricidad | \$50.00 | \$500.00 |
| 3 | Teléfono e Internet | \$780.00 | \$7,800.00 |
| 4 | Hosting AWS y Dominio. | \$500.00 | \$5,000.00 |
| 5 | Limpieza | \$2500.00 | \$25,000.00 |
| Total | | | \$158,300.00 |

| Costos y gastos de recursos materiales del proyecto | | | |
|--|-----------|-------------------|-------------------|
| Nº | Concepto | Valor mes | Valor en 10 meses |
| 1 | Papelería | \$50.00 | \$500.00 |
| 2 | Otros | \$50.00 | \$500.00 |
| Total | | \$1,000.00 | |

| Concepto | Valor en 10 Meses |
|--|--------------------------|
| Nomina | \$600,000.00 |
| Recursos materiales de Hardware y Software | \$53,744.00 |
| Recursos materiales | \$1,000.00 |
| Gastos Generales | \$158,300.00 |
| Total | \$813,044.00 |

ANÁLISIS DE RIESGOS

El análisis de riesgo (también conocido como evaluación de riesgo o PHA por sus siglas en inglés: Process Hazards Analysis) es el estudio de las causas de las posibles amenazas, y los daños y consecuencias que éstas puedan producir. Este tipo de análisis es ampliamente utilizado como herramienta de gestión en estudios financieros y de seguridad para identificar riesgos (métodos cualitativos) y otras para evaluar riesgos (generalmente de naturaleza cuantitativa).

El primer paso del análisis es identificar los activos a proteger o evaluar. La evaluación de riesgos involucra comparar el nivel de riesgo detectado durante el proceso de análisis con criterios de riesgo establecidos previamente.

Para realizar un análisis de riesgos es necesario seguir los siguientes pasos:

- Identificar los peligros.
- Decidir quién puede ser dañado y como.
- Evaluar los riesgos y decidir las precauciones.
- Registrar sus hallazgos e implementarlos.
- Revisar su análisis y poner al día si es necesario.

| Riesgos | Descripción |
|---------|--|
| R1 | El servidor no se encuentra disponible. |
| R2 | La base de datos se corrompa. |
| R3 | El usuario utilice un navegador que no estuviese señalado en los requerimientos. |
| R4 | La conexión entre la base de datos y la aplicación no se logra. |
| R5 | Uno de los frameworks tiene problemas de compatibilidad. |
| R6 | El usuario no atienda el material de capacitación. |
| R7 | El prototipo no se termine a lo largo de TT1 y TT2. |
| R8 | Mala organización en el equipo de trabajo. |

Una vez que se identificaron los peligros entre los equipamientos de trabajo, lo próximo es estimar la posibilidad de que ocurran, esta posibilidad se toma de los miembros del equipo debido a que cada uno sabe las ocupaciones que se tienen que desarrollar y los inconvenientes que tienen la posibilidad de exponer. Además, que se identifican los peligros conforme al tipo y al grado de efecto que poseen.

TIPOS DE RIESGOS

- **Riesgo de rendimiento (Rr):** El grado de incertidumbre con el que el producto encontrará sus requisitos y se adecue para su empleo pretendido.
- **Riesgo de coste (Rc):** El grado de incertidumbre que mantendrá el presupuesto del proyecto.
- **Riesgo de soporte (Rs):** El grado de incertidumbre de la facilidad del software para corregirse, adaptarse y ser mejorado.
- **Riesgo de la planificación temporal (Rpt):** El grado de incertidumbre con que se podrá mantener la planificación temporal y de que el producto se entregue a tiempo.

El impacto se considera de la siguiente manera:

Catastrófico = 1, Crítico = 2, Marginal = 3 y Despreciable = 4.

| Riesgos | Categoría | Probabilidad | Impacto | Plan de contingencia |
|---------|-----------|--------------|---------|--|
| R1 | Rr | 20% | 1 | Ponerse en contacto con el representante de Amazon Web Services para trabajar en una solución. |
| R2 | Rs | 30% | 2 | Realizar copias de seguridad (Backup) de la base de datos de manera constante. |
| R3 | Rr | 10% | 4 | Ofrecer información sobre versiones exactas con las que la aplicación es compatible. |

| Riesgos | Categoría | Probabilidad | Impacto | Plan de contingencia |
|---------|-----------|--------------|---------|--|
| R4 | Rs | 30% | 2 | Verificar si el problema se encuentra dentro de la implementación o por parte del mismo Amazon Web Services. |
| R5 | Rs | 10% | 3 | Buscar si alguna de las versiones anteriores es compatible, o si existe una actualización que corrija el problema. |
| R6 | Rc | 50% | 2 | Se otorgarán manuales de usuario, videos y capacitación a través de video conferencia al personal de Metrobús. |
| R7 | Rpt | 50% | 1 | Se realizarán constantes reuniones entre los miembros del equipo para verificar los avances. |
| R8 | Rpt | 20% | 2 | Se realizarán las actividades acordadas de acuerdo con el cronograma propuesto. |

Una vez obtenidos los riesgos se evalúa el impacto y la probabilidad en la siguiente tabla:

| Probabilidad | Impacto | Despreciable | Marginal | Crítico | Catastrófico |
|--------------|---------|--------------|----------|---------|--------------|
| | | 4 | 3 | 2 | 1 |
| Raro | 1 | R3 | R5 | R8 | |
| Moderado | 2 | | | R2,R4 | R1 |
| Muy Probable | 3 | | | | R6,R7 |
| Seguro | 4 | | | | |

En las filas los riesgos se ordenan conforme a la probabilidad de ocurrencia y en las columnas se ordenan de acuerdo con su impacto, como resultado podemos ver los riesgos más peligrosos en la parte de rojo, esos son los que tendremos que impedir o responder para disminuir su impacto en el prototipo mediante los planes de contingencia.

PLANES DE CONTINGENCIA

En este apartado describiremos los planes de contingencia mediante una hoja de información, para ser tomados en cuenta en caso de que lleguen a ocurrir.

| Hoja de Información de Riesgo | | | |
|---|----------------------|------------------|-----------------------|
| ID: R1 | Fecha: 08/Marzo/2021 | Probabilidad:20% | Impacto: Catastrófico |
| Descripción: El servidor no se encuentra disponible. | | | |
| Refinamiento/Contexto: Subcondición 1: Amazon Web Service presenta fallos. Subcondición 2: El servidor está apagado o fuera de servicio. | | | |
| Reducción/Supervisión: Se verificará el estatus de los servicios de Amazon directamente. Se revisará el estatus del servidor, para confirmar que haya iniciado. | | | |
| Plan de contingencia/Acción: Se tendrá contacto con el soporte técnico de Amazon Web Services para buscar una solución, o en su caso, para corregir cualquier problema que haya con el servidor. | | | |

| Hoja de Información de Riesgo | | | |
|---|----------------------|------------------|-----------------------|
| ID: R3 | Fecha: 08/Marzo/2021 | Probabilidad:10% | Impacto: Despreciable |
| Descripción: La versión del navegador no soporta la aplicación. | | | |
| Refinamiento/Contexto: Subcondición 1: La versión del navegador está muy atrasada. Subcondición 2: El explorador no tiene habilitados los permisos necesarios. | | | |
| Reducción/Supervisión: Se tendrá que mantener actualizados los navegadores web. Se revisará que los permisos solicitados como descarga de archivos o ejecución de plugins están activados. | | | |
| Plan de contingencia/Acción: Buscar información sobre las versiones exactas con las que la aplicación es compatible. | | | |

| Hoja de Información de Riesgo | | | |
|--|----------------------|-------------------|------------------|
| ID: R2 | Fecha: 08/Marzo/2021 | Probabilidad: 30% | Impacto: Crítico |
| Descripción: La base de datos se corrompe. | | | |
| Refinamiento/Contexto: Subcondición 1: La base de datos se vuelve inaccesible. Subcondición 2: Registros/Tablas individuales están corruptos. | | | |
| Reducción/Supervisión: Se tendrá un sistema de copias de seguridad de la base de datos completa. Se verificará constantemente el estatus de la base original y sus respaldos. | | | |
| Plan de contingencia/Acción: Respaldar la base de datos de forma tanto manual como automática, además de una revisión constante de los cambios para llevar un control de versiones. | | | |

| Hoja de Información de Riesgo | | | |
|---|----------------------|------------------|------------------|
| ID: R4 | Fecha: 08/Marzo/2021 | Probabilidad:30% | Impacto: Crítico |
| Descripción: La conexión entre la base de datos y la aplicación no es posible. | | | |
| Refinamiento/Contexto: Subcondición 1: Tanto la aplicación como la base están activas y funcionan, pero no existe comunicación entre ellas. | | | |
| Reducción/Supervisión: Se revisará cuál es el problema dentro de la conexión, si la comunicación entre Back End y Base de Datos o el mismo estado de la base. | | | |
| Plan de contingencia/Acción: Investigar si el problema se encuentra dentro de la implementación realizada en la Base de Datos o si es por parte del mismo Amazon Web Services (AWS). | | | |

| Hoja de Información de Riesgo | | | |
|--|----------------------|------------------|-------------------|
| ID: R5 | Fecha: 08/Marzo/2021 | Probabilidad:10% | Impacto: Marginal |
| Descripción: Uno de los frameworks tiene problemas de compatibilidad. | | | |
| Refinamiento/Contexto: Subcondición 1: Al integrar alguno de los frameworks se crean incompatibilidades o hace inestable a la aplicación. | | | |
| Reducción/Supervisión: Verificar el funcionamiento de los frameworks de manera individual y en conjunto. | | | |
| Plan de contingencia/Acción: Buscar una versión que sea compatible con la aplicación web y/o un nuevo framework que sea compatible que cumpla con la misma funcionalidad que se necesita. | | | |

| Hoja de Información de Riesgo | | | |
|---|----------------------|------------------|-----------------------|
| ID: R7 | Fecha: 08/Marzo/2021 | Probabilidad:50% | Impacto: Catastrófico |
| Descripción: El prototipo no se termine a lo largo de TT1 y TT2. | | | |
| Refinamiento/Contexto: Subcondición 1: Conforme se acerca la fecha de entrega no cumple con lo estipulado en el cronograma. Subconjunto 2: No se atienden en tiempo y forma las observaciones indicadas por parte de los sinodales y directores. | | | |
| Reducción/Supervisión: Estar en constante contacto con los sinodales y directores proporcionando información sobre los avances que se tienen. | | | |
| Plan de contingencia/Acción: Se planificaron reuniones internas entre los integrantes del equipo para discutir sobre los avances que se tienen. | | | |

| Hoja de Información de Riesgo | | | |
|--|----------------------|------------------|------------------|
| ID: R6 | Fecha: 08/Marzo/2021 | Probabilidad:50% | Impacto: Crítico |
| Descripción: El uso de datos es excesivo. | | | |
| Refinamiento/Contexto: Subcondición 1: El uso de la aplicación en dispositivos móviles genera un alto costo en el uso de datos. | | | |
| Reducción/Supervisión: Se verificará qué funciones tienen el mayor uso de datos. Se llevará un control del uso de datos dentro de las pruebas para informar al usuario. | | | |
| Plan de contingencia/Acción: Se investigará y tratará de optimizar las funciones que realiza la aplicación. | | | |

| Hoja de Información de Riesgo | | | |
|---|----------------------|------------------|------------------|
| ID: R4 | Fecha: 08/Marzo/2021 | Probabilidad:20% | Impacto: Crítico |
| Descripción: Mala organización en el equipo de trabajo | | | |
| Refinamiento/Contexto: Subcondición 1: Alguno de los miembros no completa las tareas que se le fueron asignadas. Subcondición 2: El trabajo no es repartido de manera que optimice los tiempos de realización. | | | |
| Reducción/Supervisión: Se realizarán constantes juntas de equipo para verificar los avances. Se mantendrá comunicación con el equipo para reportar cualquier contratiempo. | | | |
| Plan de contingencia/Acción: Realizar todas las actividades conforme indica el cronograma. | | | |

Análisis del software

ANÁLISIS DE APLICACIONES WEB

Hoy en día cualquier persona con un dispositivo móvil, tablet o computadora puede acceder a Internet. Según datos de la Encuesta Nacional sobre Disponibilidad y Uso de Tecnologías de la Información en los Hogares de 2019, el 70.1% de la población de seis años o más en México es usuaria de Internet. 20.1 millones de hogares (56.4% del total nacional) disponen de conexión a Internet [29].

Una aplicación Web se trata de una aplicación que se ejecuta directamente en el navegador de cualquier dispositivo, por lo que no necesita instalación.

Su acceso se realiza normalmente a través de un enlace directo. Es decir, en principio en un navegador cualquiera se podrá abrir el sitio web que buscas. Posiblemente la aplicación web no es el diseño más eficiente y recomendado, pero todo dependerá del objetivo que busques con ella. Como todos los tipos de aplicaciones [30].

En este proyecto se pretende realizar una aplicación web para el Metrobús que permita el llenado de formatos establecidos a través de una interfaz web, enfocándose en la automatización de procesos, ya que el Metrobús necesita agilizar los procesos de llenado de los formatos que llevan el control de las Unidades de Transporte. Por lo que será necesario hacer un análisis de Frameworks para las aplicaciones web y la base de datos a utilizar de acuerdo con las necesidades del Metrobús para fin de este proyecto. A continuación de explican las ventajas del uso de los Frameworks que se utilizaron para la Aplicación Web.

FRAMEWORKS PARA APLICACIONES WEB

Con el término framework, nos estamos refiriendo a una estructura software compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación. En otras palabras, un framework se puede considerar como una aplicación genérica incompleta y configurable a la que podemos añadirle las últimas piezas para construir una aplicación concreta.

Los objetivos principales que persigue un framework son: acelerar el proceso de desarrollo, reutilizar código ya existente y promover buenas prácticas de desarrollo como el uso de patrones. Un framework Web, por tanto, podemos definirlo como un conjunto de componentes que construyen un diseño reutilizable que facilita y agiliza el desarrollo de aplicaciones Web.

A continuación se mostrará una tabla con los marcos de trabajo más comunes y una comparación entre ellos para después pasar a la descripción completa de la tecnología utilizada y el porqué.

| Marco De Trabajo Backend | Lenguaje De Programación | Descripción | Características |
|--------------------------|--------------------------|---|---|
| Django | Python | Django es un framework de desarrollo web de código abierto líder basado en el lenguaje de programación Python. Sigue el patrón del controlador de vista de modelo (MVC, por sus siglas en inglés). Django es adecuado para el desarrollo de sitios web basados en bases de datos sofisticados y ricos en funciones. | <ul style="list-style-type: none"> • Código abierto. • Sistema de denominación. • Sintaxis simple. • Arquitectura central MVC. |
| Spring Boot | Java | Es un marco de trabajo de aplicación de código abierto y el container de inversión de control de la plataforma Java. Las aplicaciones Java pueden utilizar las características principales de este marco de trabajo. | <ul style="list-style-type: none"> • Personalización de banners. • API de construcción fluida. • Vivacidad. |
| Laravel | PHP | Es un marco de trabajo web PHP de código abierto para desarrollar aplicaciones web basadas en Symfony que siguen la arquitectura modelo-vista-controlador (MVC). Ofrece un sistema de empaquetado modular equipado con un administrador de dependencias dedicado. | <ul style="list-style-type: none"> • Motor de plantillas. • Soporte de arquitectura MVC. • Mapeo Objeto-Relacional Eloquent. • Seguridad. |

| Marco De Trabajo Backend | Lenguaje De Programación | Descripción | Características |
|--------------------------|--------------------------|---|--|
| Asp .NET | C# | Es un marco de trabajo de código abierto y gratuito que sigue los pasos de ASP.NET, un backend ampliamente utilizado creado en asociación con .NET Foundation. ASP.NET Core es un marco de trabajo modular que se puede ejecutar en la totalidad de .NET Framework en Windows y .NET Core. | <ul style="list-style-type: none"> • Programación mínima. • Mantenimiento sencillo. • Alto rendimiento. • Multiplataforma. |
| ExpressJS | NodeJS | Es un marco de trabajo de aplicación web Node.js y un software de código abierto disponible bajo la licencia MIT. Se utiliza para crear API y aplicaciones web y se considera un marco de trabajo de servidor Node.js estándar. | <ul style="list-style-type: none"> • Programación rápida del servidor. • Enrutamiento. • Depuración. • Curva de aprendizaje suave. • Alto rendimiento. |
| Ruby on Rails | Ruby | Es un marco de trabajo de aplicación web del lado del servidor basado en Ruby con una licencia MIT. Rails es un marco de trabajo MVC que ofrece estructuras, páginas web y servicios web predeterminados de bases de datos. Ruby on Rails promueve el uso de estándares web como XML o JSON para transferencias de datos y CSS, JavaScript y HTML para la interfaz. | <ul style="list-style-type: none"> • Prioriza el uso de patrones de ingeniería de software. • Eficiencia de tiempo. • Arquitectura MVC. • Registro activo. • Pruebas simples. |

| Marco De Trabajo Backend | Lenguaje De Programación | Descripción | Características |
|--------------------------|--------------------------|--|--|
| Angular | TypeScript | Es un framework open-source desarrollado por Google para facilitar la creación y programación de aplicaciones web de una sola página, las webs SPA (Single Page Application). | <ul style="list-style-type: none"> • Evita escribir código repetitivo. • Inyección de dependencias. • Routing. • Separa frontend y backend. |
| Vue.js | JavaScript | Vue es un framework open source de JavaScript, el cual nos permite construir interfaces de usuarios de una forma muy sencilla. | <ul style="list-style-type: none"> • Es incremental. • Es progresivo. • Renderiza más rápido. |
| React.js | Javascript | Es una famosa biblioteca para la creación de interfaces web llamada React.js. Esto quiere decir que React no es en sí un framework, es únicamente una librería de gestión del renderizado. | <ul style="list-style-type: none"> • No define un flujo de trabajo. • Solo gestiona el renderizado. • Se basa en la programación funcional. • Utiliza un motor de templating flexible. • Integra librerías Javascript. • Es el más usado en el mercado. • La curva de aprendizaje es suave. |

REACT

React es una librería Javascript focalizada en el desarrollo de interfaces de usuario. Así se define la propia librería y evidentemente, esa es su principal área de trabajo. Sin embargo, lo cierto es que en React encontramos un excelente aliado para hacer todo tipo de aplicaciones web, SPA (Single Page Application) o incluso aplicaciones para móviles. Para ello, alrededor de React existe un completo ecosistema de módulos, herramientas y componentes capaces de ayudar al desarrollador a cubrir objetivos avanzados con relativamente poco esfuerzo. Para fines de este proyecto, React nos da la flexibilidad necesaria para que la aplicación no solamente tenga una estética profesional sin la necesidad de escribir hojas de estilo individuales, sino que también funge como herramienta para el ahorro de datos, debido a su naturaleza SPA y que recicla componentes conforme se van necesitando, evita que la página completa sea recargada y cada vez que se acceda a una nueva, necesite de descargar todos los elementos de nuevo, como suele ocurrir con aplicaciones web convencionales que solo utilizan HTML, Javascript y hojas de estilo CSS, sin mencionar que facilita mucho el desarrollo, ya que nos ofrece muchas cosas ya listas, en las que no necesitamos invertir más tiempo para elaborarlo.

EXPRESS JS

Express.js es un framework rápido, minimalista y flexible de Node.js. Permite crear APIs y aplicaciones web fácilmente, provee un conjunto de características como manejo de rutas (direcciónamiento), archivos estáticos, uso de motor de plantillas, integración con bases de datos, manejo de errores, middlewares entre otras.

Con este proyecto, Express es la mejor opción porque posee métodos para especificar qué función ha de ser llamada dependiendo del verbo HTTP usado en la petición (GET, POST, SET, etc.) y la estructura de la URL (“ruta”). También tiene los métodos para especificar qué plantilla (“view”) o gestor de visualización utilizar, donde están guardadas las plantillas de HTML que han de usarse y cómo generar la visualización adecuada para cada caso. Puede utilizarse además cualquier sistema de trabajo con bases de datos, que sea soportado por Node.

BASE DE DATOS

En el desarrollo de aplicaciones de software una de las piezas fundamentales son las bases de datos, el modelo más habitual para el almacenamiento, con persistencia, de la información que manejan las aplicaciones.

Existen diversos sistemas de bases de datos, creados por distintos fabricantes. Estos sistemas se les conoce generalmente como SGBD (Sistemas gestores de bases de datos) y básicamente nos permiten crear y administrar en ellos un número indeterminado de bases de datos. A los SGBD también se les conoce habitualmente como motores de bases de datos [34].

| Nombre Del Gestor De Base De Datos | Descripción | Ventajas |
|------------------------------------|---|--|
| MariaDB | Este SGBD es una derivación de MySQL que cuenta con la mayoría de las características de este e incluye varias extensiones. Nace a partir de la adquisición de MySQL por parte de Oracle para seguir la filosofía Open Source y tiene la ventaja de que es totalmente compatible con MySQL. | <ul style="list-style-type: none"> Aumento de motores de almacenamiento. Gran escalabilidad. Seguridad y rapidez en transacciones. Extensiones y nuevas características relacionadas. Ofrece medidas de seguridad más estrictas. El rendimiento es más rápido. Mejor asistencia al usuario. |
| MySQL | Es el sistema gestor de bases de datos relacional por excelencia. Es un SGBD multihilo y multiusuario utilizado en la gran parte de las páginas web actuales. Además es el más usado en aplicaciones creadas como software libre. | <ul style="list-style-type: none"> Facilidad de uso. Facilidad de instalación. Soporte multiplataforma. |
| SQLite | SQLite es un sistema de gestión de bases de datos relacional compatible con ACID, contenida en una relativamente pequeña biblioteca escrita en C. | <ul style="list-style-type: none"> El tamaño es menor que cualquier SGBD. Gran portabilidad. Gran rendimiento. Reúne los criterios ACID. |

| Nombre Del Gestor De Base De Datos | Descripción | Ventajas |
|------------------------------------|--|--|
| Redis | <p>Redis está basado en el almacenamiento clave-valor. Podríamos verlo como un vector enorme que almacena todo tipo de datos, desde cadenas, hashes, listas, etc.</p> <p>El principal uso de este SGBD es para el almacenamiento en memoria caché y la administración de sesiones.</p> | <ul style="list-style-type: none"> • No relacional. • Simplicidad. • Multiplataforma. |
| PostgreSQL | <p>Este sistema gestor de base de datos relacional está orientado a objetos y es libre, publicado bajo la licencia BSD.</p> | <ul style="list-style-type: none"> • Control de concurrencias. • Flexibilidad en cuanto a lenguajes de programación. • Multiplataforma. • Dispone de una herramienta. • Robustez. |
| Microsoft SQL Server | <p>Es un sistema gestor de bases de datos relacionales basado en el lenguaje Transact-SQL, capaz de poner a disposición de muchos usuarios grandes cantidades de datos de manera simultánea.</p> | <ul style="list-style-type: none"> • Soporte exclusivo por parte de Microsoft. • Escalabilidad, estabilidad y seguridad. • Posibilidad de cancelar consultas. |

| Nombre Del Gestor De Base De Datos | Descripción | Ventajas |
|------------------------------------|--|---|
| Oracle | <p>Tradicionalmente, Oracle ha sido el SGBD por excelencia para el mundo empresarial, considerado siempre como el más completo y robusto.</p> | <ul style="list-style-type: none"> • Estabilidad. • Multiplataforma. • Soporte de transacciones. |
| MongoDB | <p>Estamos ante el Sistema Gestor de Bases de Datos no relacionales (SGBD NoSQL) más popular y utilizado actualmente. MongoDB es un SGBD NoSQL orientado a ficheros que almacena la información en estructuras BSON con un esquema dinámico que permite su facilidad de integración.</p> | <ul style="list-style-type: none"> • No relacional. • Balanceo de carga. • Consultas ad hoc. • Open Source. |
| Cassandra | <p>Apache Cassandra se trata de un software NoSQL distribuido y basado en un modelo de almacenamiento de «clave-valor», de código abierto que está escrita en Java. Permite grandes volúmenes de datos en forma distribuida.</p> | <ul style="list-style-type: none"> • No relacional. • Multiplataforma. • Propio lenguaje de consultas. • Distribuido. |

MARIADB

MariaDB Server es una de las bases de datos relacionales de código abierto más populares. Está hecho por los desarrolladores originales de MySQL y se garantiza que seguirá siendo de código abierto. Es parte de la mayoría de las ofertas en la nube y es el predeterminado en la mayoría de las distribuciones de Linux.

Se basa en los valores de rendimiento, estabilidad y apertura, y MariaDB Foundation garantiza que las contribuciones serán aceptadas por méritos técnicos.

Debido a su naturaleza open source, sentencias comunes de tipo MySQL y en general, la compatibilidad con los servidores que provee Amazon Web Service, MariaDB es la mejor opción como SGBD para este proyecto.

SEGURIDAD

A nivel aplicación, la seguridad tiene como propósito evitar o impedir el robo o manipulación de datos dentro de la aplicación. Debe abarcar las consideraciones de seguridad que se deben tener en cuenta para desarrollar y diseñar aplicaciones, además de los sistemas y los enfoques para protegerlas después de distribuirlas.

La seguridad de las aplicaciones web puede incluir hardware, software y procedimientos que identifican o minimizan las vulnerabilidades de seguridad, como por ejemplo, un enrutador que impide que se vea la dirección IP de un ordenador desde internet ya es en sí una forma de seguridad de las aplicaciones. A nivel aplicación, estas medidas de seguridad que se implementan suelen integrarse al software, como en el caso de un cortafuegos o “firewall” que define estrictamente qué actividades están permitidas y cuáles están prohibidas.

Para el desarrollo de aplicaciones web pueden integrarse procedimientos de autenticación y autorización para asegurarse de que solo acceden los usuarios autorizados, los procedimientos de autenticación garantizan la identidad de los usuarios y se consigue obligando a los mismos a proporcionar un nombre de usuario y contraseña para iniciar sesión dentro de la aplicación. También, debido a que en las aplicaciones web, para poder acceder a los servicios el usuario debe acceder a través de un navegador web, desde servidores remotos, la información de entrada y de salida se tiene que transmitir por Internet, y esto la hace vulnerable si no se encuentra debidamente protegida.

Para nuestro trabajo terminal, en base a las solicitudes derivadas de los requisitos del usuario final, dividiremos la seguridad de nuestra aplicación en 2 partes:

- Software
 - Sistema de autenticación
 - Control de Cuentas
 - Codificación de imágenes
 - Encriptación de passwords por medio de bcrypt.
- Hardware
 - Control del acceso de red a las instancias, mediante la configuración de la VPC y los grupos de seguridad.
 - Gestión de las credenciales utilizadas para la conexión de instancias.
 - Gestión del sistema operativo invitado y el software implementado en dicho sistema.

SISTEMA DE AUTENTICACIÓN

Este es el proceso que debe seguir un usuario para tener acceso a los recursos de un sistema o aplicación. Este proceso indica una identificación (decirle al sistema quien es) y una autenticación (demostrar al usuario que si es esa persona). La autenticación por sí sola no verifica derechos de acceso del usuario, si no en el proceso de autorización [37].

Para este proyecto, el sistema de autenticación de usuarios se separa en tres partes:

- **Autenticación:** El proceso en el cual el usuario se identifica de forma inequívoca; es decir, sin duda o equivocación de decir quién es. La aplicación solicita el correo y la contraseña con la que el usuario fue registrado por primera vez. Esta contraseña es encriptada y guardada en la base de datos de manera que la aplicación verifique estas licencias cada vez que el usuario desea acceder a la misma, comparando su valor hash en vez de desencriptarla y vulnerarla. El tipo de autenticación será específicamente por conocimientos, es decir, basado en información que solo conoce el usuario.
- **Autorización:** El proceso por el cual la aplicación web autoriza al usuario identificado a acceder a determinados recursos de la aplicación. Una vez que se ha registrado, será necesario que una de las cuentas con permisos superiores apruebe la cuenta y la acepte. Esto evita que cualquier persona no autorizada pueda acceder a información confidencial.
- **Registro:** El proceso mediante el cual la aplicación registra todos y cada uno de los usuarios ligados a esta. Como se comentó anteriormente, no es posible ingresar a la aplicación solamente registrándose, es necesario que otro usuario verifique y la active primero.

CONTROL DE CUENTAS

La administración de cuentas de usuario es una parte esencial de la administración de acceso a una aplicación. La razón principal de este método es verificar la identidad de la persona que va a utilizar la aplicación web, y por otro lado, permitir la utilización personalizada de recursos y privilegios de acceso. Estas cuentas tienen diferentes componentes, como el nombre de usuario, contraseña y datos necesarios para la identificación del personal dentro del sistema como el nombre completo.

El control que se le dará a estas cuentas es que inicien en estado desactivado, solamente cuentas con permisos superiores pueden verificar estas cuentas y activarlas dentro de la aplicación web.

CODIFICACIÓN DE IMÁGENES

Base64 es un tipo de codificación diseñado para que datos binarios sobrepasen capas de transporte que no son de 8 bits. La información codificada con este sistema dará como resultado una secuencia de letras, que constituye un archivo y puede ser decodificado y mostrado correctamente por cualquier navegador web. Este es un sistema de numeración posicional que usa 64 como base.

En este proyecto se utiliza para que el almacenamiento de imágenes se vuelve más sencillo, debido a que la transferencia de imágenes como archivos a un almacenamiento regular, es decir, el servidor o una especie de Docker, es complicado y presenta más retos, en su lugar, utilizaremos este método de codificación donde se transfiere una cadena de caracteres en lugar del archivo completo.

La segunda es que la misma codificación de los datos permite asegurarnos que en ningún momento se puede ver comprometida la transferencia de imágenes, es decir, que algún tipo de ataque les permita a personas externas obtenerlas de manera directa o detectarlas de manera fácil, debido a que solo se ve como una cadena más de datos al ser transferida.

CONTROL DEL ACCESO DE RED A LAS INSTANCIAS MEDIANTE LA CONFIGURACIÓN DE LA VPC Y LOS GRUPOS DE SEGURIDAD.

Amazon Virtual Private Cloud (Amazon VPC) es un servicio que permite lanzar recursos de Amazon Web Services en una red virtual aislada de forma lógica que el administrador define. De esta manera se puede controlar todos los aspectos del entorno de red virtual, como la selección de un rango de direcciones IP, la creación de subredes y la configuración de tablas de enrutamiento y gateway de red. Es posible utilizar tanto IPv4 como IPv6 para la mayoría de los recursos de la nube virtual privada, lo que ayuda a garantizar el acceso seguro y fácil a los recursos y las aplicaciones [40].

Para este proyecto, nos permite controlar cualquier acceso a la red virtual y el entorno en el que se encuentra, todas estas características avanzadas que nos proporciona de manera nativa el uso de los servicios de Amazon nos permiten tener una barrera más de seguridad que puede evitar una entrada no reconocida y por lo tanto, no autorizada. En general, esta herramienta nos ofrece más posibilidades de control de la red del sistema.

GESTIÓN DE LAS CREDENCIALES UTILIZADAS PARA LA CONEXIÓN DE INSTANCIAS.

Cada comando de Amazon Web Services debe incluir un conjunto de credenciales, que se utilizan para firmar criptográficamente la solicitud del servicio web correspondiente. Es posible especificar las credenciales por comando, por sesión o para todas las sesiones.

El administrador del sistema, que en este caso somos nosotros y una vez que el sistema sea entregado al cliente final, un nuevo administrador es designado, debe tener acceso a un par de claves o credencial de acceso para poder conectarse a la instancia a través de la red, esto le da más formalidad a la administración y la deja en un ambiente controlado. Estas llaves, o par de llaves, constan de una privada y una pública, es un set de credenciales seguras que se utilizarán para demostrar la identidad del administrador al conectarse a la instancia, en lugar de una contraseña que puede ser vulnerada.

GESTIÓN DEL SISTEMA OPERATIVO INVITADO Y EL SOFTWARE IMPLEMENTADO EN DICHO SISTEMA.

Los asuntos relacionados con la seguridad son compartidos entre Amazon Web Services y el cliente. Este modelo alivia la carga operativa del administrador, debido a que el otro opera, administra y controla los componentes del sistema operativo host y la capa de virtualización hasta la seguridad física de las instalaciones en las que funcionan los servicios. El cliente asume la responsabilidad y la administración del sistema operativo invitado (incluidas las actualizaciones y los parches de seguridad), de cualquier otro software de aplicaciones asociado y de la configuración del firewall del grupo de seguridad que ofrece.

Esto nos beneficia de dos maneras, la primera es que como se mencionó anteriormente la responsabilidad es dividida y no recae por completo ni en el sistema operativo del cliente ni en el anfitrión, por lo tanto si uno falla aún puede evitar problemas de seguridad su contraparte, y por otro lado, tenemos la seguridad de que el software utilizado en la instancia está actualizado en todo momento, sin la intervención del usuario.

ENcriptación de passwords por medio de bcrypt.

Bcrypt es una función de hashing de passwords diseñado por Niels Provos y David Maxieres, basado en el cifrado de Blowfish. Lo cual es un cifrado de bloque simétrico que se puede utilizar como reemplazo directo de DES o IDEA. Se necesita una clave de longitud variable, desde 32 bits hasta 448 bits. Se usa por defecto en sistemas OpenBSD y algunas distribuciones Linux y SUSE. Lleva incorporado un valor llamado salt, que es un fragmento aleatorio que se usará para generar el hash asociado a la passwords, y se guardará junto con ella en la base de datos. Así se evita que dos passwords iguales generen el mismo hash y los problemas que ello conlleva [43].

Para esta implementación, hemos decidido utilizar bcrypt ya que con la ayuda de esta librería podemos generar el hash para los passwords de nuestro sistema. Podemos elegir el valor del saltRound, lo que nos permite tener el control sobre el coste del procesamiento del dato. Cuanto más alto sea el valor que escogimos mayor tiempo se requerirá una máquina para calcular el hash. De igual manera tomara un mayor tiempo si alguien desea encontrar las passwords por medio de fuerza bruta.

SECCIÓN 8

DESARROLLO

| | | |
|---|----|---|
| Diseño técnico del sistema de información | | |
| 00 Selección del software | 00 | Servicios ofrecidos |
| 00 Modelo de datos | | Despliegue y configuración de los componentes |
| 00 Unidades de transporte | 00 | Instalación |
| 00 Diagrama entidad-relación | 00 | Configuración del servidor |
| 00 Casos de uso | 00 | Despliegue de la información |
| 00 Diagrama de actividades | 00 | Requisitos de los sistemas |
| 00 Estaciones y mantenimiento | 00 | Errores técnicos comunes |
| 00 Diagrama entidad-relación | | |
| 00 Casos de uso | | |
| 00 Diagrama de actividades | | |
| 00 Funcionalidad | | |
| 00 Unidades de transporte | | |
| 00 Estaciones y mantenimiento | | |

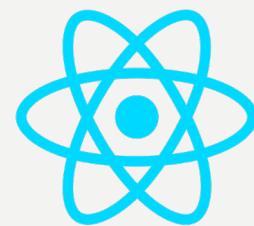
Diseño técnico del sistema de información

Selección del software

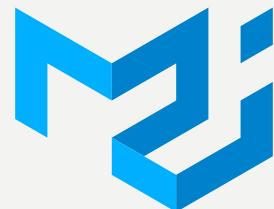
La selección finla del software y herramientas de software utilizadas durante el desarrollo del sistema fueron los y las siguientes:



Node JS
Framework back-end



React JS
Framework front-end



Material UI
Framework front-end



Amazon Web Services
Servicios de host



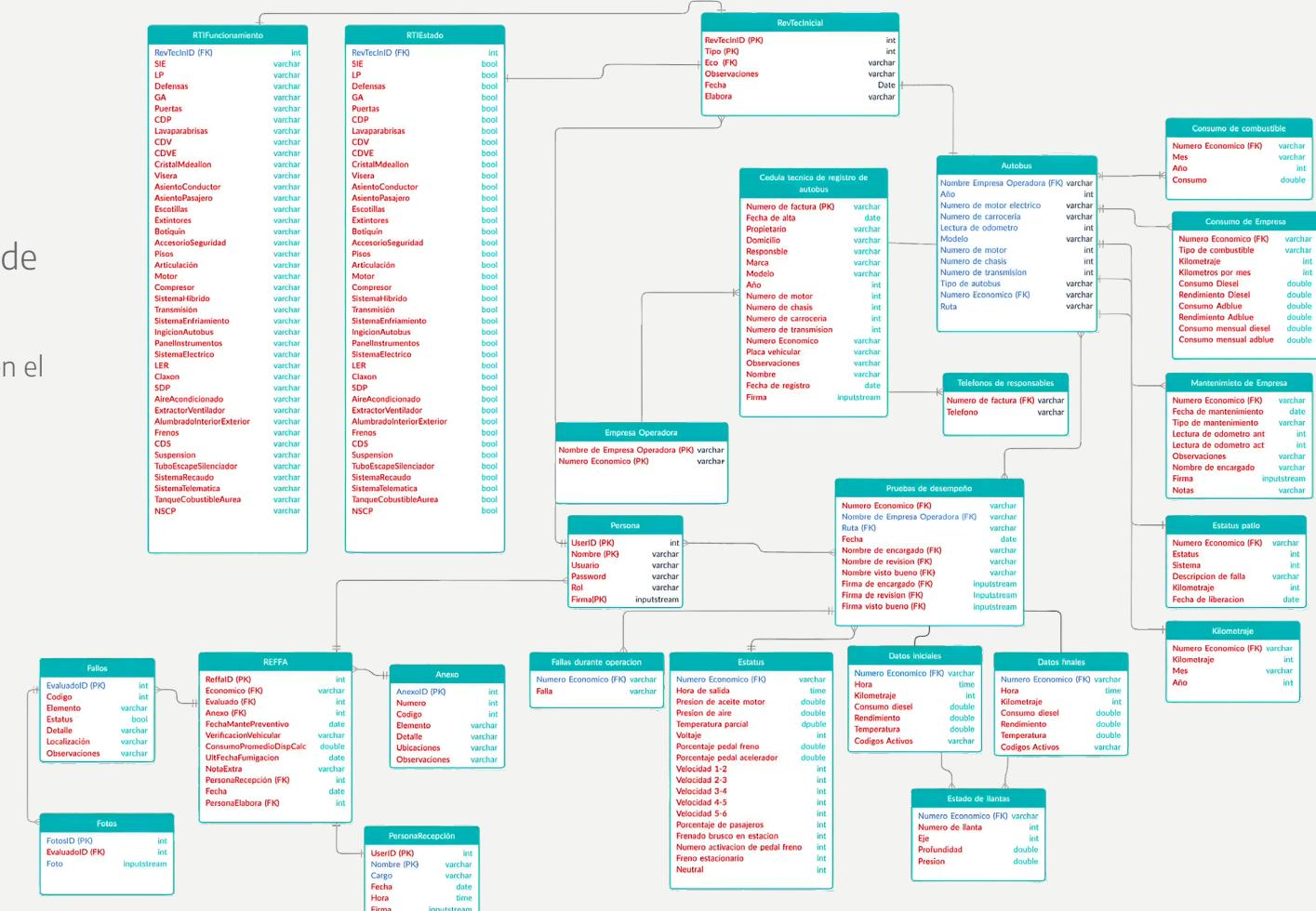
MySQL
Base de datos

Modelo de datos

Unidades de transporte

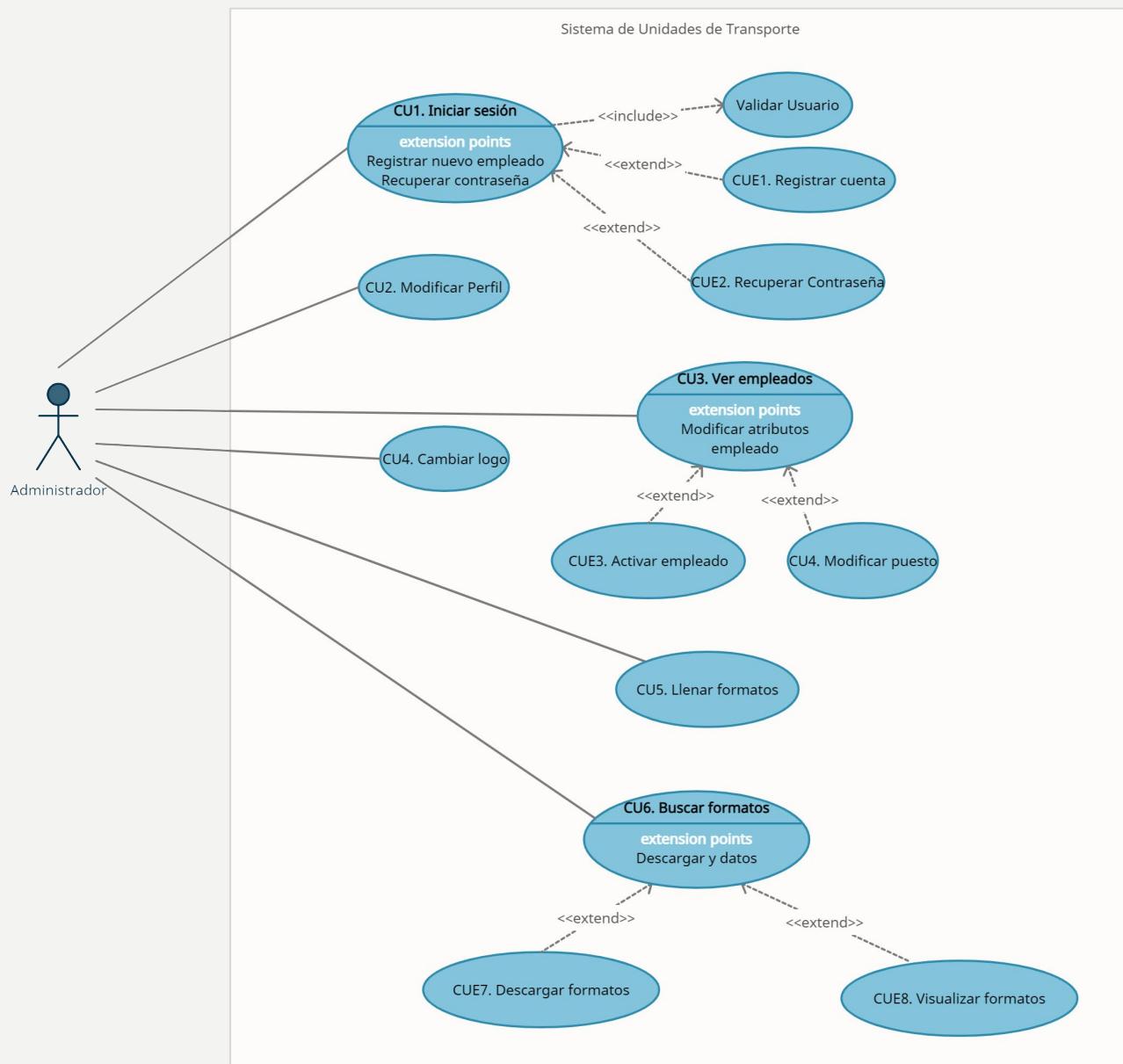
Diagrama Entidad-Relación de la base de datos.

El resumen detallado tabla a tabla se encuentra en el anexo, página 00.



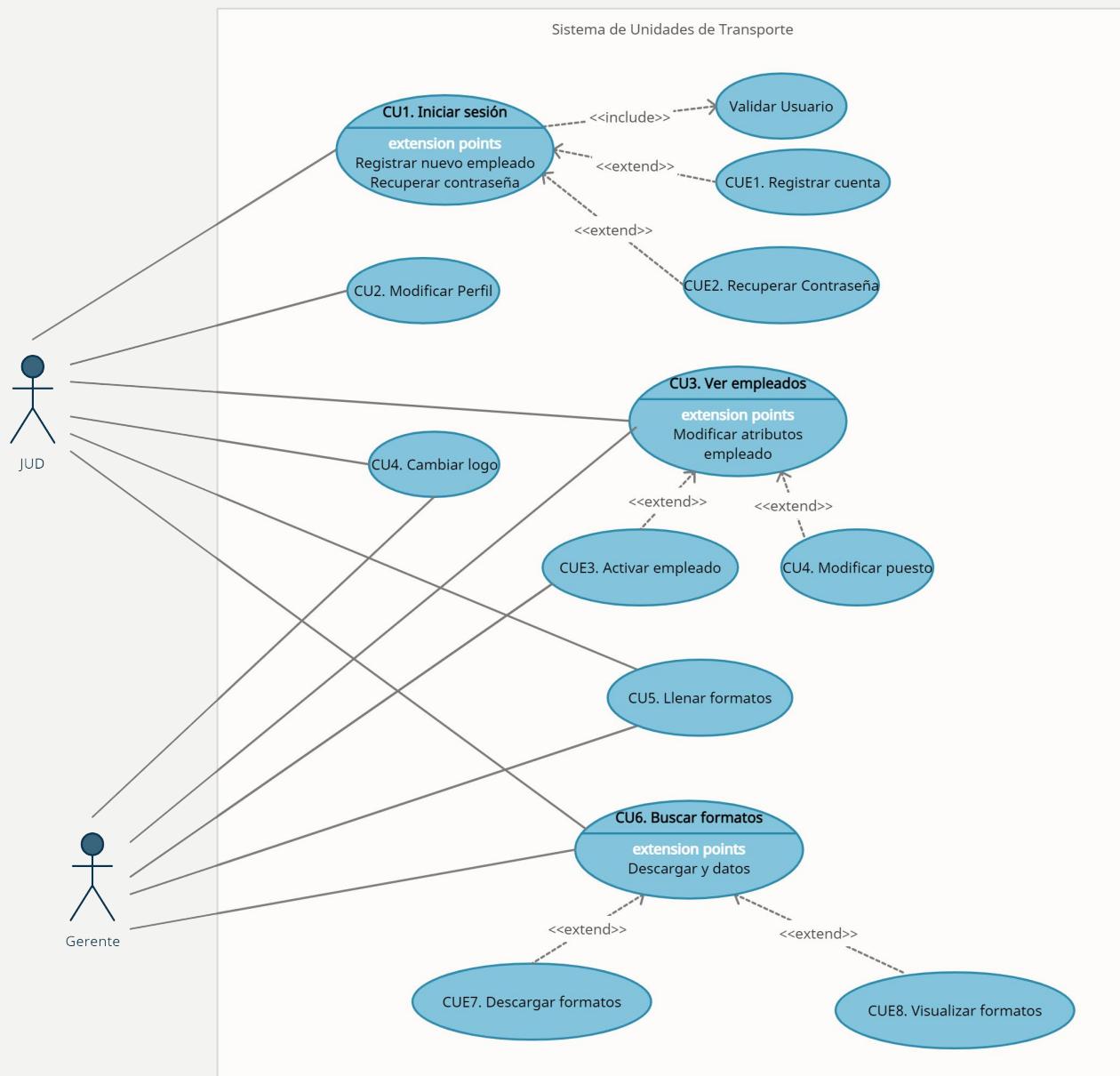
Casos de uso del sistema.

Administrador.



Casos de uso del sistema.

JUD y Gerente.



Casos de uso del sistema.

Supervisor y Auditor.

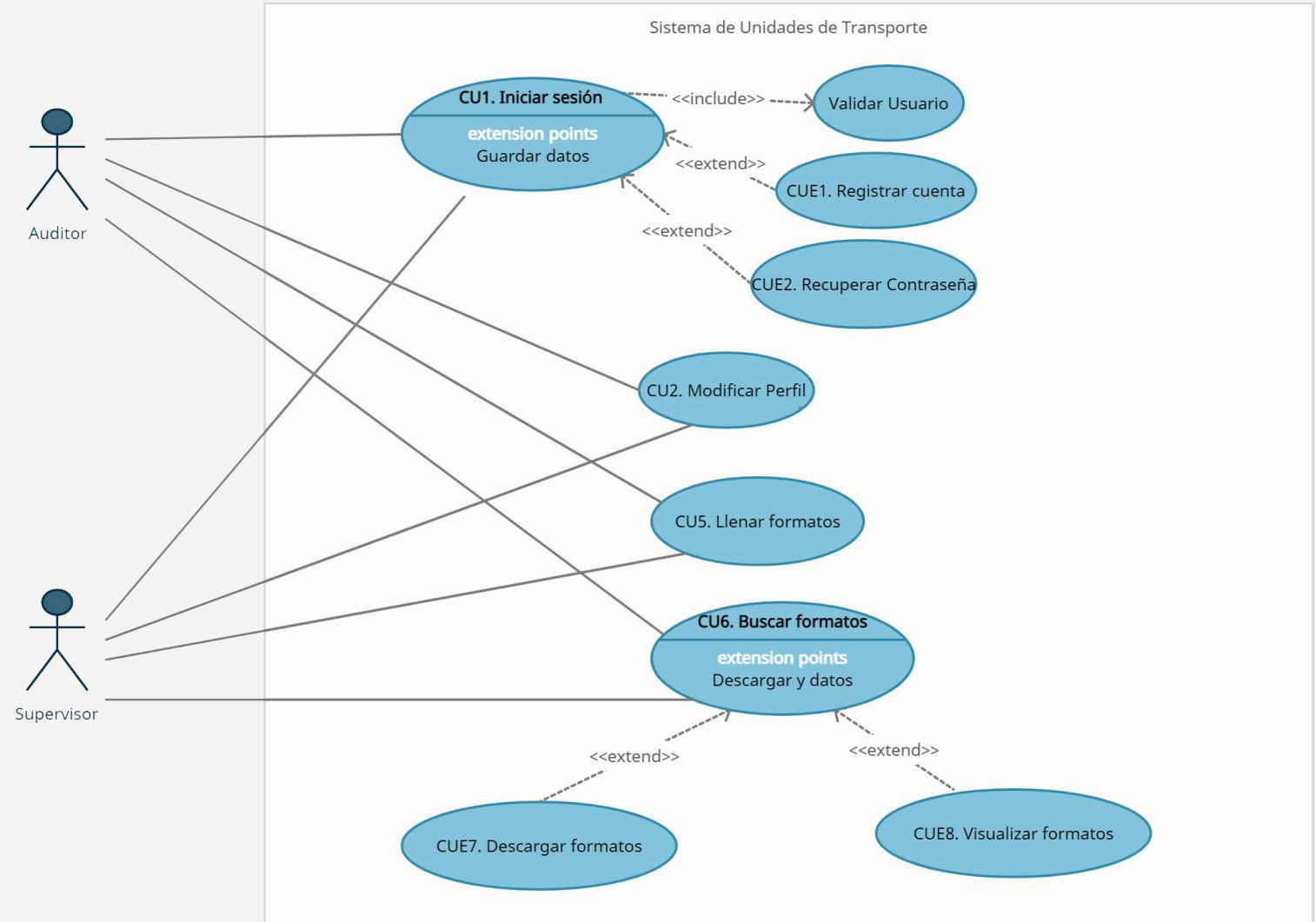


Diagrama de actividades.

Diagrama 1. Inicio de sesión.

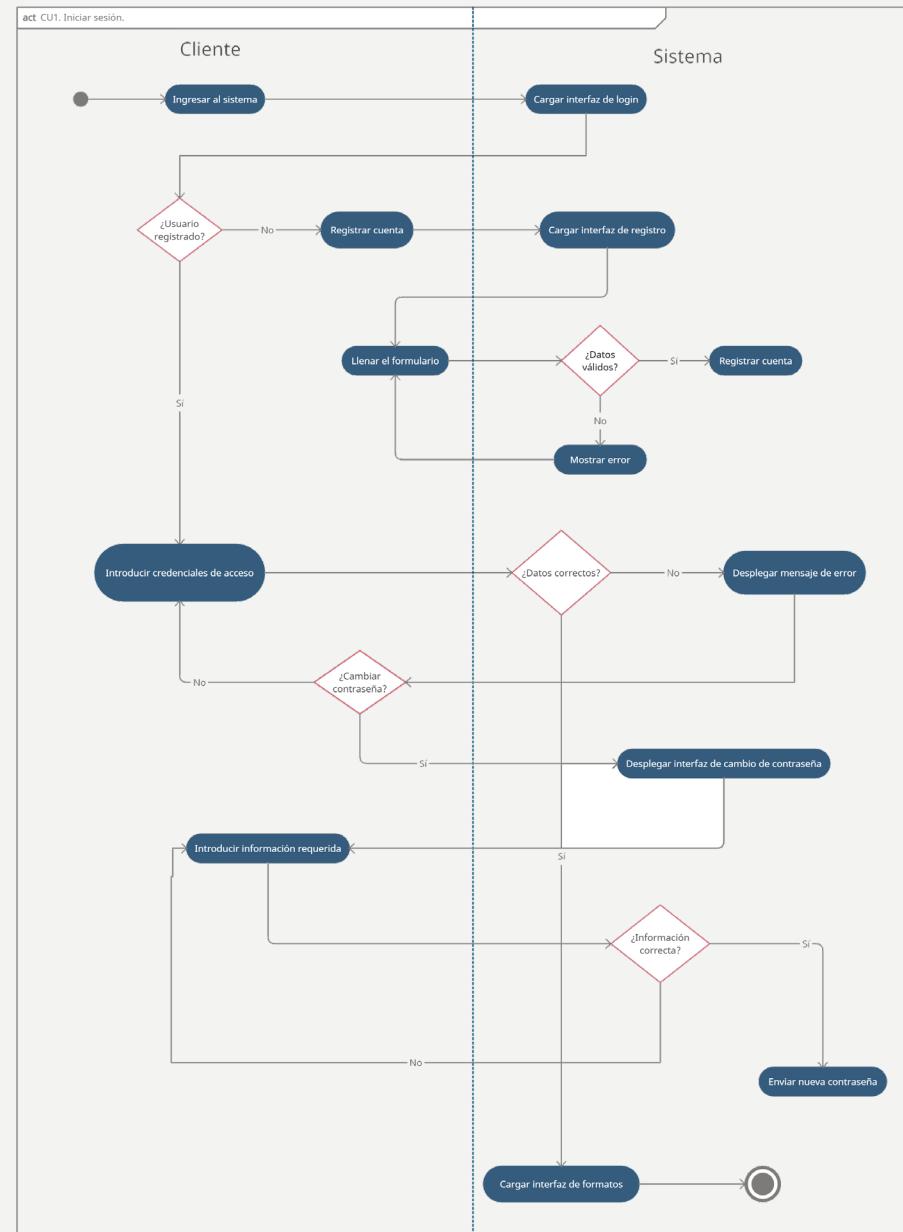


Diagrama de actividades.

Diagrama 2. Modificar perfil.

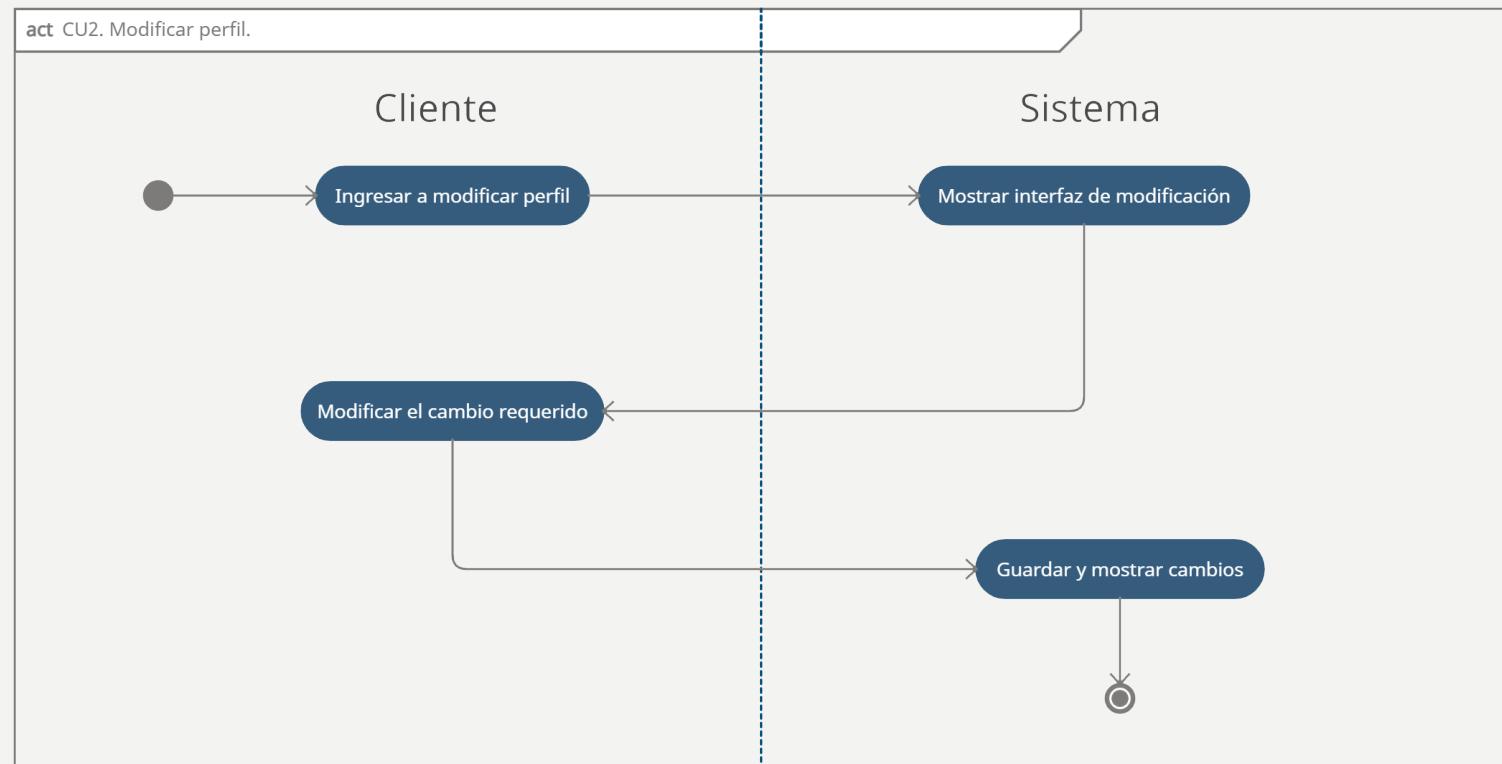


Diagrama de actividades.

Diagrama 3. Ver empleados.

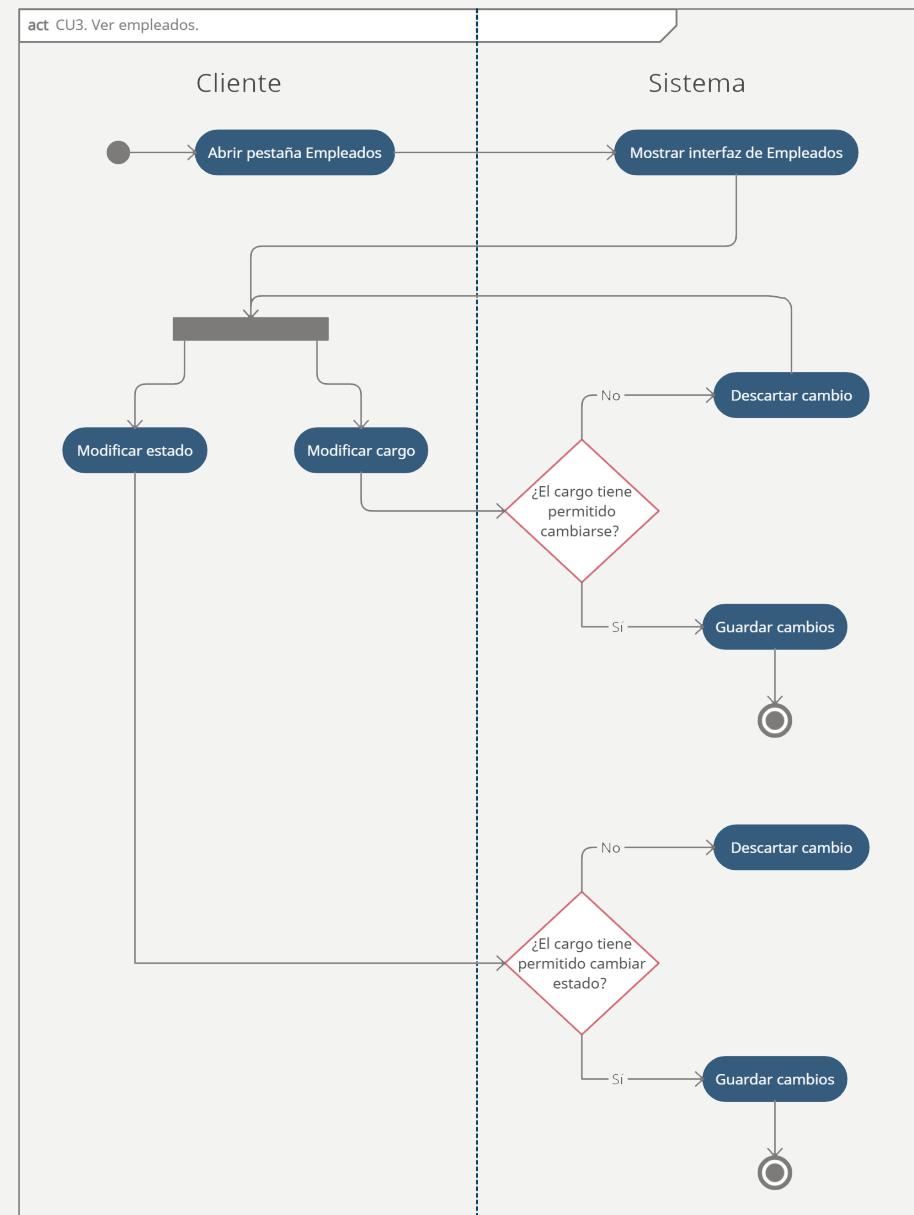


Diagrama de actividades.

Diagrama 4. Cambiar logo.

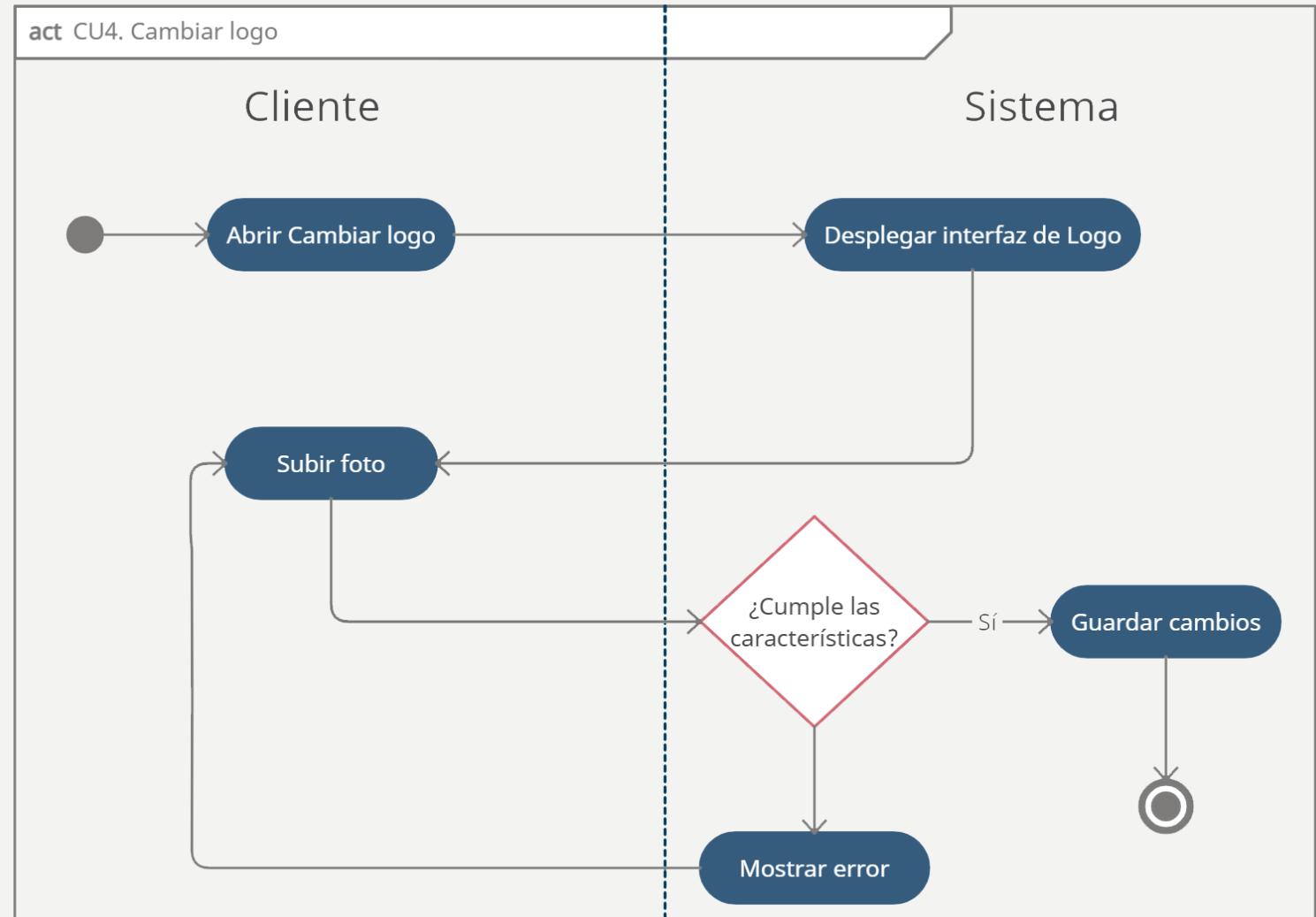


Diagrama de actividades.

Diagrama 5. Llenar formatos.

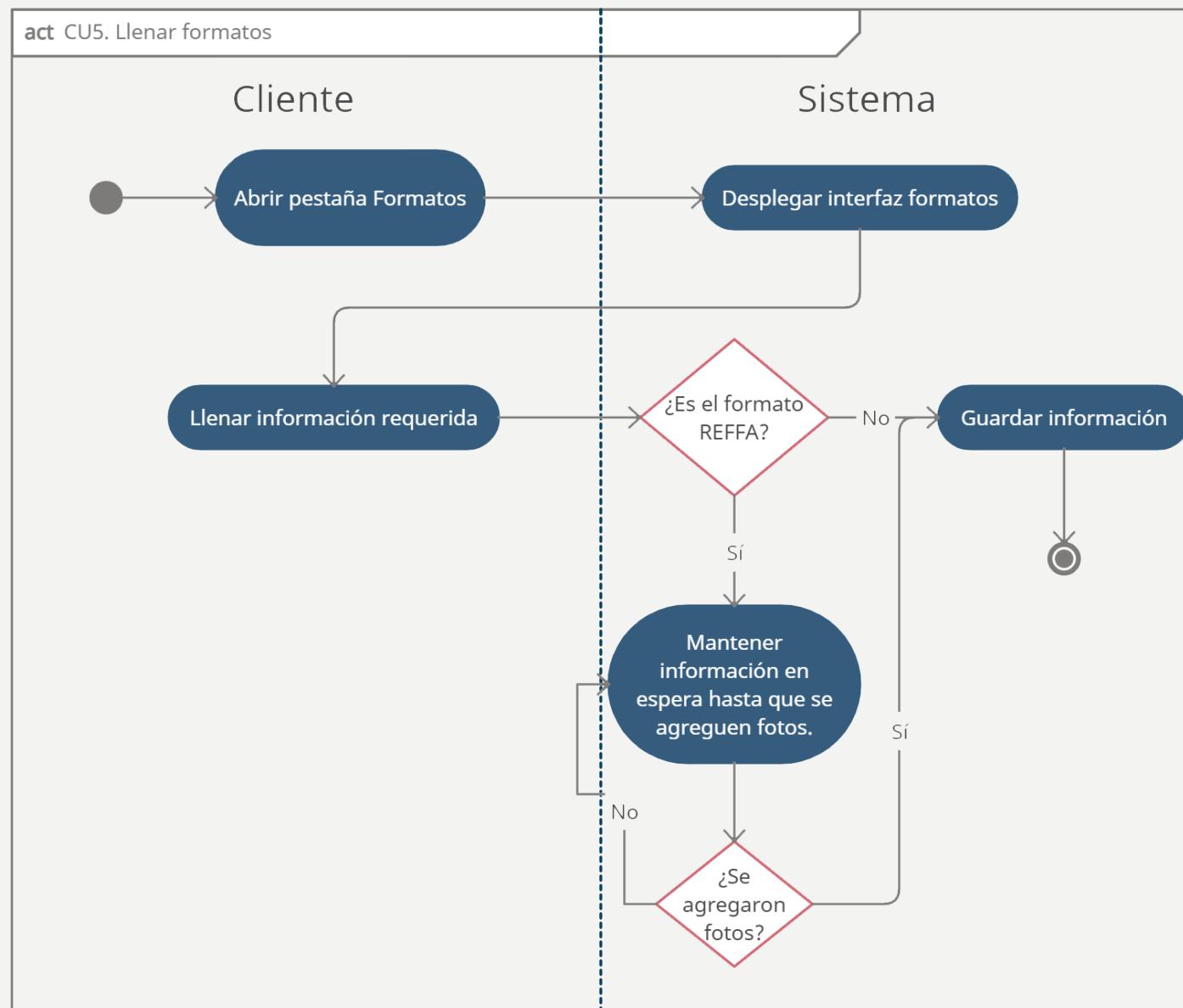
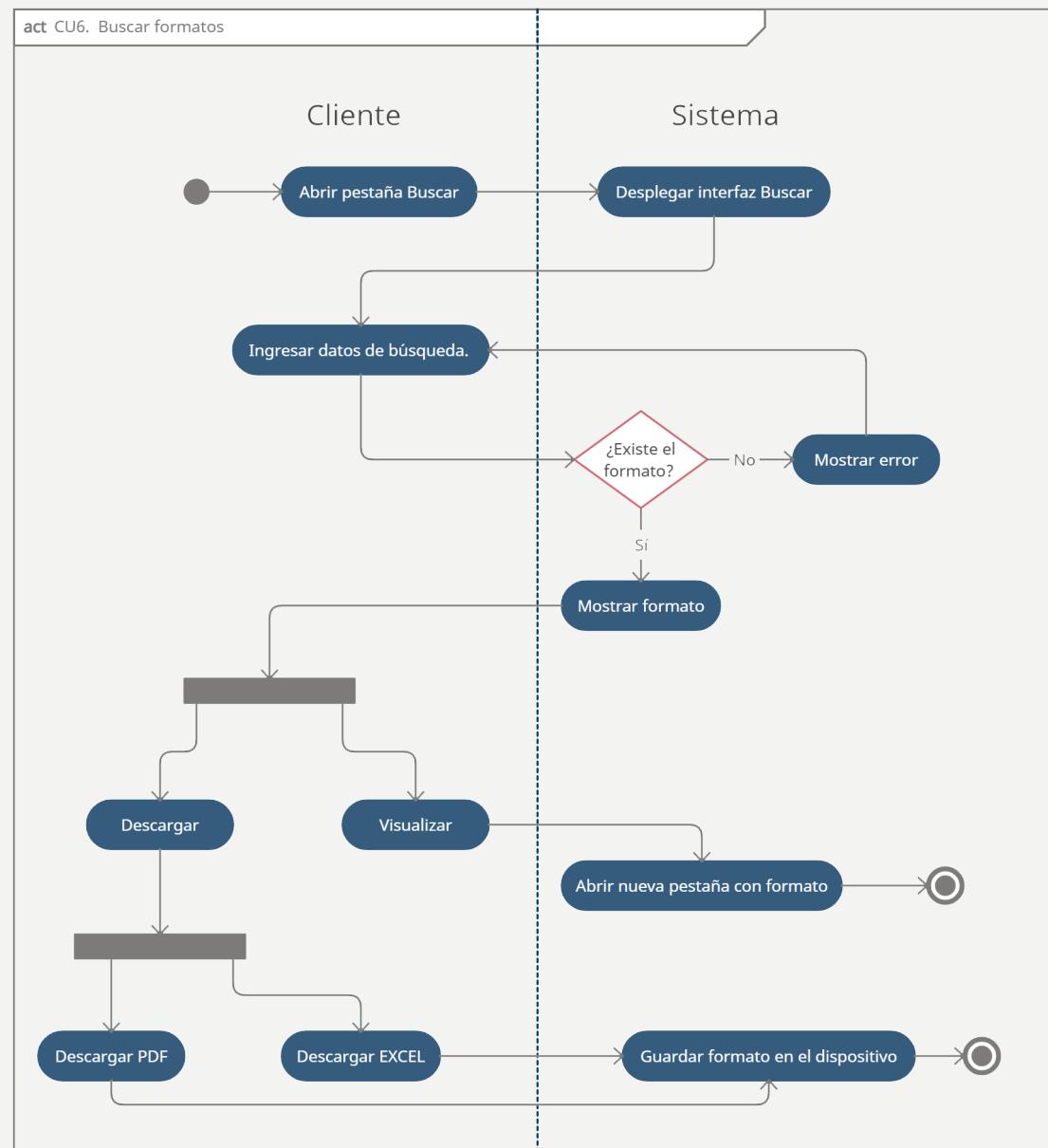


Diagrama de actividades.

Diagrama 6. Buscar formatos.



Estaciones y mantenimiento

Diagrama Entidad-Relación de la base de datos.

El resumen detallado tabla a tabla se encuentra en el anexo, página 00.

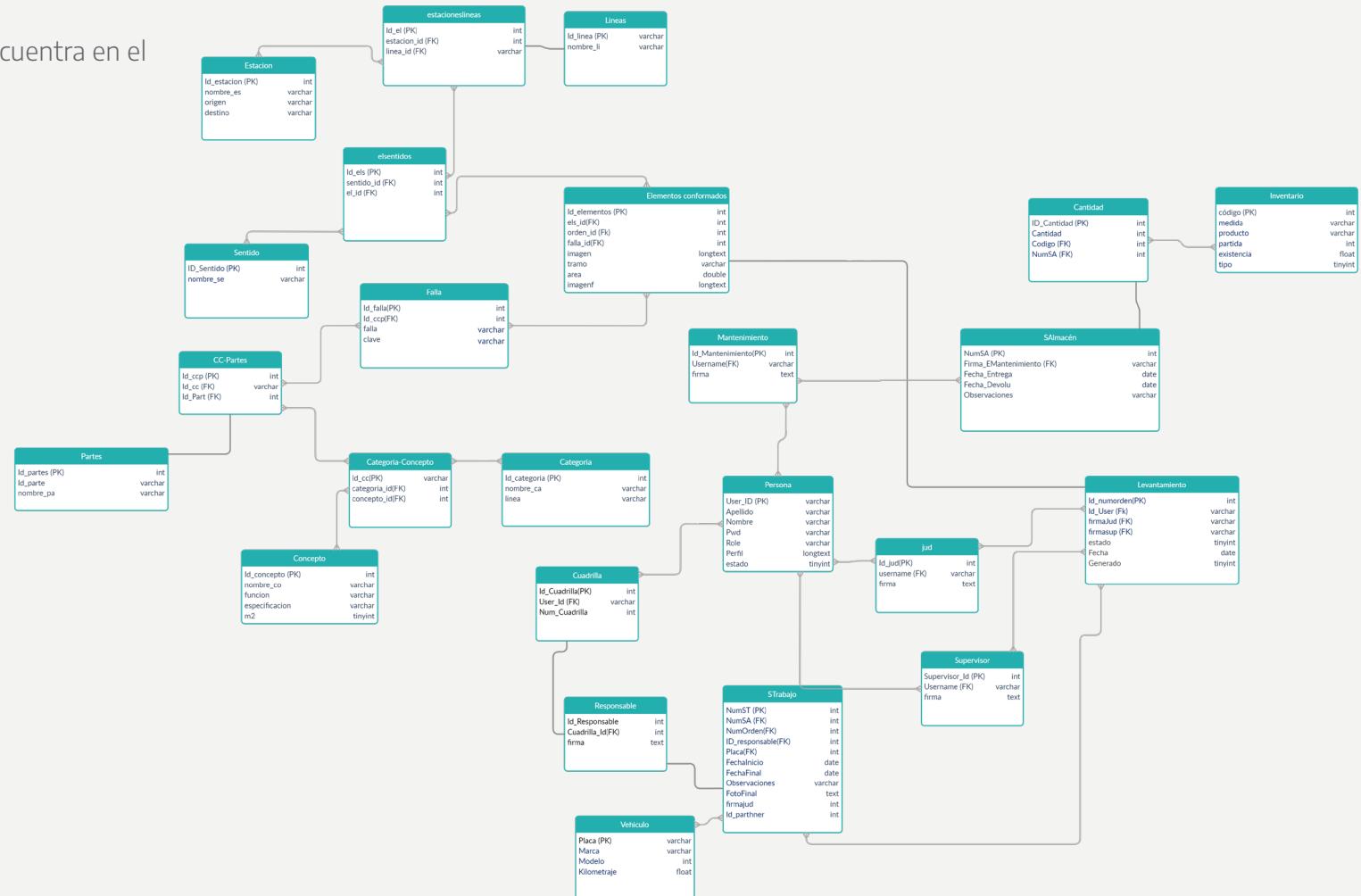


Diagrama de casos de uso.

JUD.

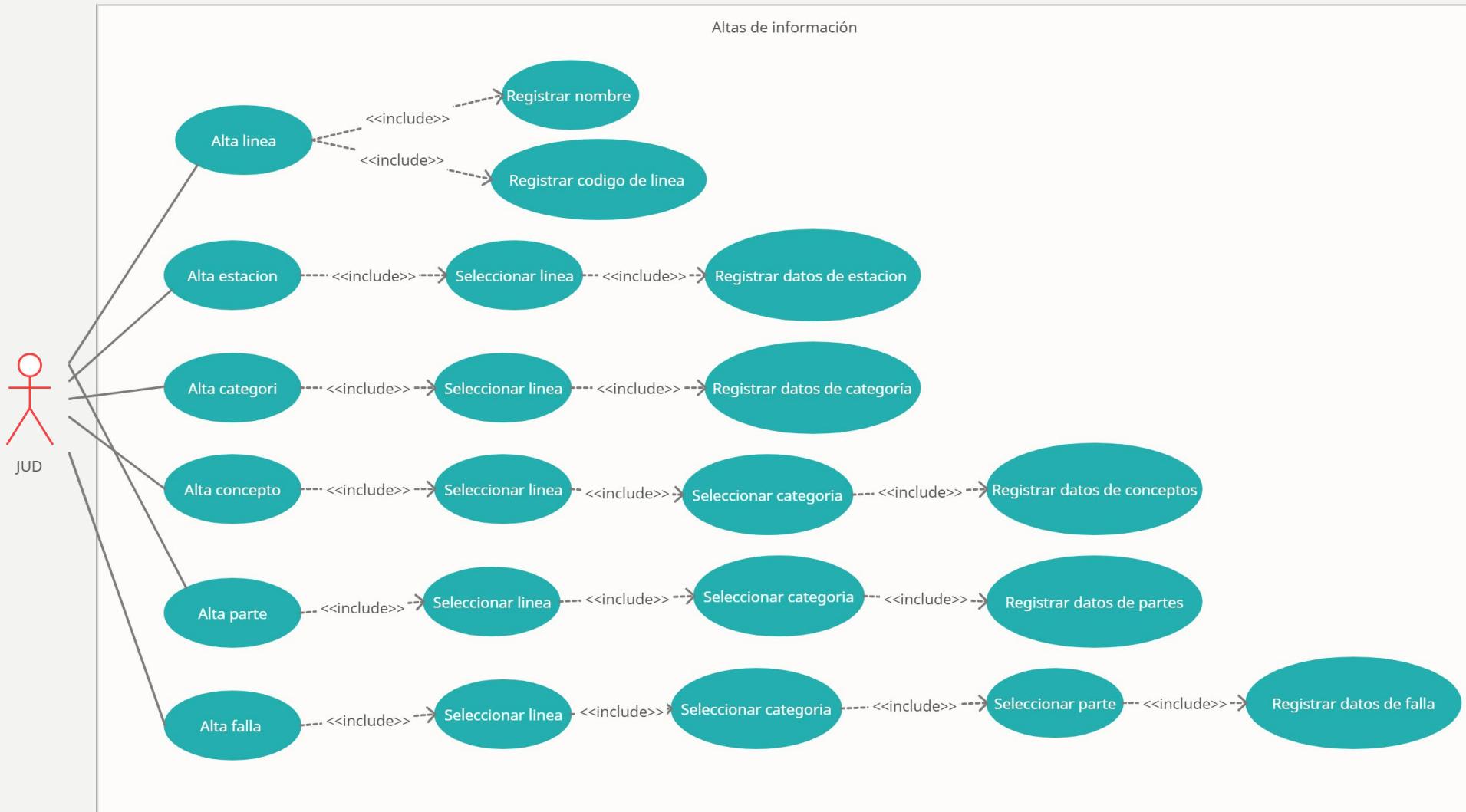


Diagrama de casos de uso.

JUD.

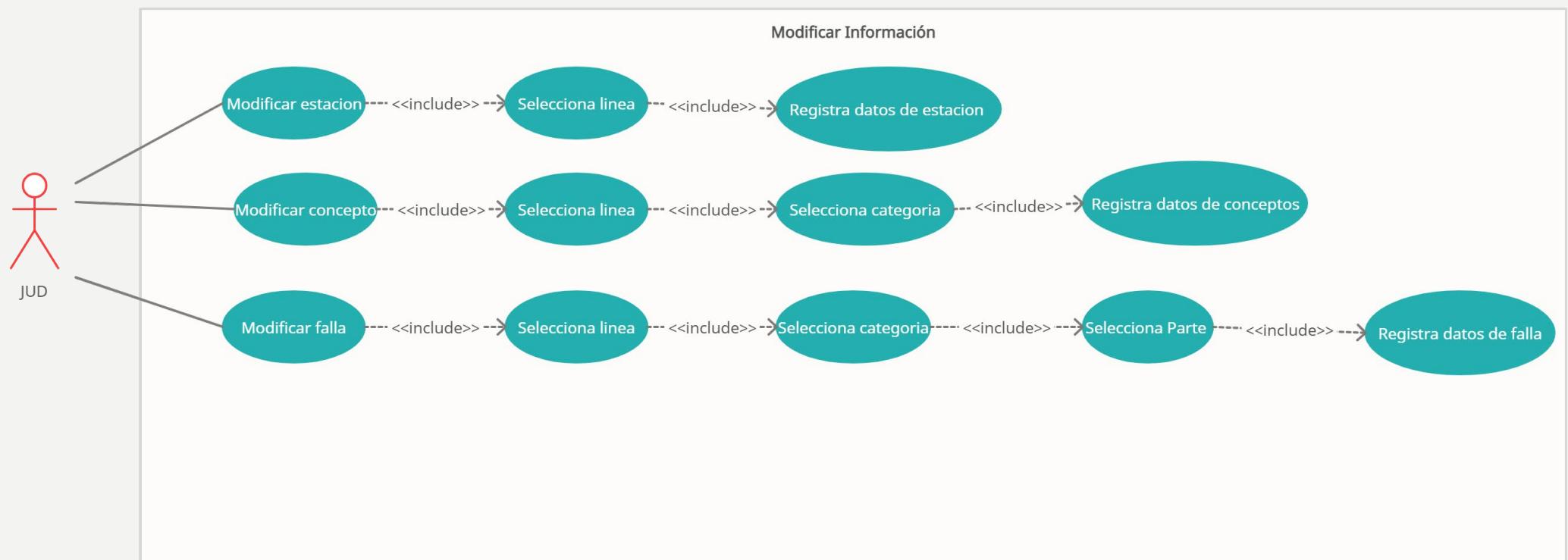


Diagrama de casos de uso.

JUD.

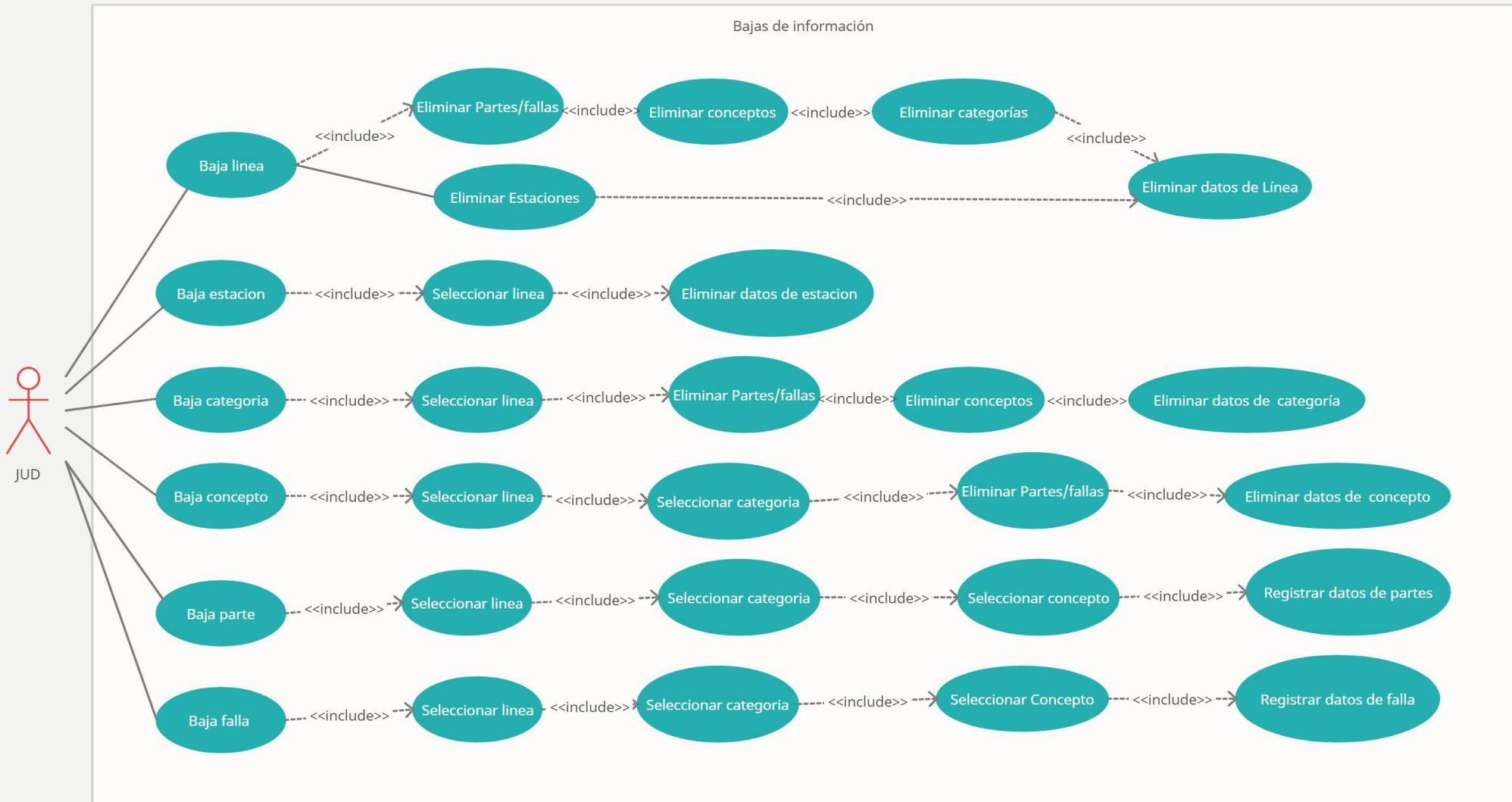


Diagrama de casos de uso.

JUD, Mantenimiento, Usuario.

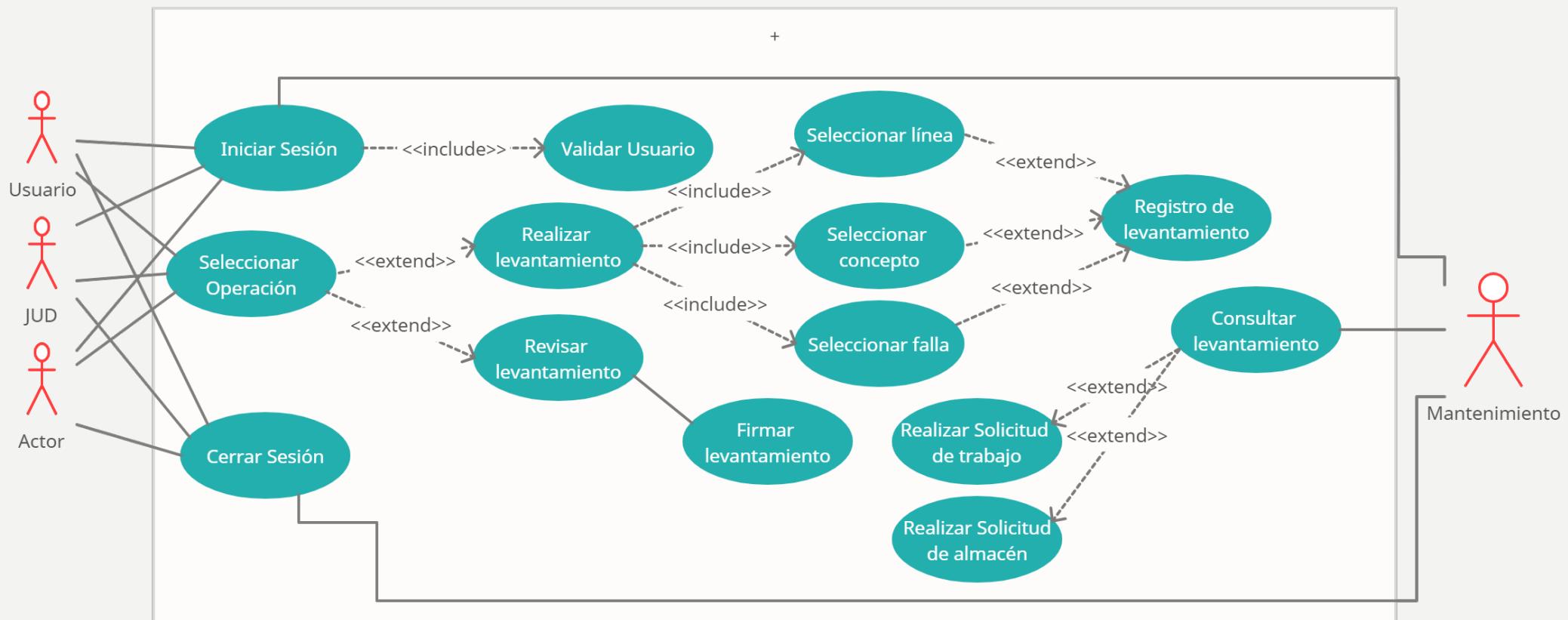


Diagrama de casos de uso.

JUD, Supervisor.

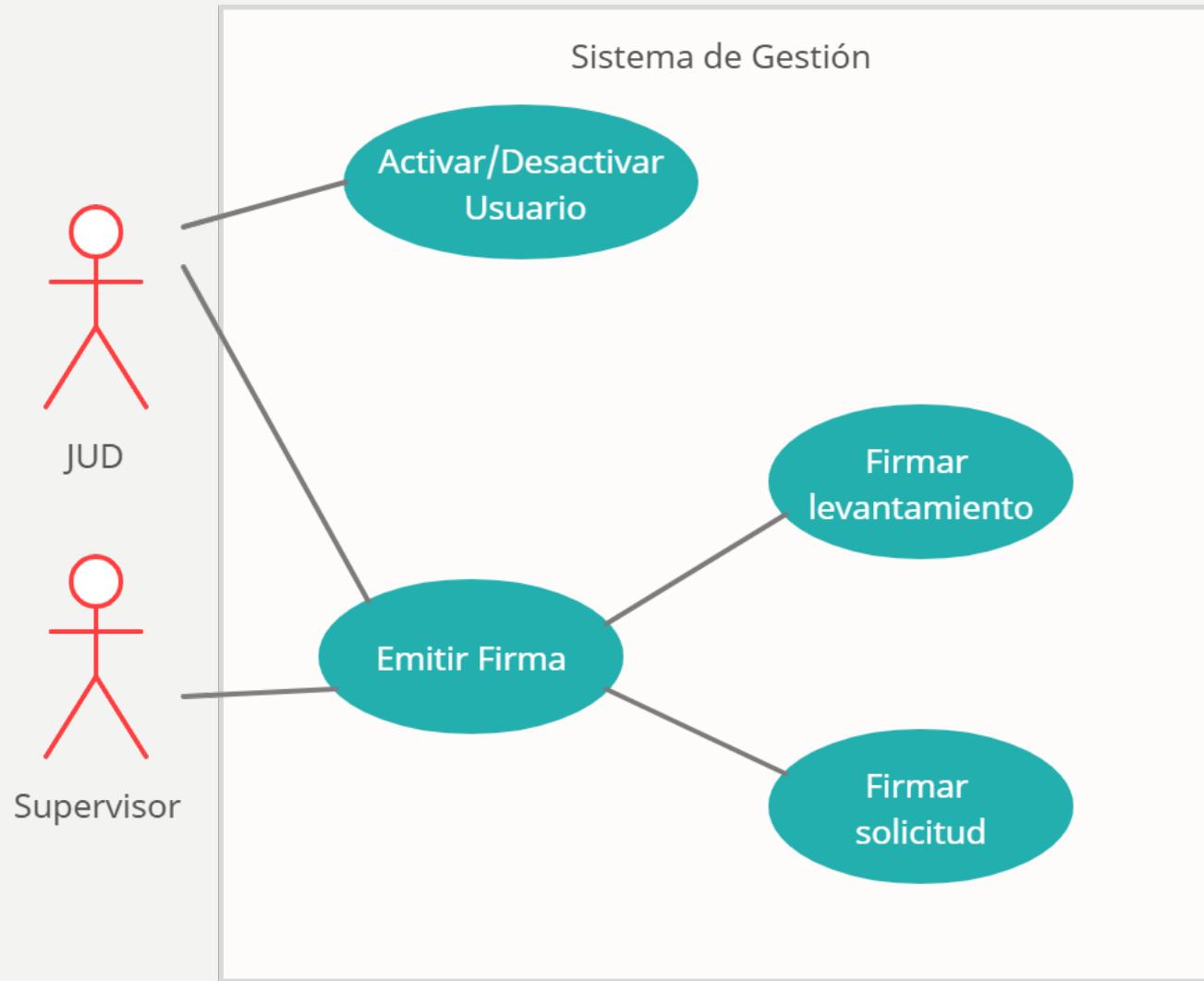


Diagrama de casos de uso.

Mantenimiento.

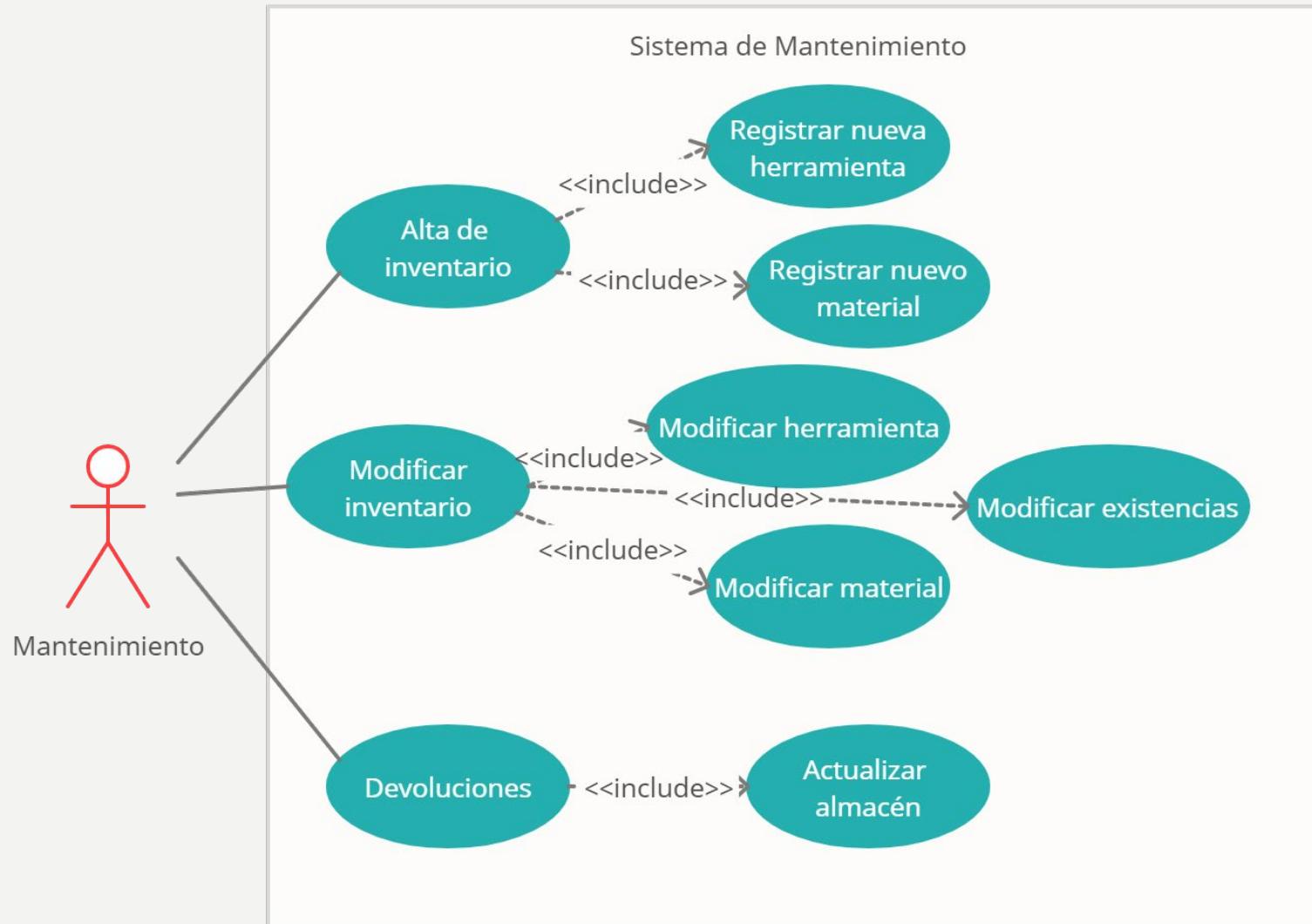


Diagrama de casos de uso.

Jefe de cuadrilla.

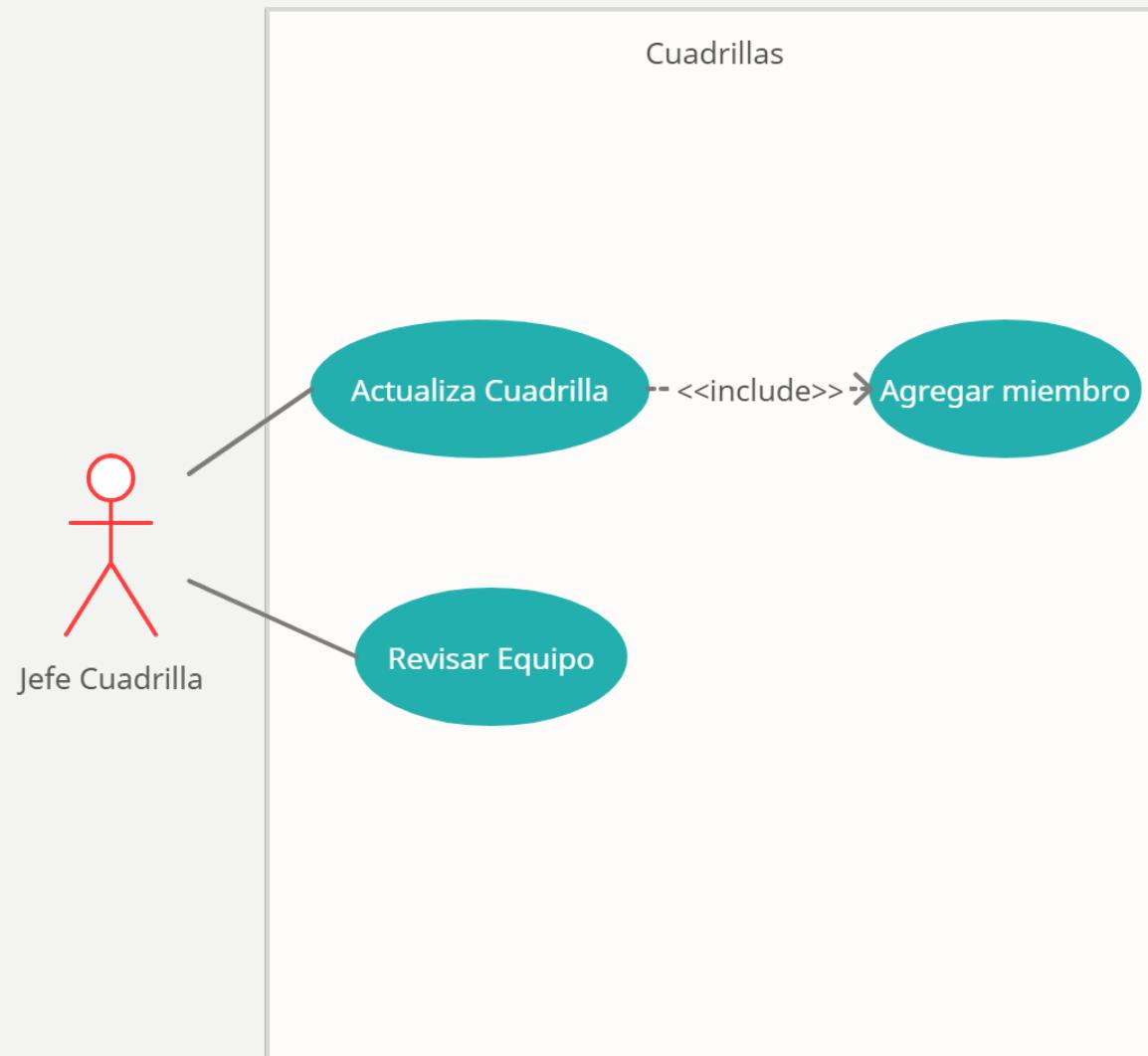


Diagrama de actividades.

Diagrama 1. Inicio de sesión.

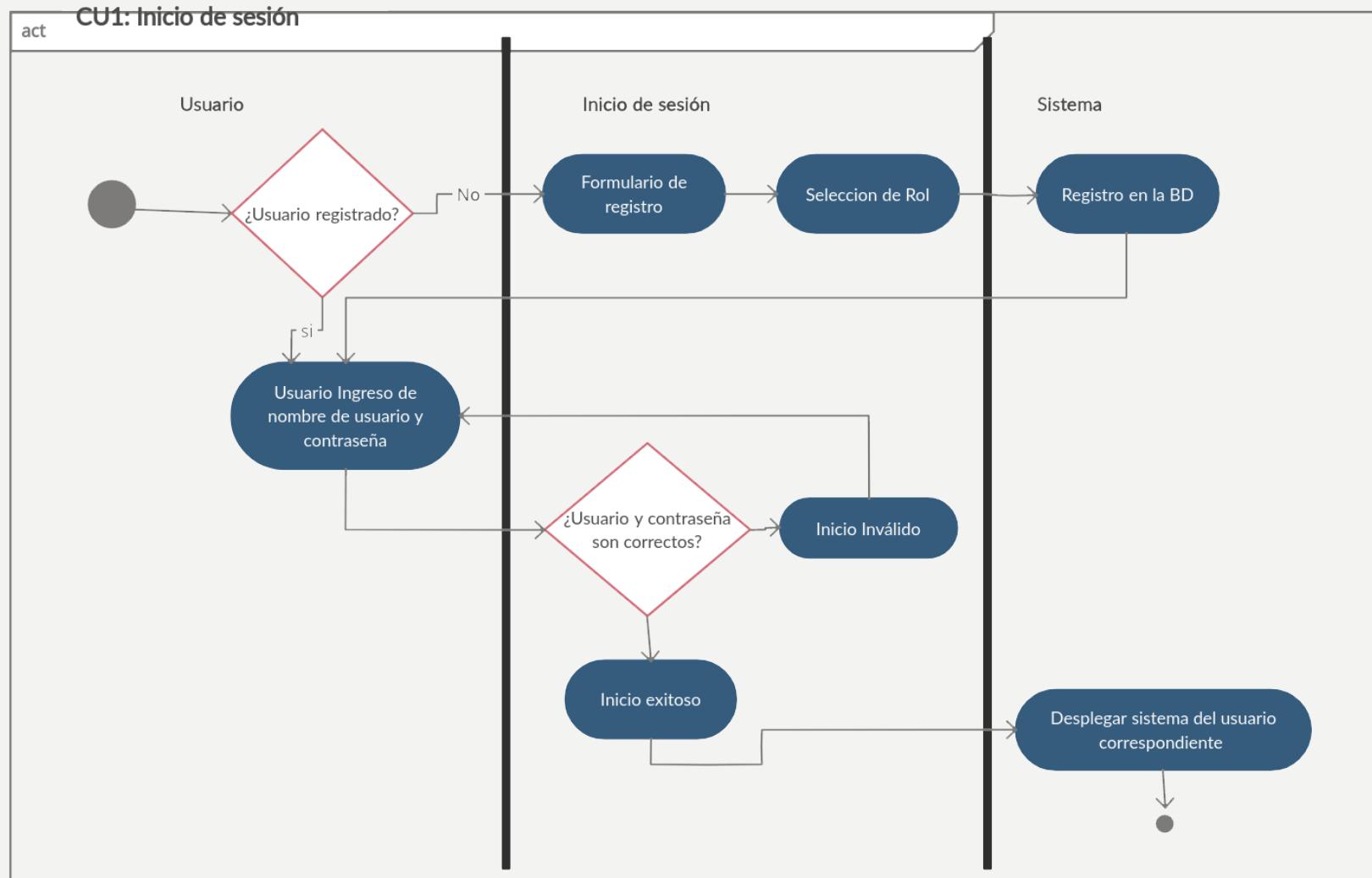


Diagrama de actividades.

Diagrama 2. Seleccionar operación.

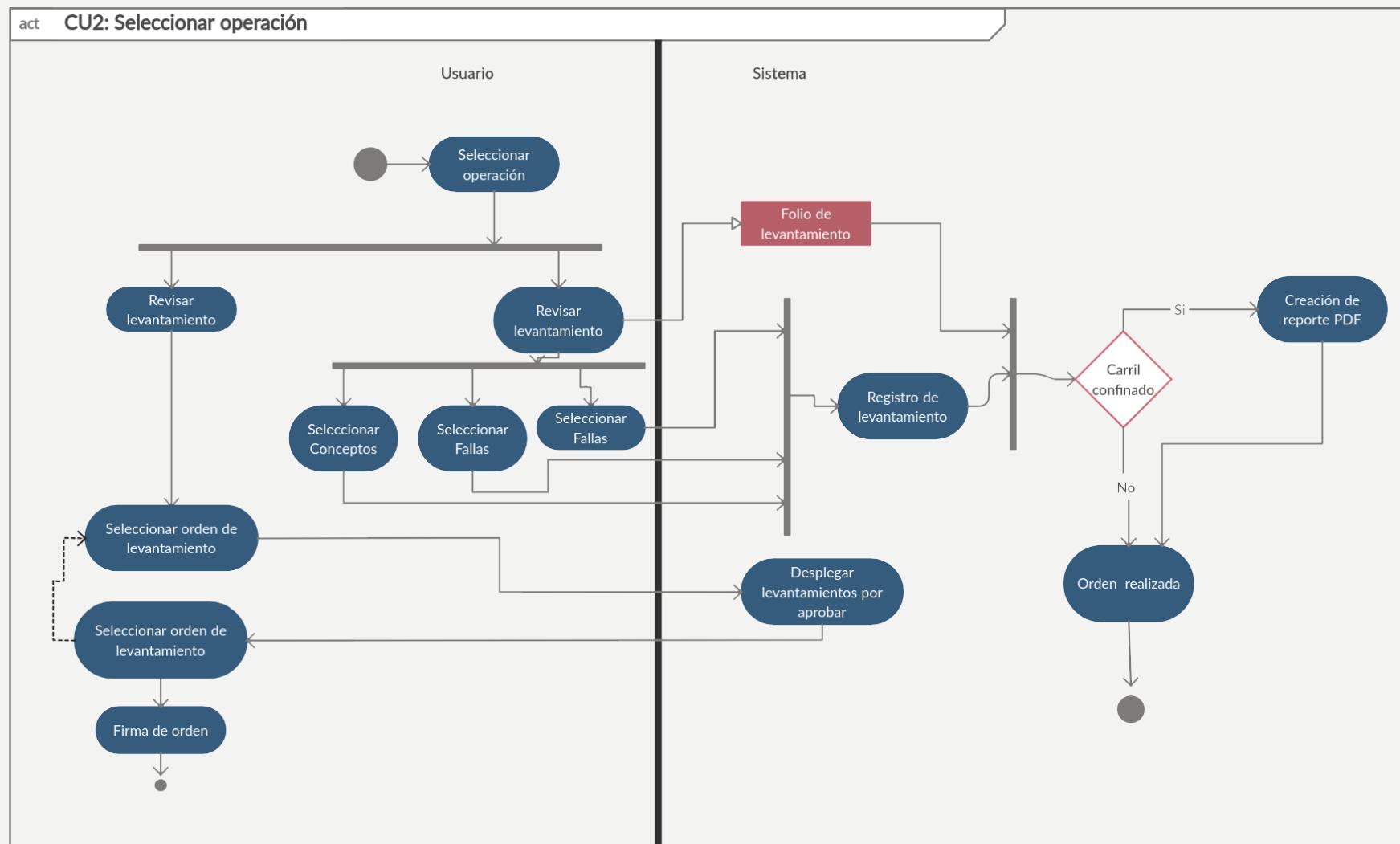


Diagrama de actividades.

Diagrama 3. Consultar levantamiento.

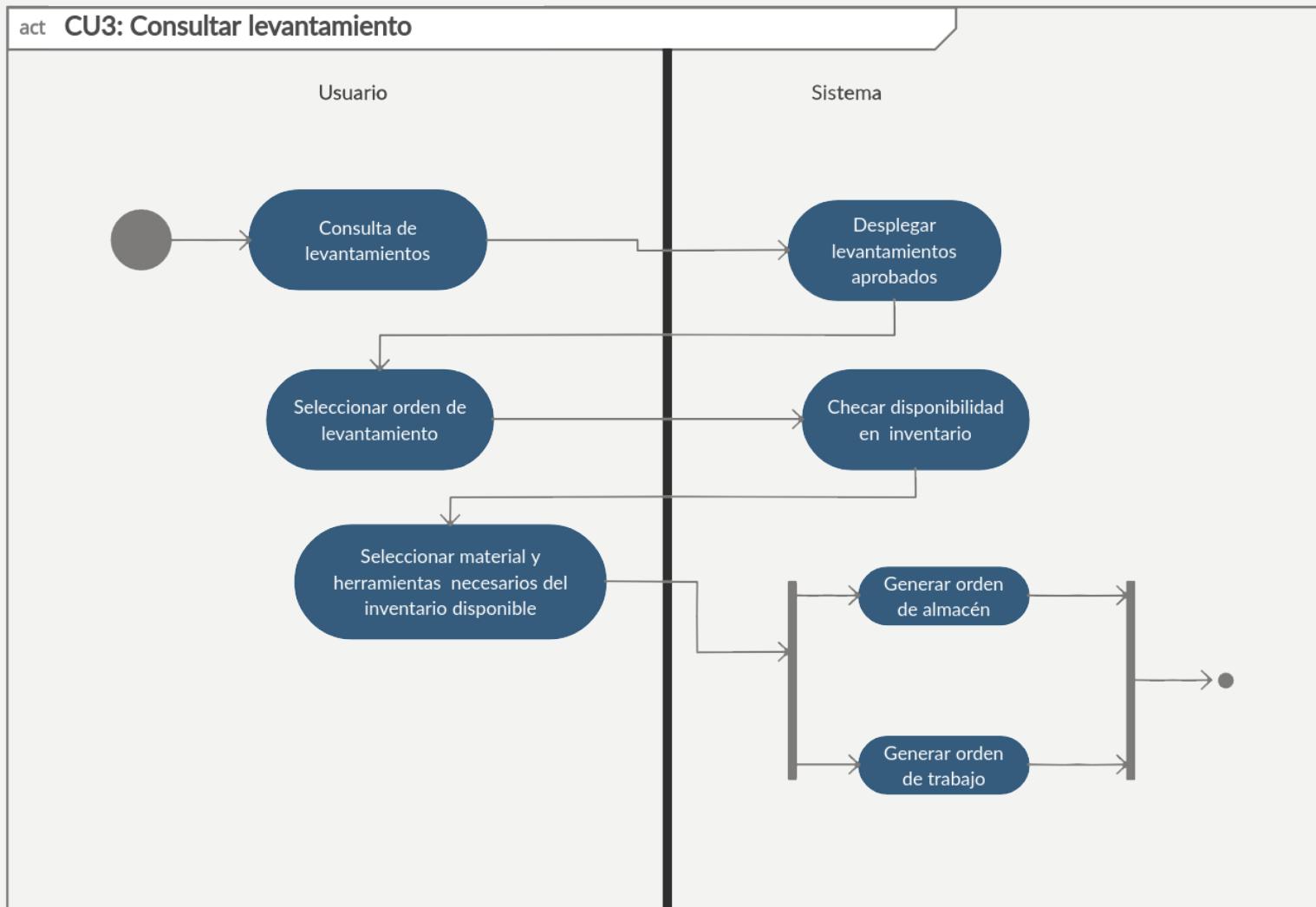


Diagrama de actividades.

Diagrama 4. Cerrar sesión.

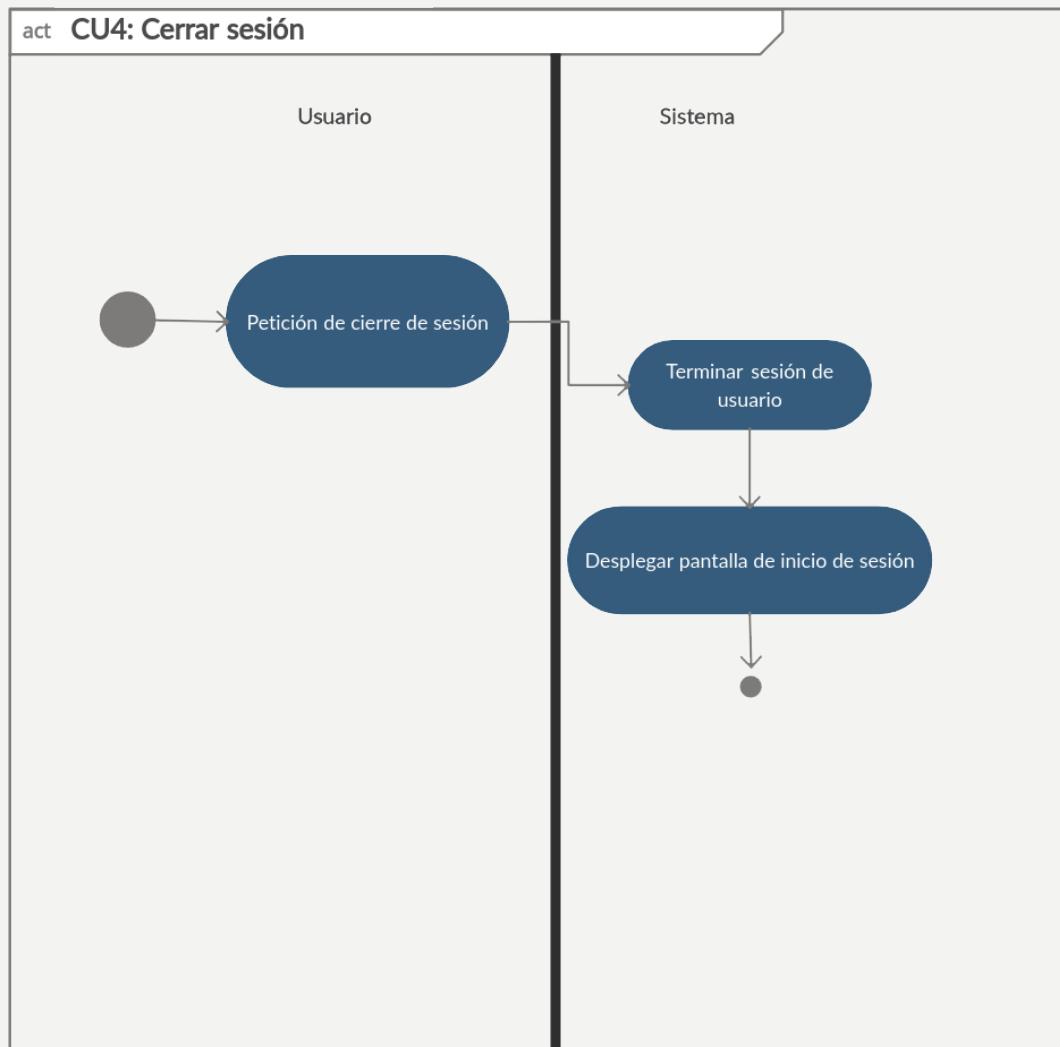


Diagrama de actividades.

Diagrama 5. Alta de línea.

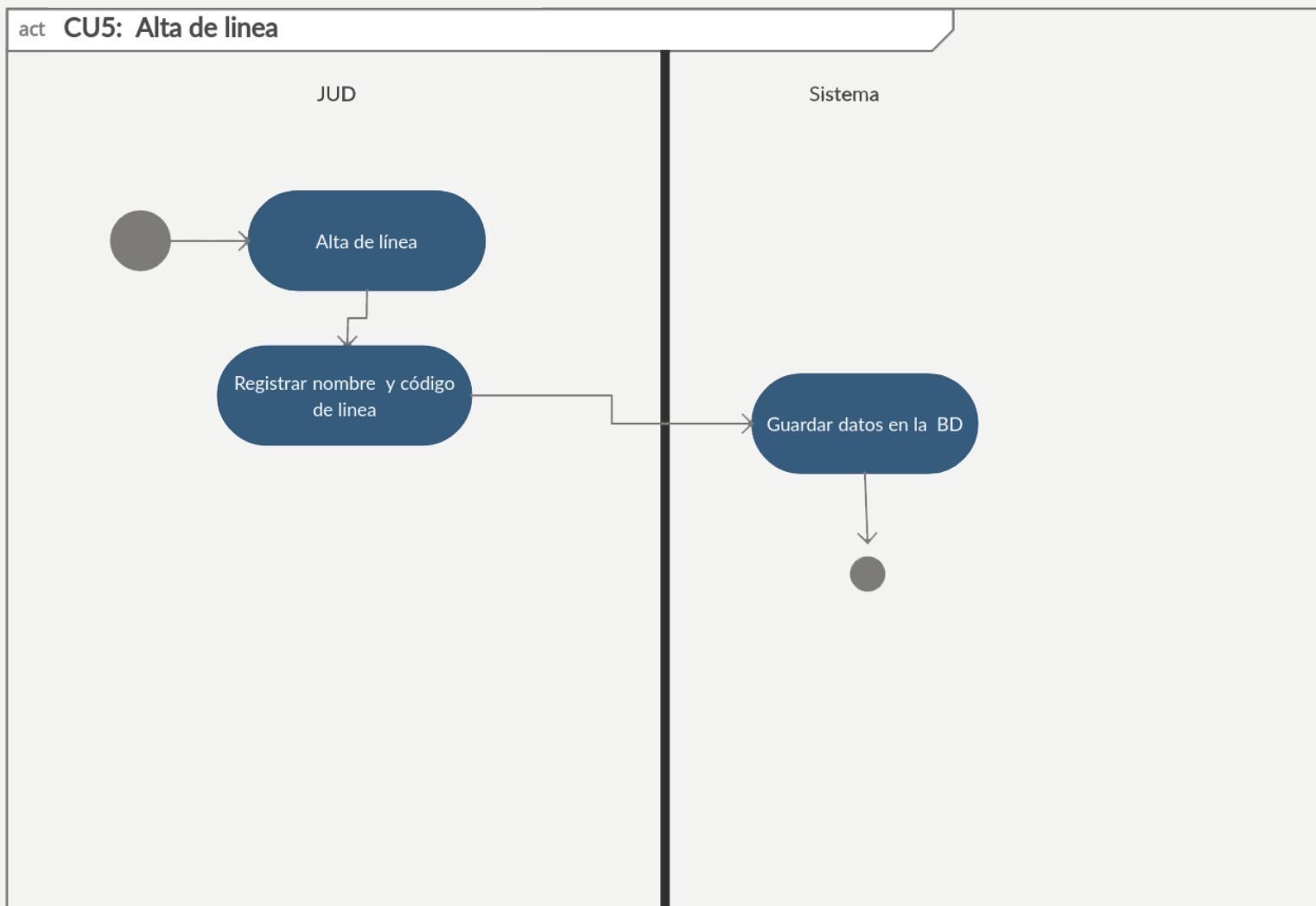


Diagrama de actividades.

Diagrama 6. Alta de estación.

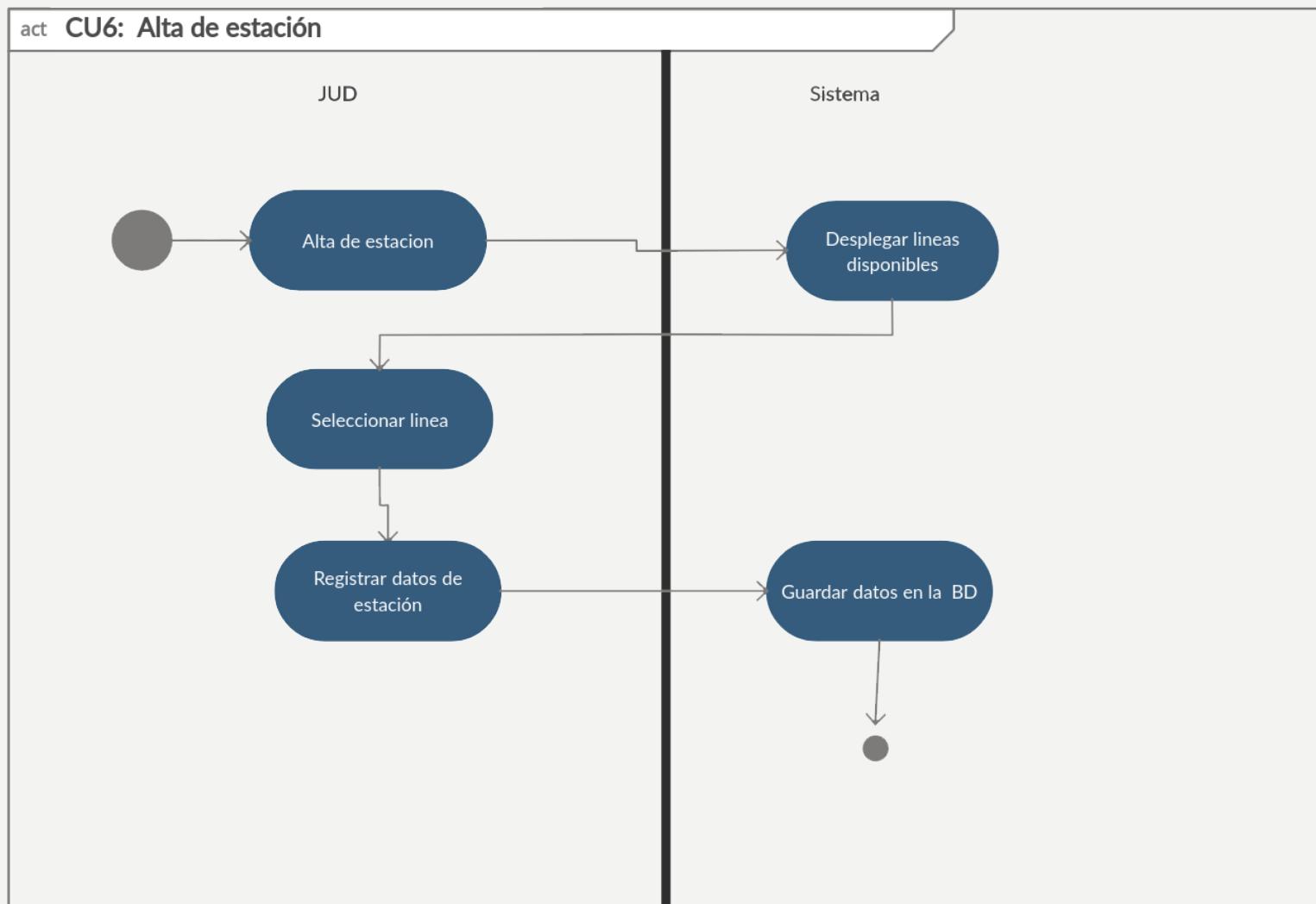


Diagrama de actividades.

Diagrama 7. Alta de categoría.

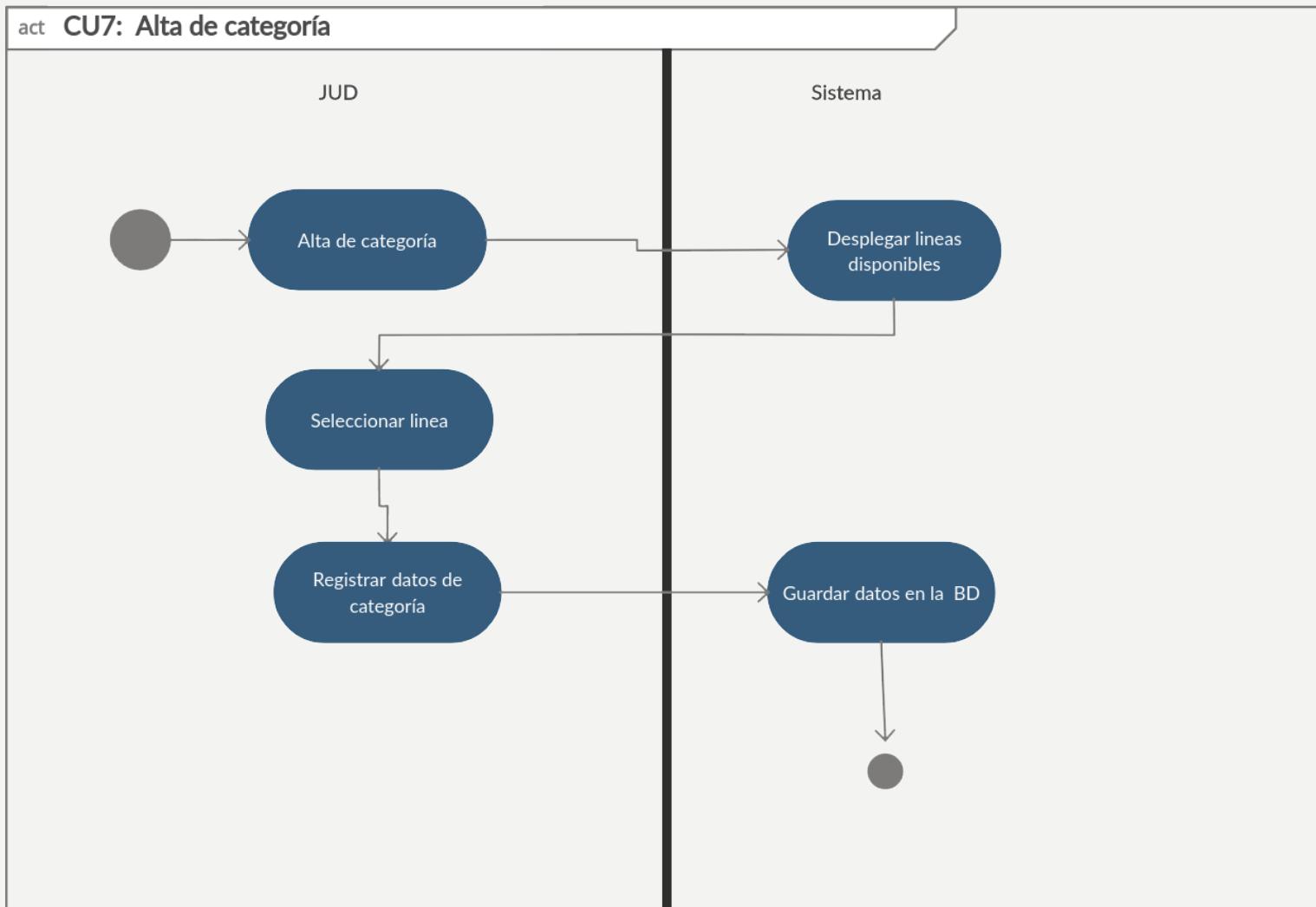


Diagrama de actividades.

Diagrama 8. Alta de concepto.

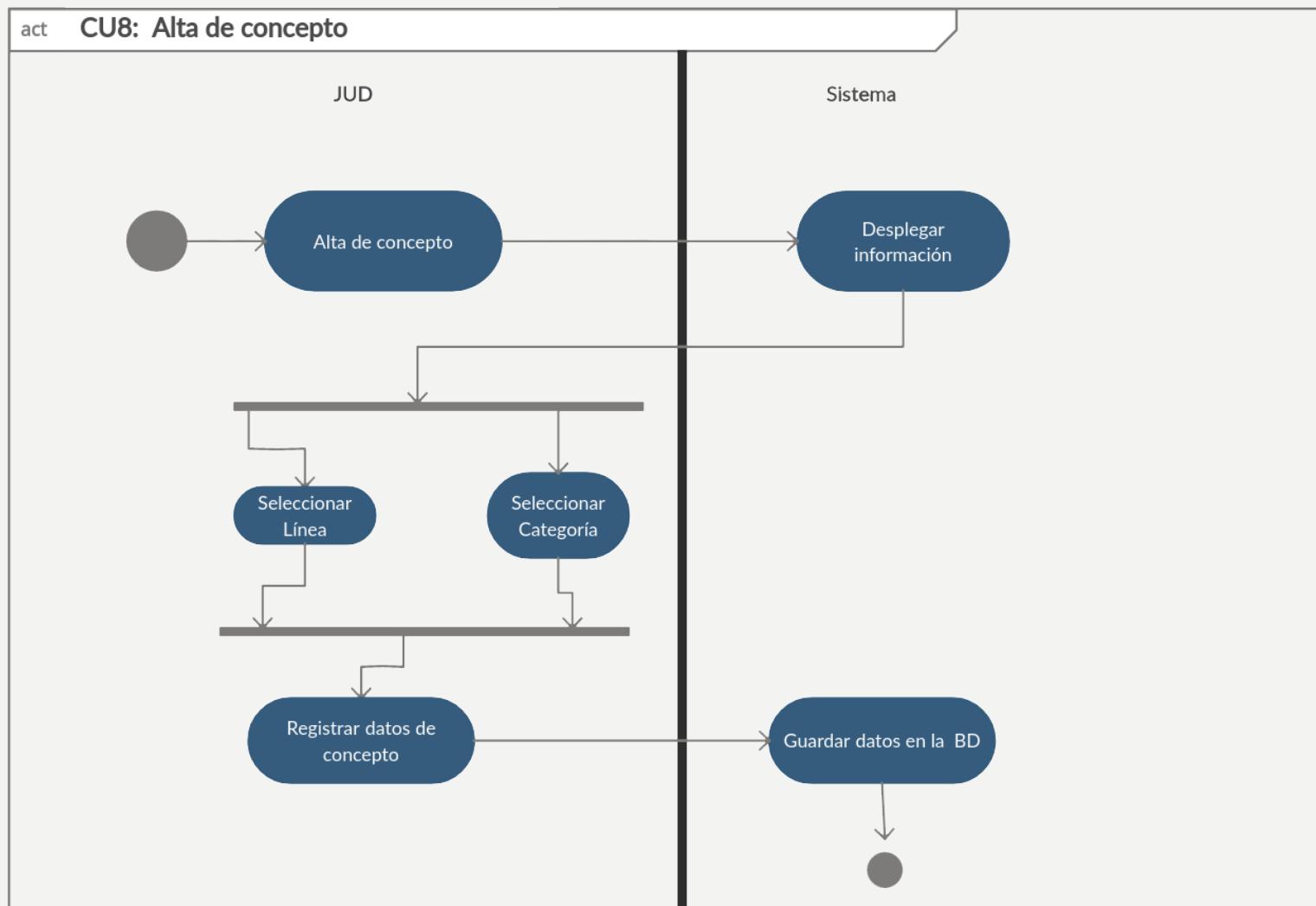


Diagrama de actividades.

Diagrama 9. Alta de parte.

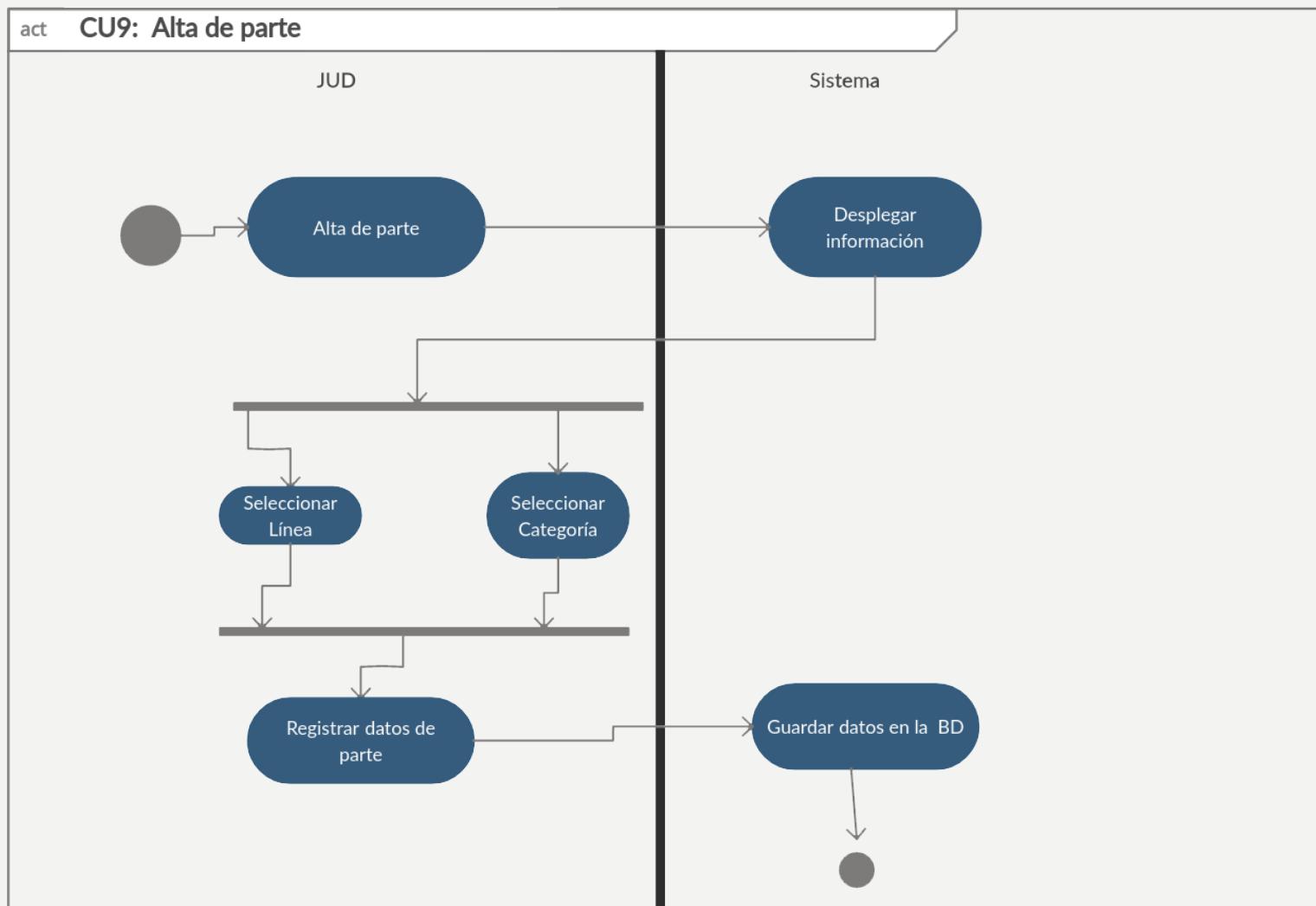


Diagrama de actividades.

Diagrama 10. Alta de falla.

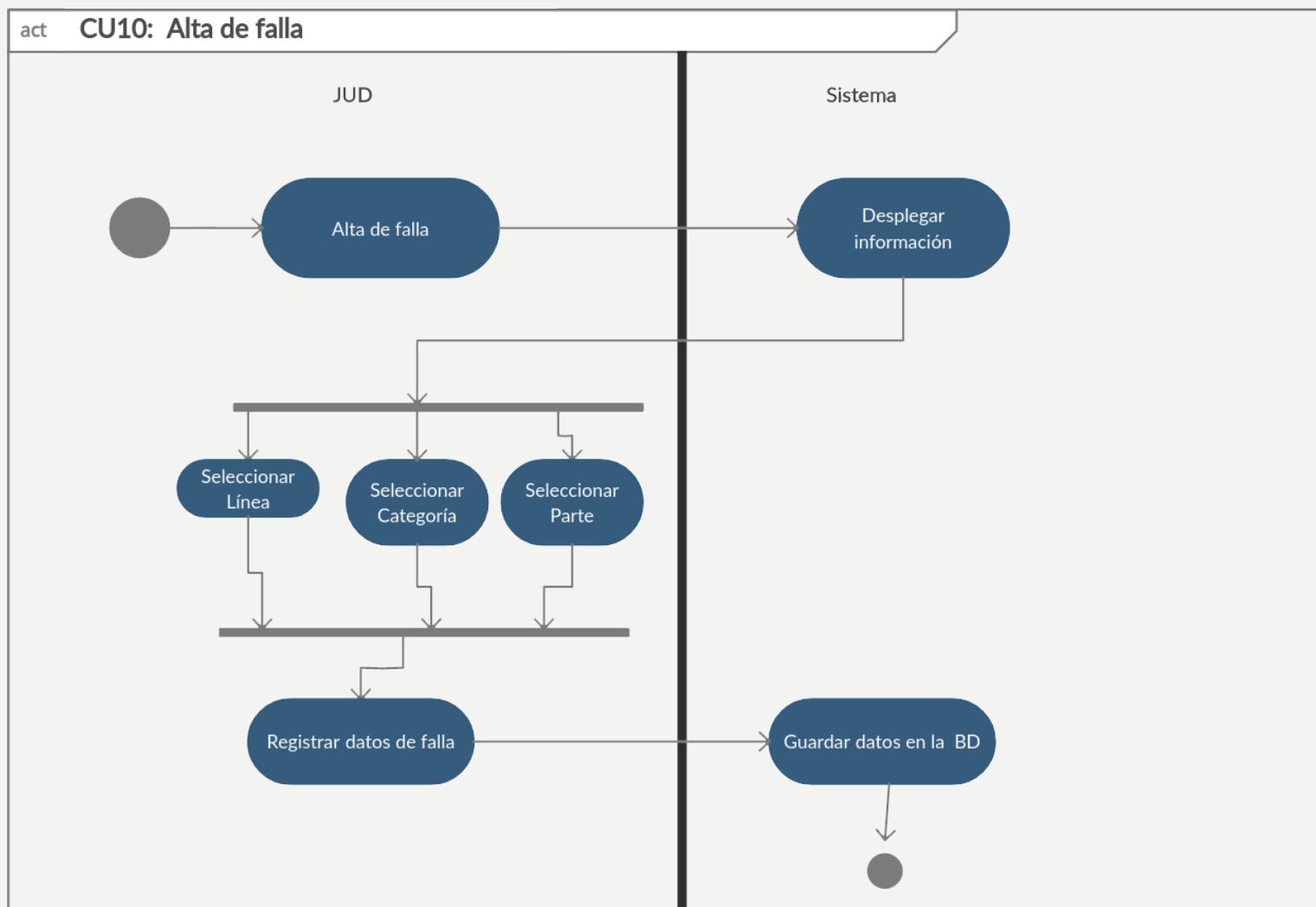


Diagrama de actividades.

Diagrama 11. Modificar estación.

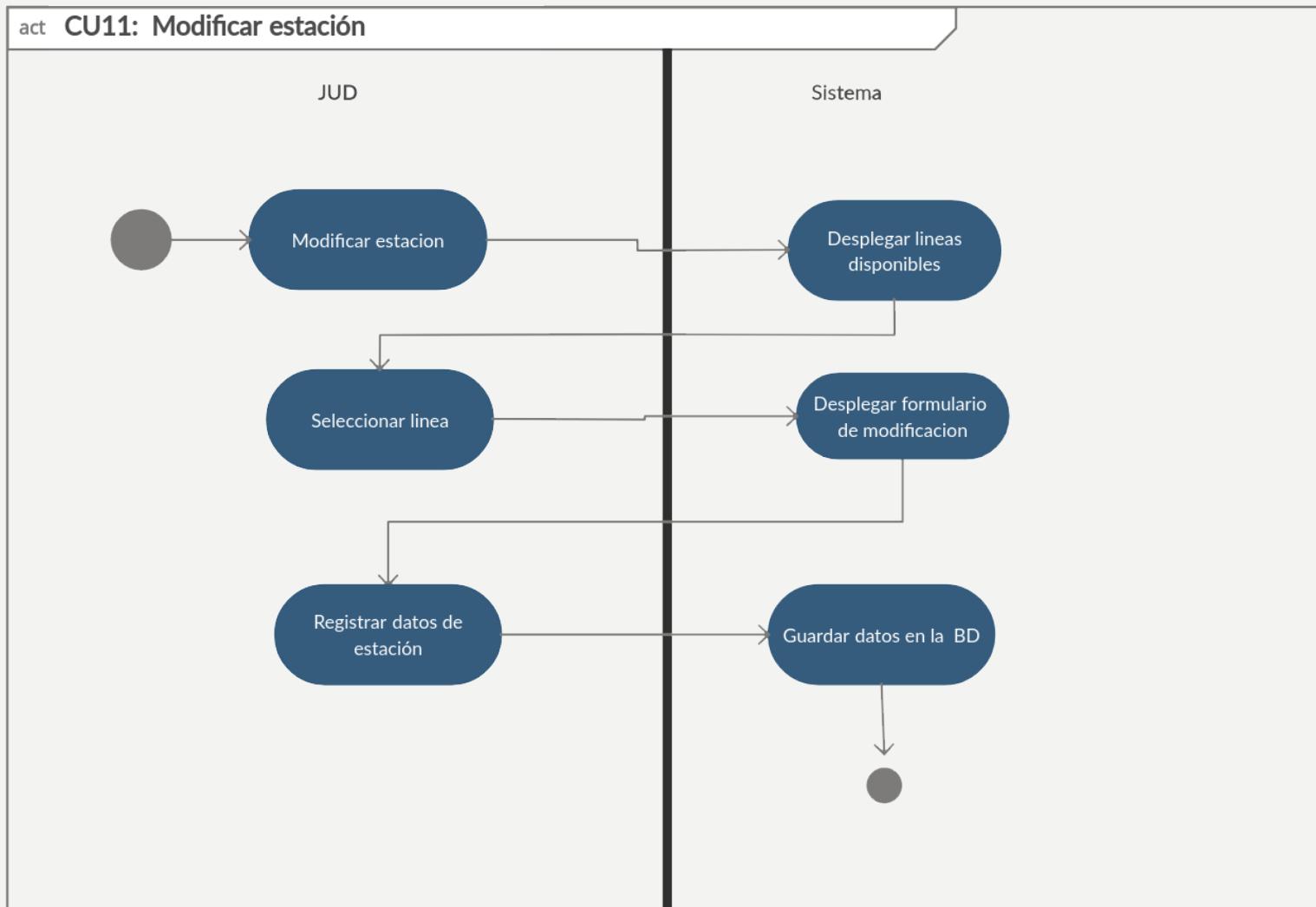


Diagrama de actividades.

Diagrama 12. Modificar concepto.

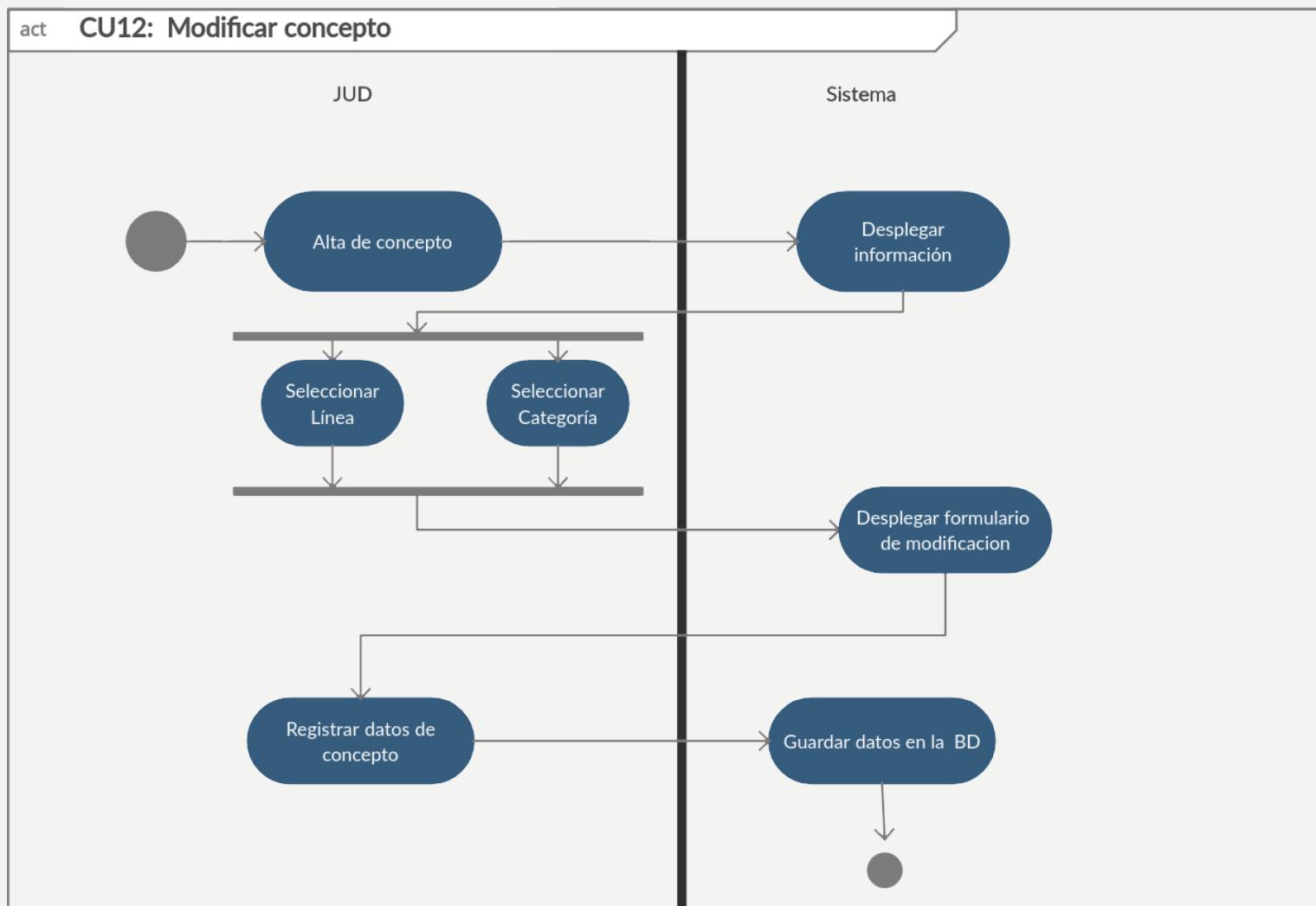


Diagrama de actividades.

Diagrama 13. Modificar falla.

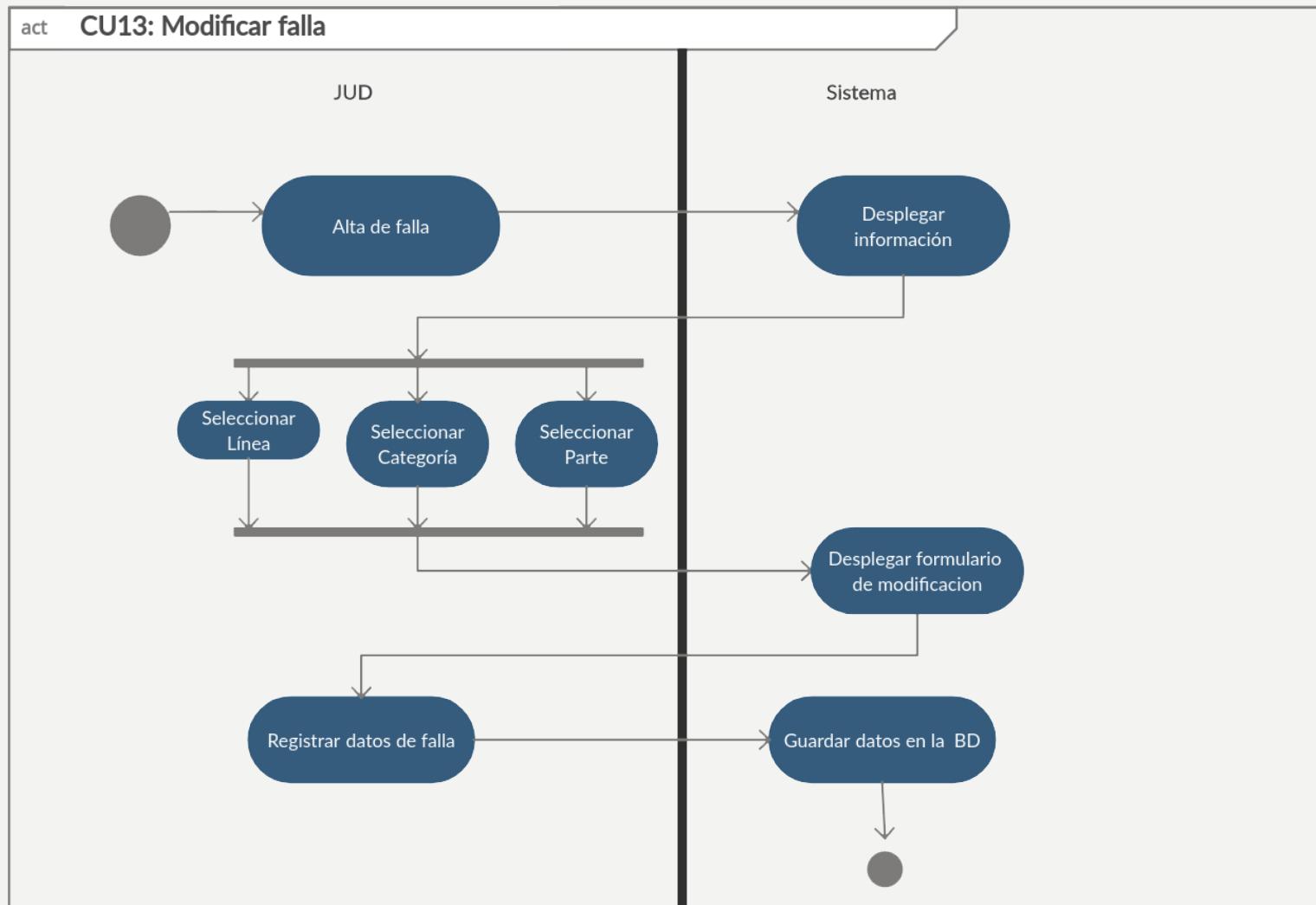


Diagrama de actividades.

Diagrama 14. Activar usuario.

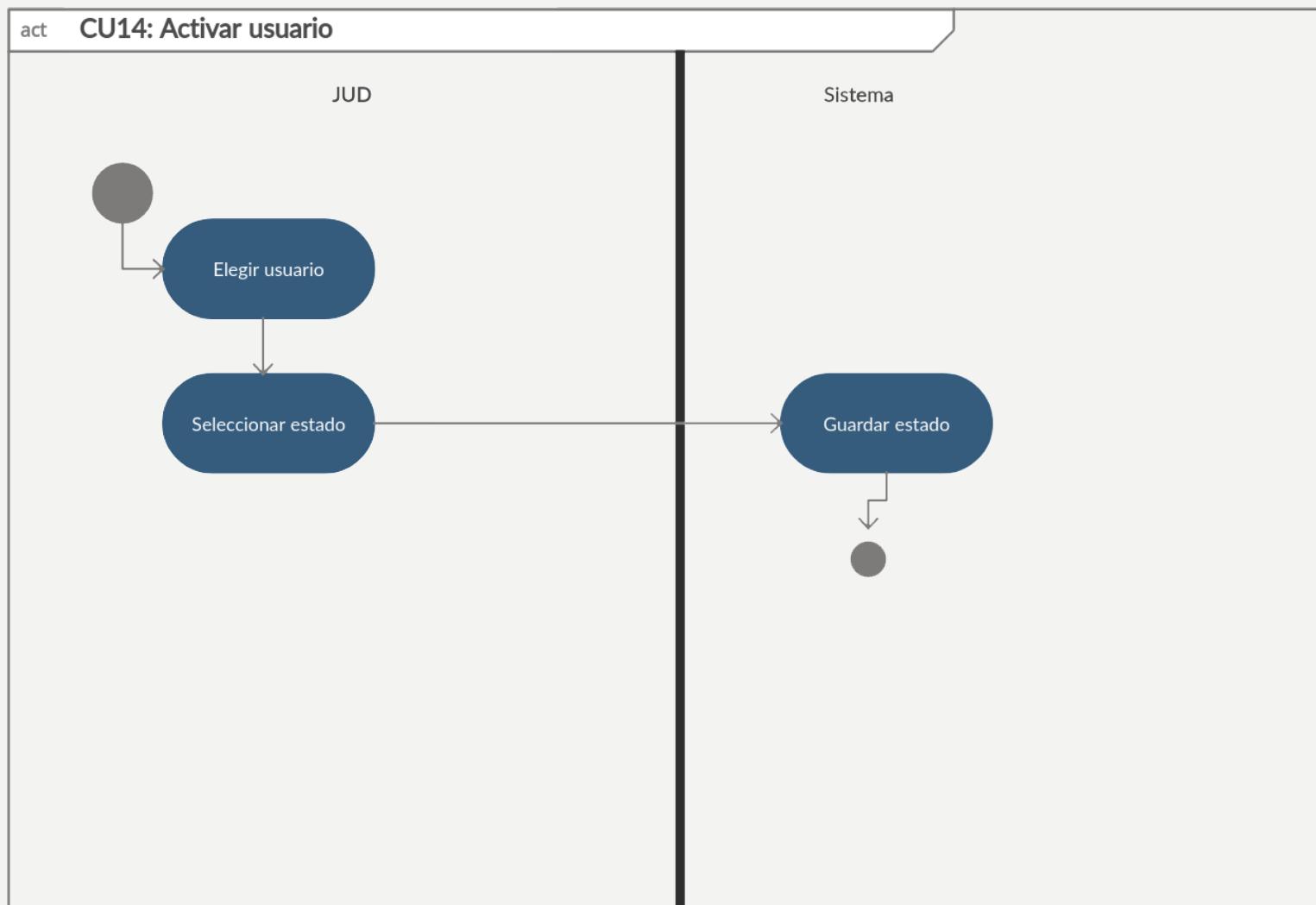


Diagrama de actividades.

Diagrama 15. Firmar levantamiento y/o solicitud.

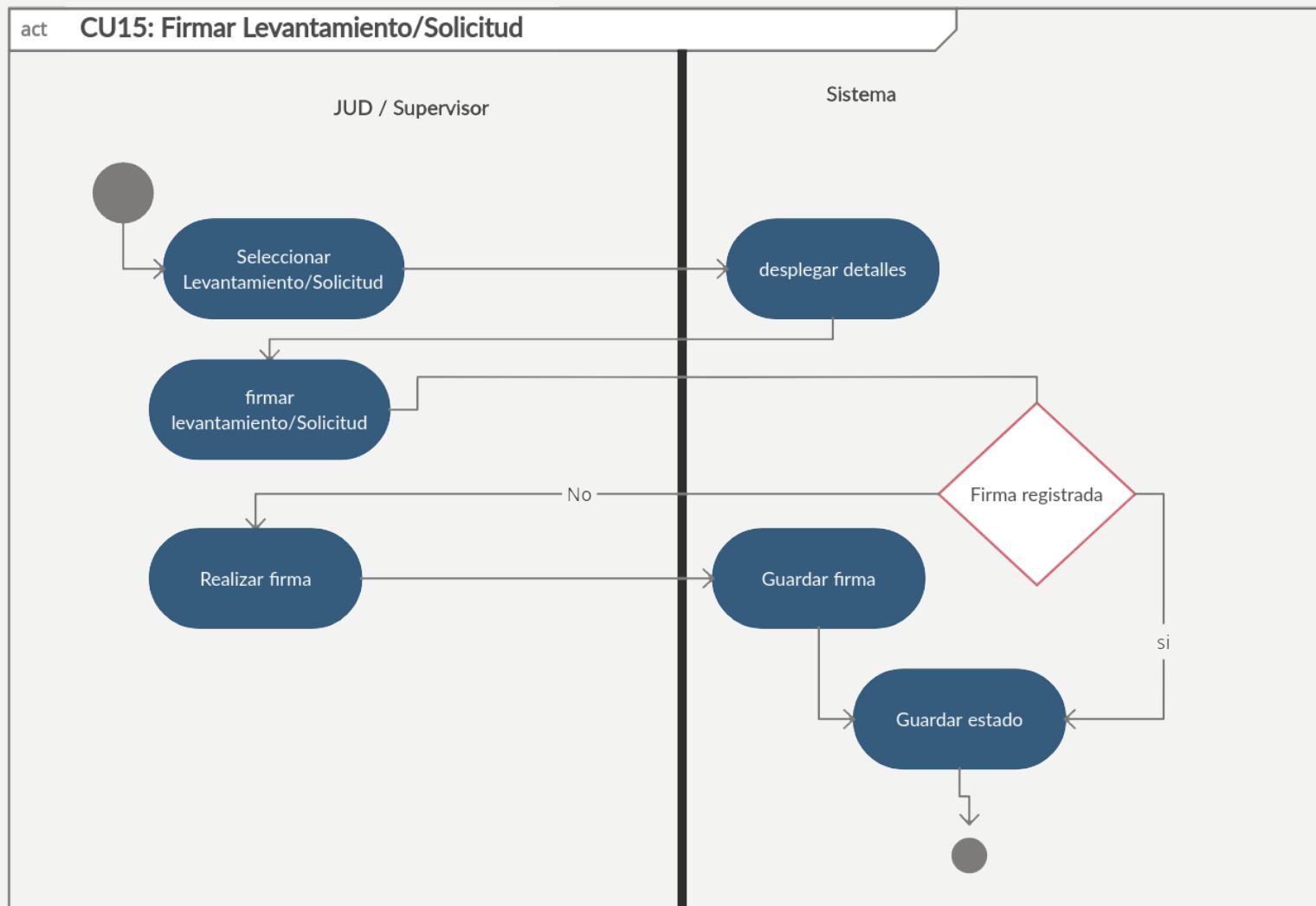


Diagrama de actividades.

Diagrama 16. Alta de inventario.

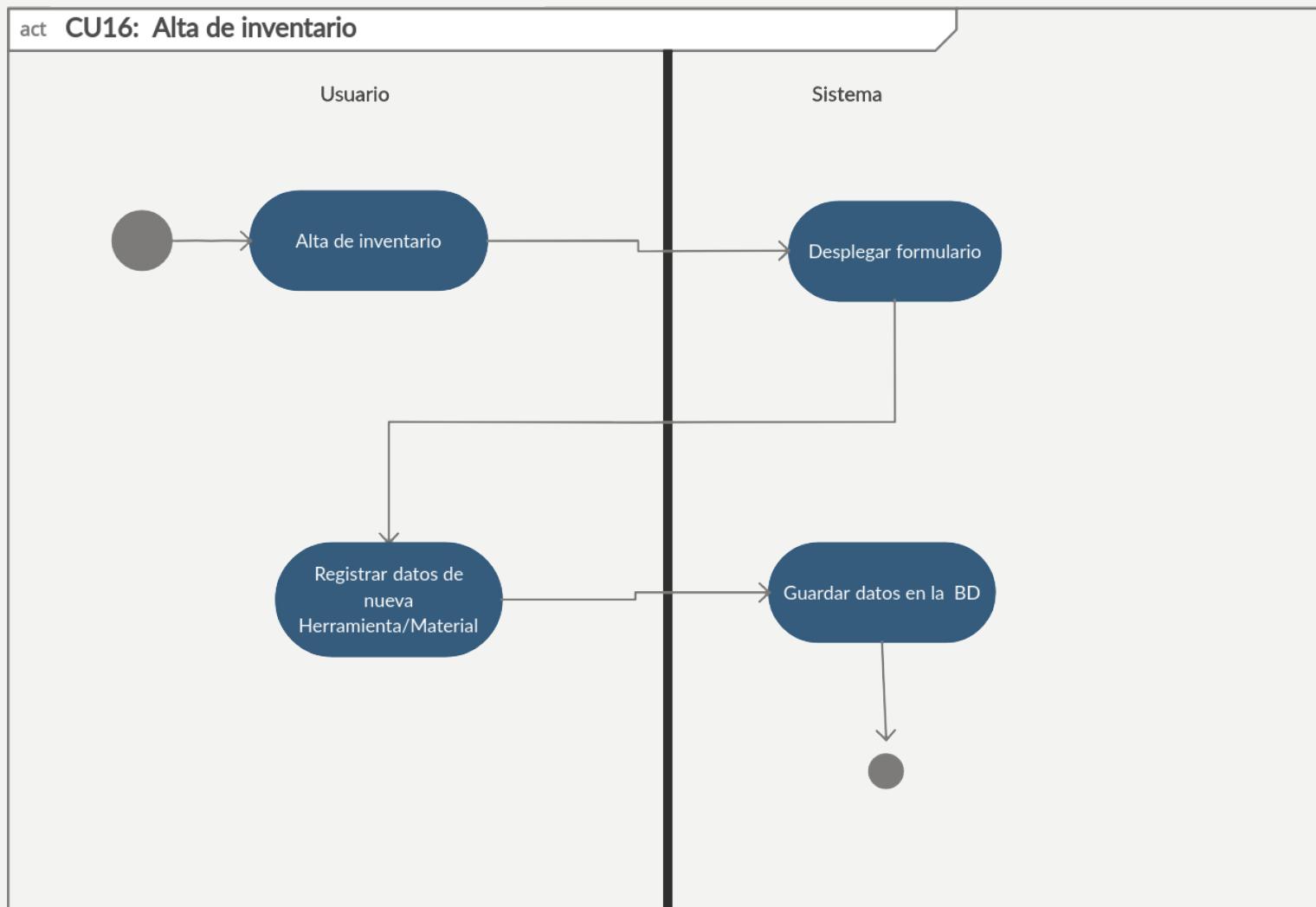


Diagrama de actividades.

Diagrama 17. Modificación de inventario.

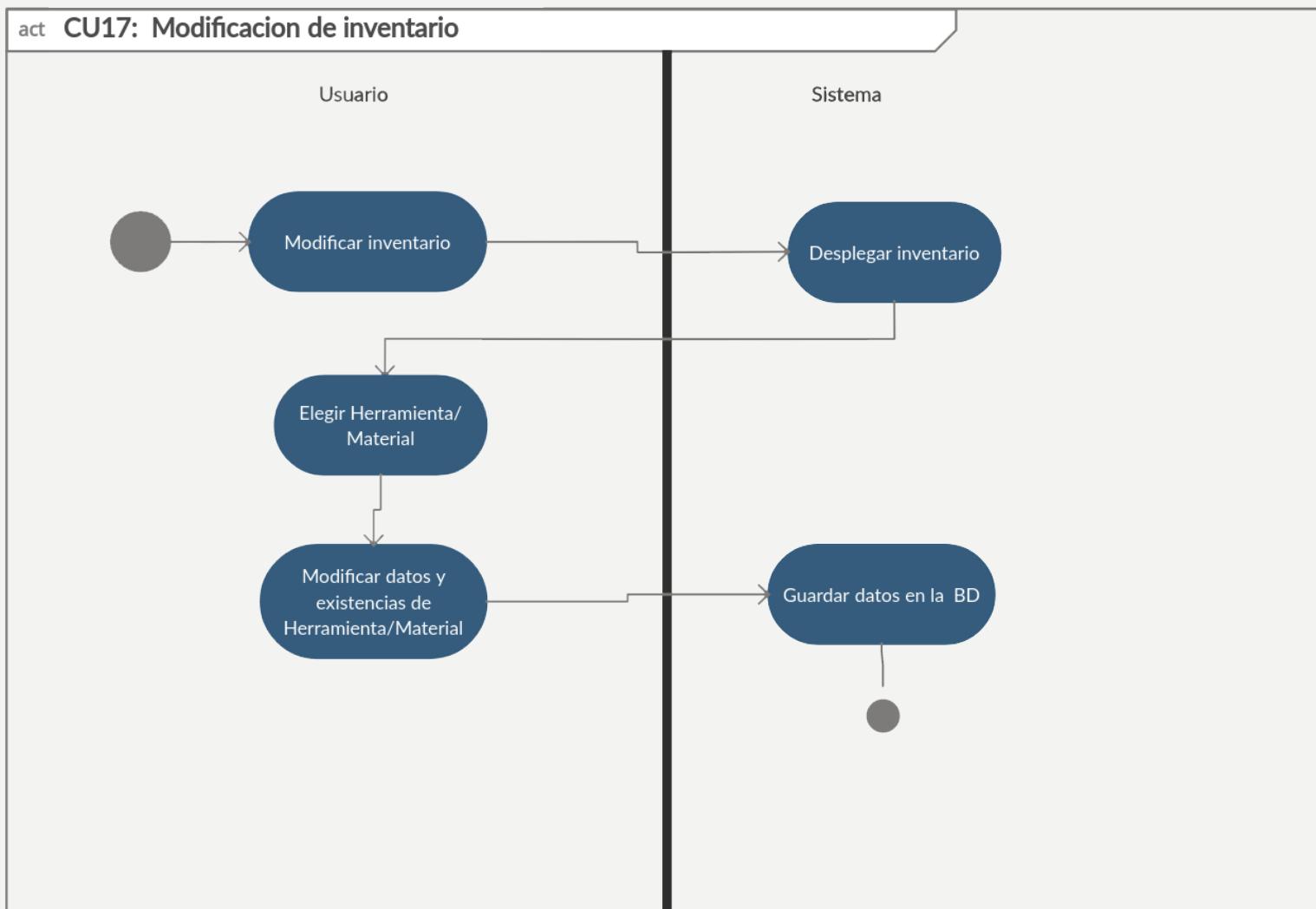


Diagrama de actividades.

Diagrama 18. Devoluciones de inventario.

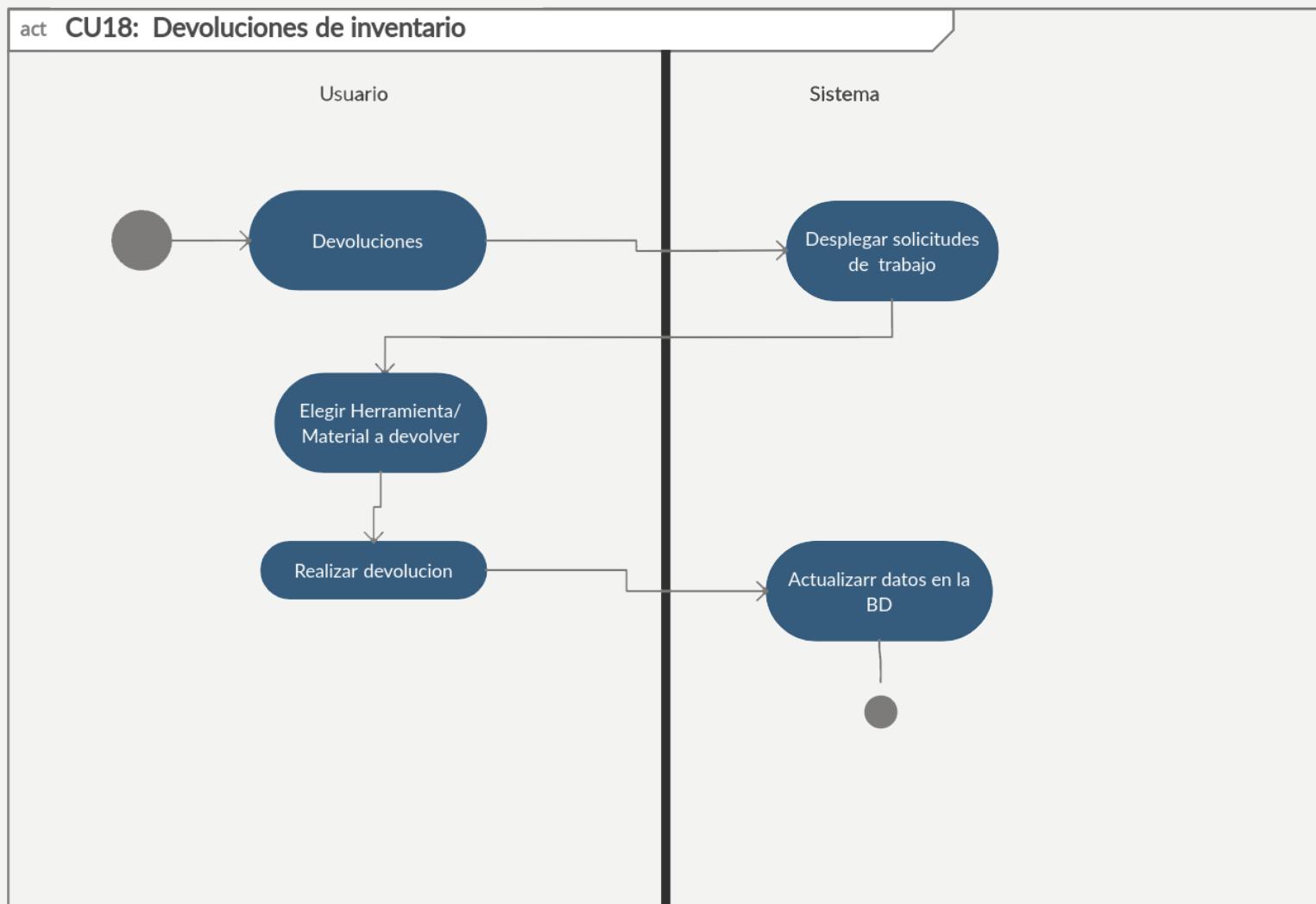


Diagrama de actividades.

Diagrama 19. Solicitud de trabajo y/o almacén.

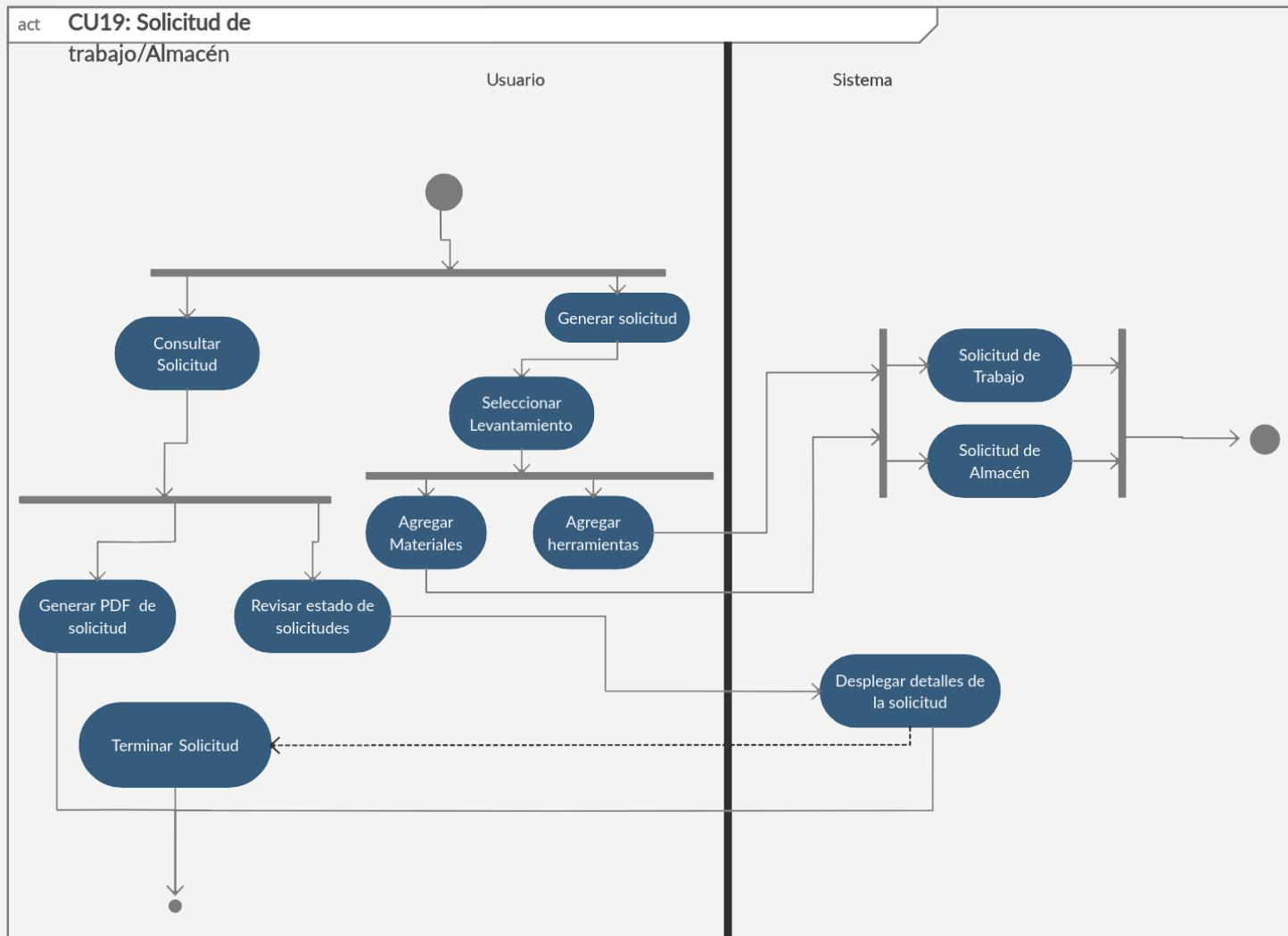


Diagrama de actividades.

Diagrama 20. Cuadrillas.

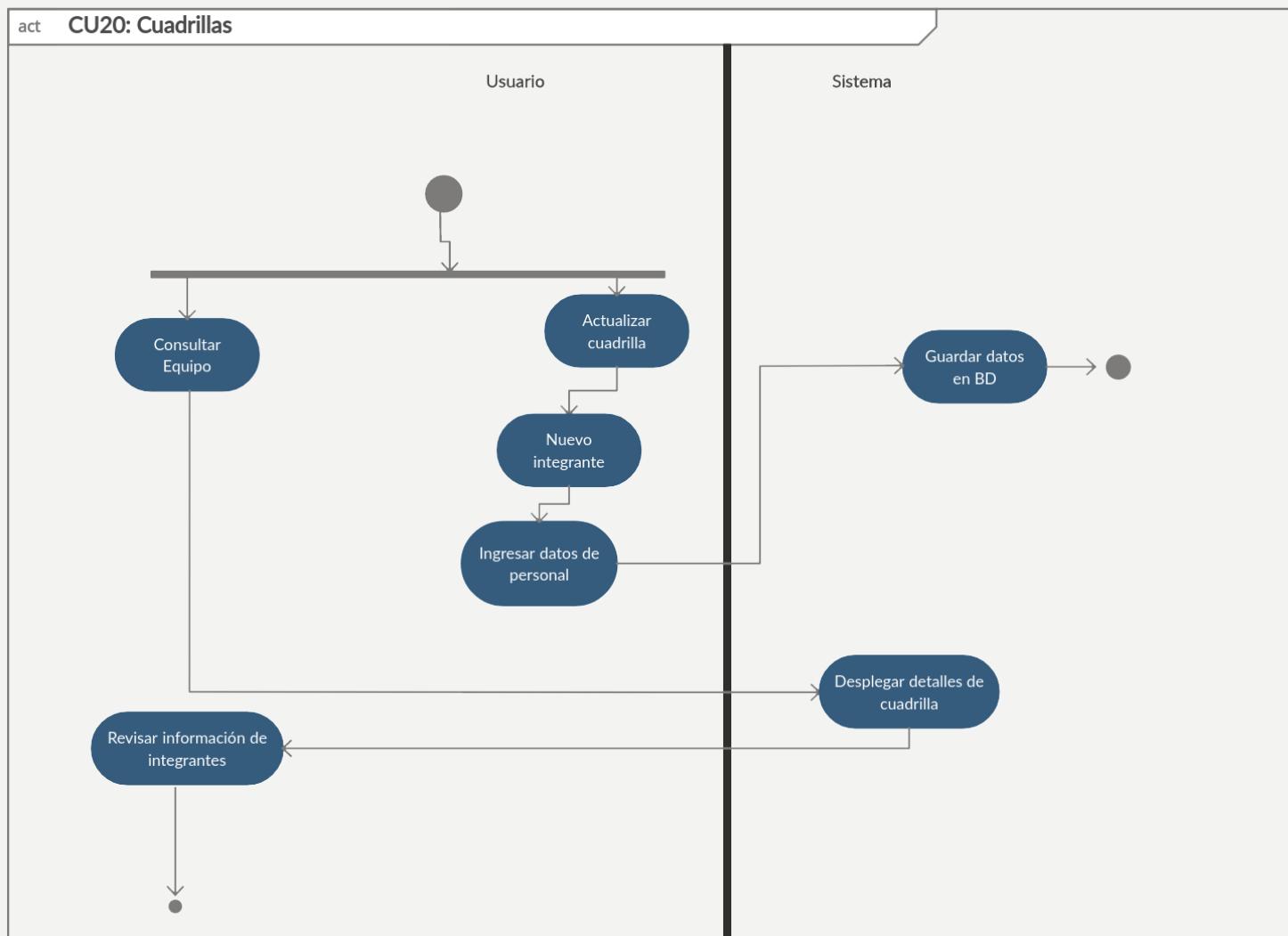


Diagrama de actividades.

Diagrama 21. Baja de línea y/o estación.

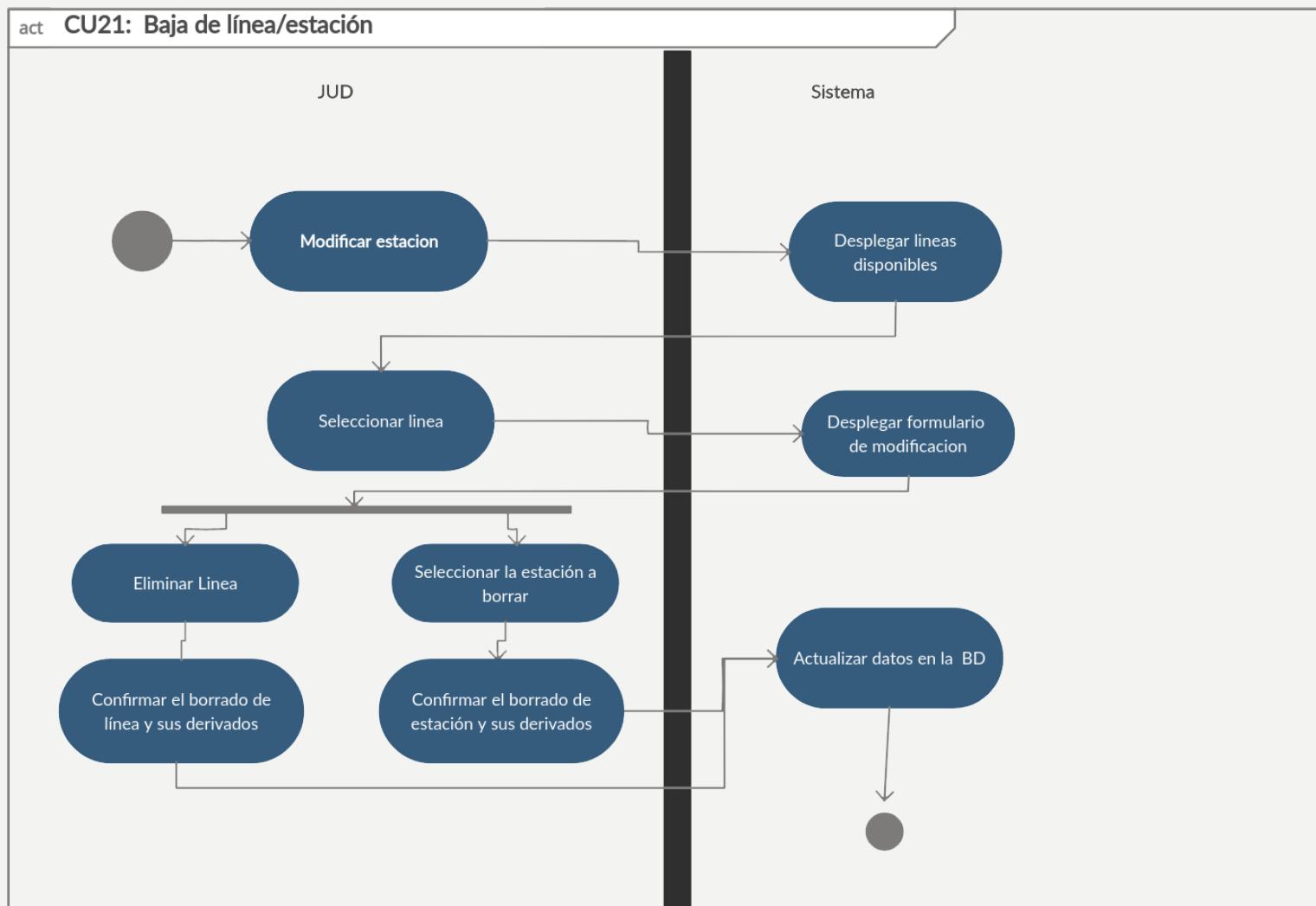


Diagrama de actividades.

Diagrama 22. Baja de categoría y/o concepto.

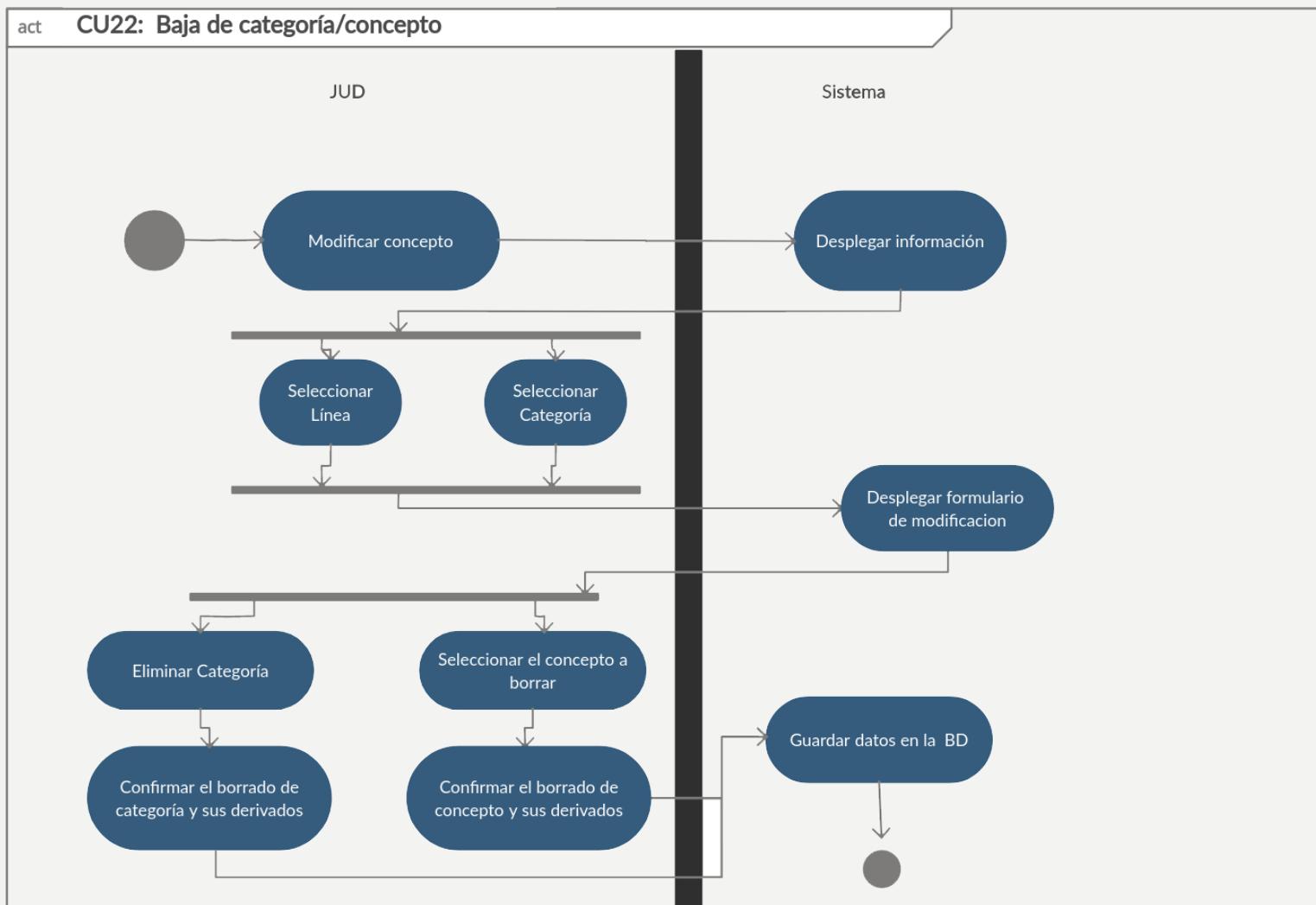
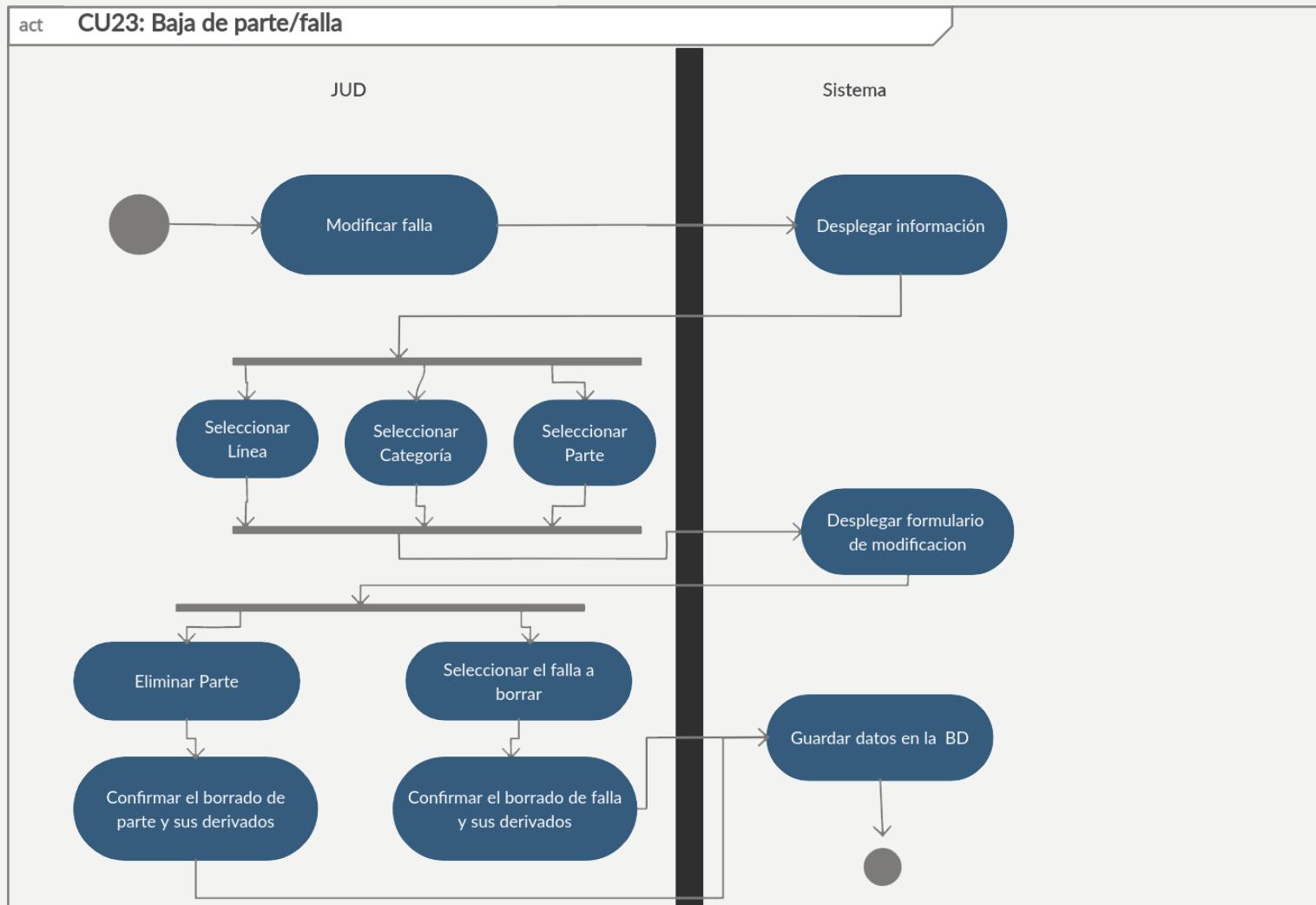


Diagrama de actividades.

Diagrama 23. Baja de parte y/o falla.



Funcionalidad

Unidades de Transporte.

Texto y más texto.

más texto...

más texto...

Estaciones y Mantenimiento.

Texto y más texto.

más texto...

Servicios ofrecidos.

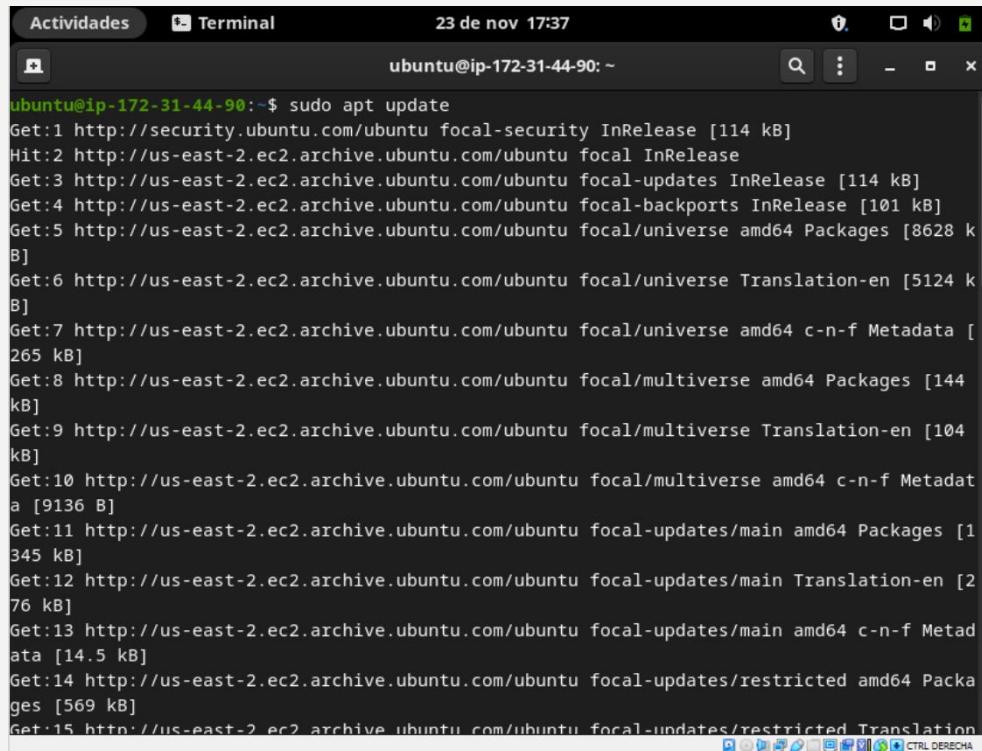
Texto y más texto.

Despliegue y configuración de los componentes

Pasos previos a la instalación.

Con acceso al servior de, en la terminal se ingresan los siguientes comandos:

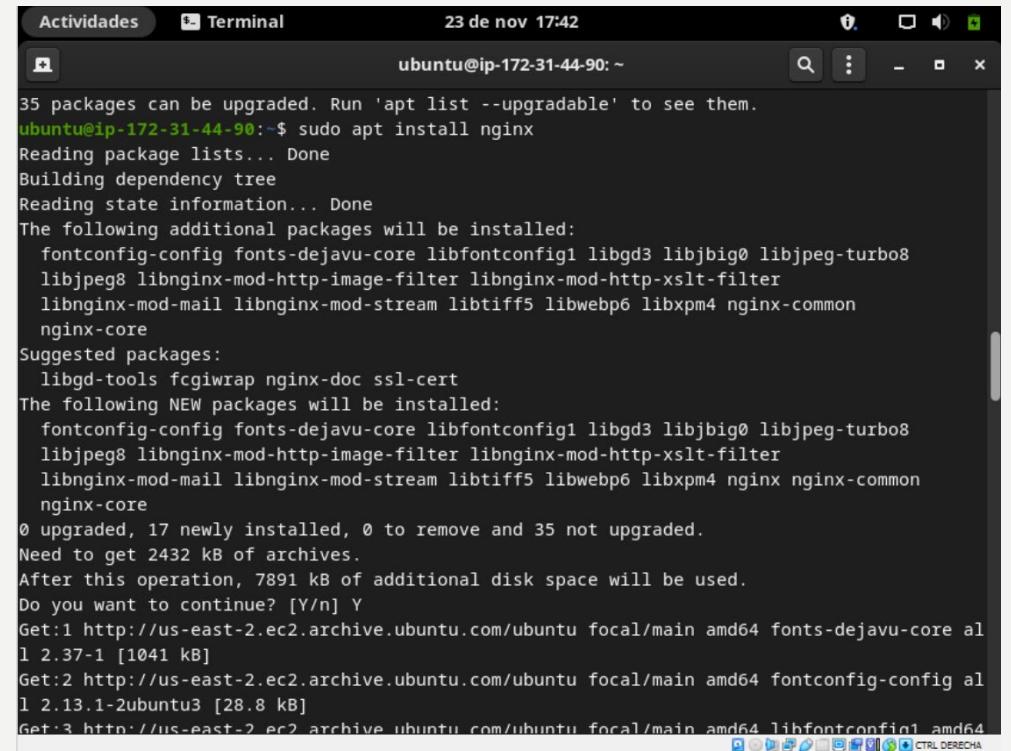
- Primero se actualizan todos los repositorios de Linux.



```
Actividades Terminal 23 de nov 17:37
ubuntu@ip-172-31-44-90:~$ sudo apt update
Get:1 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Hit:2 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal InRelease
Get:3 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:4 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-backports InRelease [101 kB]
Get:5 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal/universe amd64 Packages [8628 kB]
B]
Get:6 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal/universe Translation-en [5124 kB]
B]
Get:7 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal/universe amd64 c-n-f Metadata [265 kB]
Get:8 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal/multiverse amd64 Packages [144 kB]
Get:9 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal/multiverse Translation-en [104 kB]
Get:10 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal/multiverse amd64 c-n-f Metadata [9136 kB]
Get:11 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [1345 kB]
Get:12 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-updates/main Translation-en [276 kB]
Get:13 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-updates/main amd64 c-n-f Metadata [14.5 kB]
Get:14 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-updates/restricted amd64 Packages [569 kB]
Get:15 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-updates/restricted Translation-en [1041 kB]
```

Instalación del servidor.

- Despues, usando “sudo apt install Nginx” para instalar el software que se encarga de actuar como servidor web/proxy.



```
Actividades Terminal 23 de nov 17:42
ubuntu@ip-172-31-44-90:~$ sudo apt install nginx
35 packages can be upgraded. Run 'apt list --upgradable' to see them.
ubuntu@ip-172-31-44-90:~$ sudo apt install nginx
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  fontconfig-config fonts-dejavu-core libfontconfig1 libgd3 libjbig0 libjpeg-turbo8
  libjpeg8 libnginx-mod-http-image-filter libnginx-mod-http-xslt-filter
  libnginx-mod-mail libnginx-mod-stream libtiff5 libwebp6 libxpm4 nginx-common
  nginx-core
Suggested packages:
  libgd-tools fcgiwrap nginx-doc ssl-cert
The following NEW packages will be installed:
  fontconfig-config fonts-dejavu-core libfontconfig1 libgd3 libjbig0 libjpeg-turbo8
  libjpeg8 libnginx-mod-http-image-filter libnginx-mod-http-xslt-filter
  libnginx-mod-mail libnginx-mod-stream libtiff5 libwebp6 libxpm4 nginx nginx-common
  nginx-core
0 upgraded, 17 newly installed, 0 to remove and 35 not upgraded.
Need to get 2432 kB of archives.
After this operation, 7891 kB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal/main amd64 fonts-dejavu-core all 2.37-1 [1041 kB]
Get:2 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal/main amd64 fontconfig-config all 2.13.1-2ubuntu3 [28.8 kB]
Get:3 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal/main amd64 libfontconfig1 amd64
```

- A continuación, es necesario habilitar un firewall denominado “uncomplicated”, que simplifica el proceso de la configuración de este. Se utiliza “**sudo ufw enable**”.
 - Se puede comprobar que el servidor Nginx esta corriendo de forma correcta utilizando el comando “**systemctl status nginx**”.

```
Actividades Terminal 23 de nov 17:44
ubuntu@ip-172-31-44-90: ~
ubuntu@ip-172-31-44-90:~$ sudo ufw enable
Command may disrupt existing ssh connections. Proceed with operation (y|n)? y
Firewall is active and enabled on system startup
```

- Haciendo uso del firewall que se inicio, se activa el protocolo “OpenSSH” necesario para el correcto funcionamiento de Nginx. Se utiliza “`sudo ufw allow 'OpenSSH'`”.

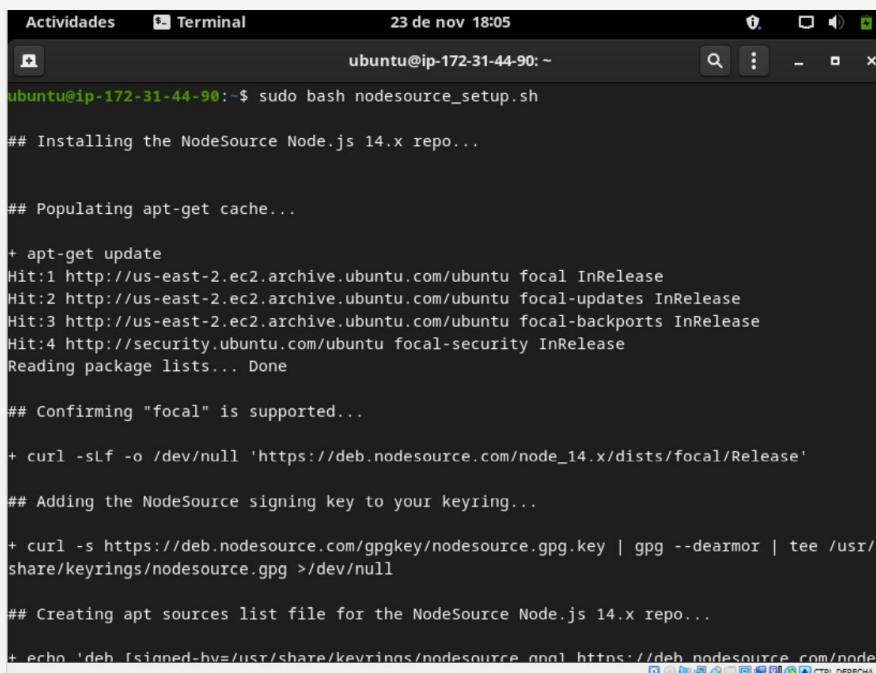
```
Actividades Terminal 23 de nov 17:46
ubuntu@ip-172-31-44-90:~$ sudo ufw allow 'OpenSSH'
Rule added
Rule added (v6)
```

```
Actividades Terminal 23 de nov 17:49
ubuntu@ip-172-31-44-90:~$ systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
  Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset: enabled)
  Active: active (running) since Tue 2021-11-23 23:42:06 UTC; 7min ago
    Docs: man:nginx(8)
 Main PID: 1998 (nginx)
    Tasks: 2 (limit: 1154)
   Memory: 4.9M
      CPU: 0.000 CPU(s) since start
     CGroup: /system.slice/nginx.service
             └─1998 nginx: master process /usr/sbin/nginx -g daemon on; master_process on;
                 ├─1999 nginx: worker process

Nov 23 23:42:06 ip-172-31-44-90 systemd[1]: Starting A high performance web server and a >
Nov 23 23:42:06 ip-172-31-44-90 systemd[1]: Started A high performance web server and a >
lines 1-13/13 (END)
```

Instalación de Node JS.

Se instalará una instancia de Node.js (el motor encargado de ejecutar nuestra aplicación) en la máquina virtual. Primero con el comando “**curl -sL https://deb.nodesource.com/setup_14.x -o nodesource_setup.sh**”. Nos cambiamos al directorio donde se ha descargado el archivo y ejecutamos los comandos bash con la siguiente línea: “**sudo bash nodesource_setup.sh**”.

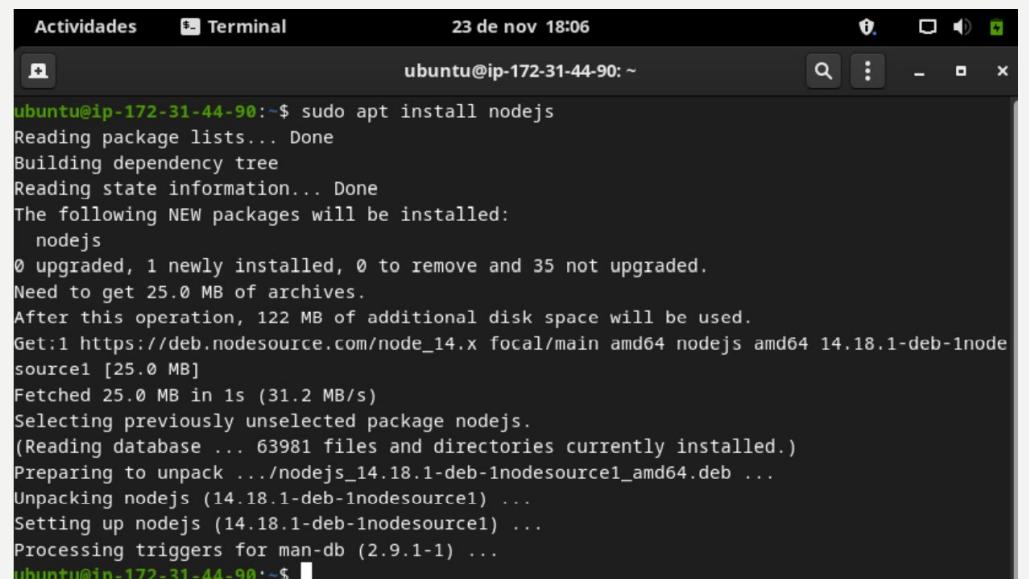


```

Actividades Terminal 23 de nov 18:05
ubuntu@ip-172-31-44-90:~$ sudo bash nodesource_setup.sh
## Installing the NodeSource Node.js 14.x repo...
## Populating apt-get cache...
+ apt-get update
Hit:1 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal InRelease
Hit:2 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:3 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu focal-security InRelease
Reading package lists... Done
## Confirming "focal" is supported...
+ curl -sLF -o /dev/null 'https://deb.nodesource.com/node_14.x/dists/focal/Release'
## Adding the NodeSource signing key to your keyring...
+ curl -s https://deb.nodesource.com/gpgkey/nodesource.gpg.key | gpg --dearmor | tee /usr/share/keyrings/nodesource.gpg >/dev/null
## Creating apt sources list file for the NodeSource Node.js 14.x repo...
+ echo 'deb [signed-by=/usr/share/keyrings/nodesource.gpg] https://deb.nodesource.com/node_14.x focal main' > /etc/apt/sources.list.d/nodesource.list

```

- Una vez terminado, instalaremos los paquetes desde el repositorio que obtuvimos, utilizando el comando: “**sudo apt install nodejs**”.

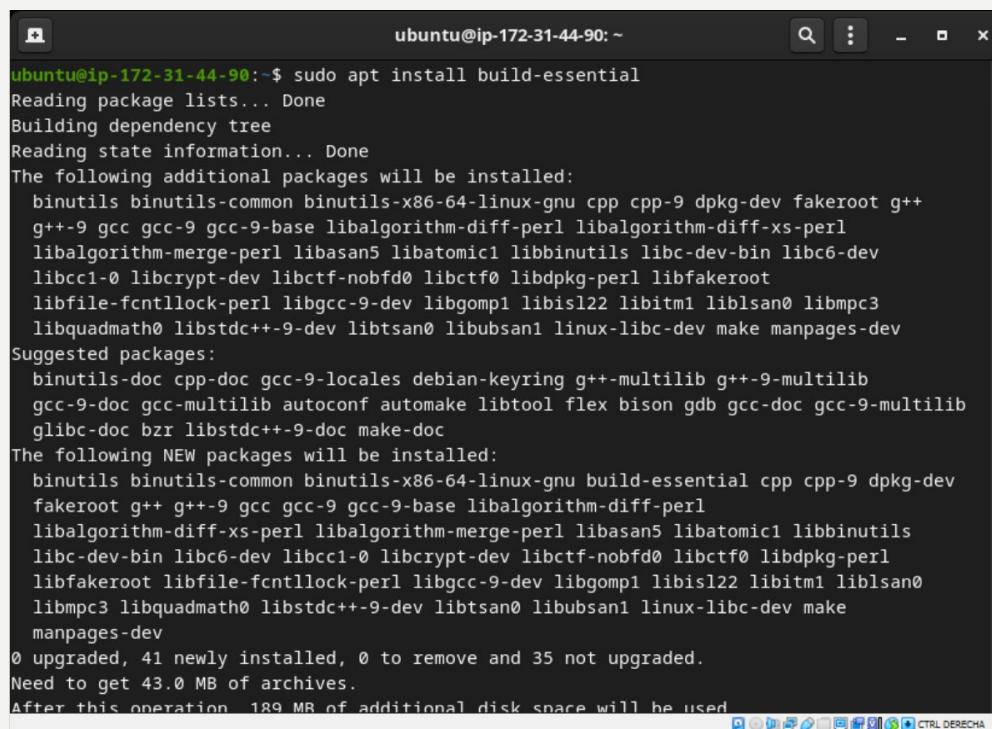


```

Actividades Terminal 23 de nov 18:06
ubuntu@ip-172-31-44-90:~$ sudo apt install nodejs
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  nodejs
0 upgraded, 1 newly installed, 0 to remove and 35 not upgraded.
Need to get 25.0 MB of archives.
After this operation, 122 MB of additional disk space will be used.
Get:1 https://deb.nodesource.com/node_14.x focal/main amd64 nodejs amd64 14.18.1-deb-1node
source1 [25.0 MB]
Fetched 25.0 MB in 1s (31.2 MB/s)
Selecting previously unselected package nodejs.
(Reading database ... 63981 files and directories currently installed.)
Preparing to unpack .../nodejs_14.18.1-deb-1node
source1_amd64.deb ...
Unpacking nodejs (14.18.1-deb-1node
source1) ...
Setting up nodejs (14.18.1-deb-1node
source1) ...
Processing triggers for man-db (2.9.1-1) ...
ubuntu@ip-172-31-44-90:~$ 

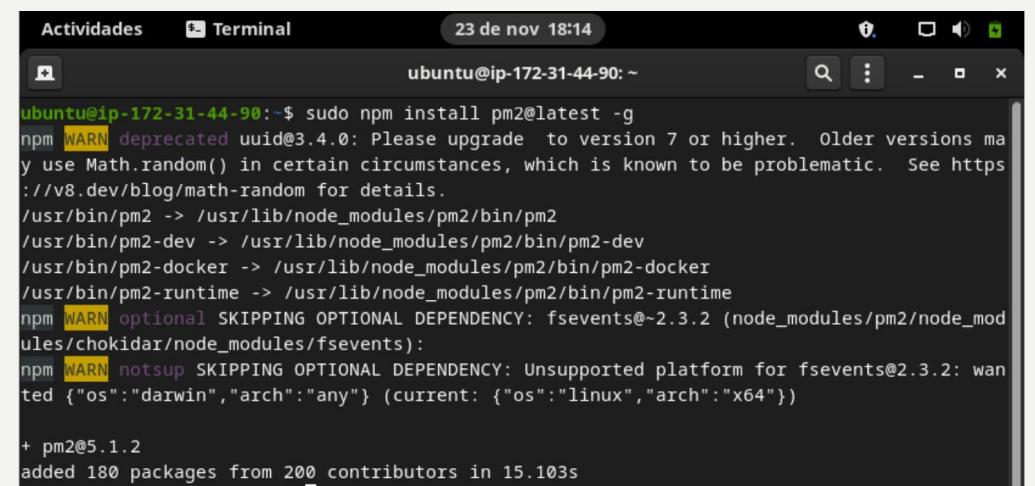
```

- En seguida utilizaremos el comando “**sudo apt install build-essential**”.



```
ubuntu@ip-172-31-44-90:~$ sudo apt install build-essential
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  binutils binutils-common binutils-x86_64-linux-gnu cpp cpp-9 dpkg-dev fakeroot g++
  g++-9 gcc gcc-9 gcc-9-base libalgorithm-diff-perl libalgorithm-diff-xs-perl
  libalgorithm-merge-perl libasan5 libatomic1 libbinutils libc-dev-bin libc6-dev
  libcc1-0 libcrypt-dev libctf-nobfd0 libctf0 libdpkg-perl libfakeroot
  libfile-fcntllock-perl libgcc-9-dev libgomp1 libis122 libitm1 liblsan0 libmpc3
  libquadmath0 libstdc++-9-dev libtsan0 libubsan1 linux-libc-dev make manpages-dev
Suggested packages:
  binutils-doc cpp-doc gcc-9-locales debian-keyring g++-multilib g++-9-multilib
  gcc-9-doc gcc-multilib autoconf automake libtool flex bison gdb gcc-doc gcc-9-multilib
  glibc-doc bzr libstdc++-9-doc make-doc
The following NEW packages will be installed:
  binutils binutils-common binutils-x86_64-linux-gnu build-essential cpp cpp-9 dpkg-dev
  fakeroot g++ g++-9 gcc gcc-9 gcc-9-base libalgorithm-diff-perl
  libalgorithm-diff-xs-perl libalgorithm-merge-perl libasan5 libatomic1 libbinutils
  libc-dev-bin libc6-dev libcc1-0 libcrypt-dev libctf-nobfd0 libctf0 libdpkg-perl
  libfakeroot libfile-fcntllock-perl libgcc-9-dev libgomp1 libis122 libitm1 liblsan0
  libmpc3 libquadmath0 libstdc++-9-dev libtsan0 libubsan1 linux-libc-dev make
  manpages-dev
0 upgraded, 41 newly installed, 0 to remove and 35 not upgraded.
Need to get 43.0 MB of archives.
After this operation, 189 MB of additional disk space will be used.
```

- Por último, instalaremos la paquetería de PM2 necesaria para mantener como un proceso activo el servidor, para ello, utilizamos “**sudo npm install pm2@latest -g**”.



```
Actividades Terminal 23 de nov 18:14
ubuntu@ip-172-31-44-90:~$ sudo npm install pm2@latest -g
npm WARN deprecated uuid@3.4.0: Please upgrade to version 7 or higher. Older versions may use Math.random() in certain circumstances, which is known to be problematic. See https://v8.dev/blog/math-random for details.
/usr/bin/pm2 -> /usr/lib/node_modules/pm2/bin/pm2
/usr/bin/pm2-dev -> /usr/lib/node_modules/pm2/bin/pm2-dev
/usr/bin/pm2-docker -> /usr/lib/node_modules/pm2/bin/pm2-docker
/usr/bin/pm2-runtime -> /usr/lib/node_modules/pm2/bin/pm2-runtime
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@~2.3.2 (node_modules/pm2/node_modules/chokidar/node_modules/fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@2.3.2: wanted {"os": "darwin", "arch": "any"} (current: {"os": "linux", "arch": "x64"})
+ pm2@5.1.2
added 180 packages from 200 contributors in 15.103s
```

Levantamiento del servidor.

Primero editaremos un archivo necesario para la configuración de Nginx, para ello, editamos dicho archivo utilizando el comando “sudo nano /etc/nginx/sites-enabled”, una vez que nos encontremos en el editor, tenemos que bajar hasta el apartado “location” y poner lo siguiente:

```
File Edit View Terminal Tabs Help
GNU nano 4.8                               default
root /var/www/html;

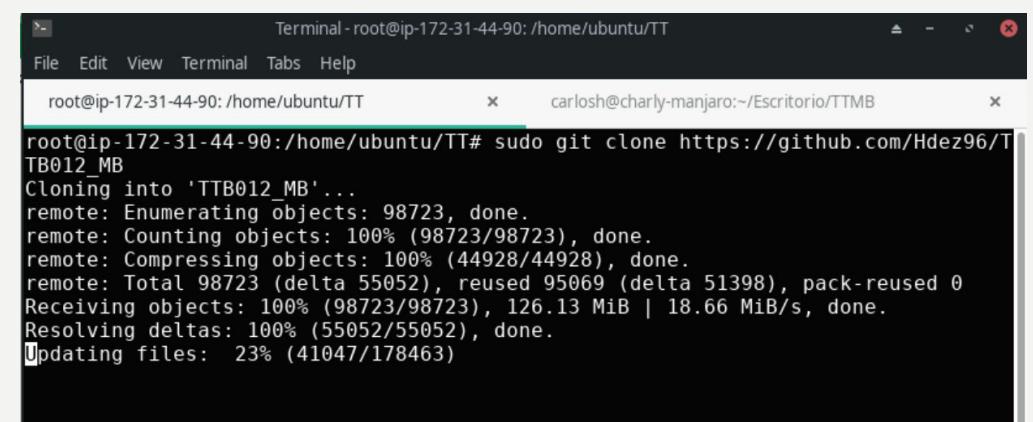
# Add index.php to the list if you are using PHP
index index.html index.htm index.nginx-debian.html;

server_name _;

location / {
    proxy_pass http://localhost:3000;
proxy_http_version 1.1;
proxy_set_header Upgrade $http_upgrade;
proxy_set_header Connection 'upgrade';
proxy_set_header Host $host;
proxy_cache_bypass $http_upgrade;
}

# pass PHP scripts to FastCGI server
#
^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text  ^J Justify  ^C Cur Pos
^X Exit      ^R Read File  ^\ Replace   ^U Paste Text^T To Spell  ^ Go To Line
```

Guardamos el archivo y reiniciamos los servicios de Nginx utilizando el comando: “sudo systemctl restart nginx”. (Este manual asume que el usuario tiene conocimientos para el uso de Git, además de tenerlo instalado en su equipo). Para continuar, es necesario copiar el sistema a la máquina virtual, la forma más sencilla de hacer esto es utilizar la plataforma GitHub. Para ello, utilizaremos el comando “ sudo git clone https://github.com/Hdez96/TTB012_MB“.



The terminal window shows the command "sudo git clone https://github.com/Hdez96/TTB012_MB" being run. The output indicates the cloning process is complete, with 98723 objects enumerated, 100% compressed, and 126.13 MiB received at 18.66 MiB/s. The files are being updated to 23% completion.

```
Terminal - root@ip-172-31-44-90: /home/ubuntu/TT
File Edit View Terminal Tabs Help
root@ip-172-31-44-90: /home/ubuntu/TT          x carlosh@charly-manjaro:~/Escritorio/TTMB x
root@ip-172-31-44-90:/home/ubuntu/TT# sudo git clone https://github.com/Hdez96/TTB012_MB
Cloning into 'TTB012_MB'...
remote: Enumerating objects: 98723, done.
remote: Counting objects: 100% (98723/98723), done.
remote: Compressing objects: 100% (44928/44928), done.
remote: Total 98723 (delta 55052), reused 95069 (delta 51398), pack-reused 0
Receiving objects: 100% (98723/98723), 126.13 MiB | 18.66 MiB/s, done.
Resolving deltas: 100% (55052/55052), done.
Updating files: 23% (41047/178463)
```

Por último, deberemos ingresar a la nueva carpeta que se creó, en este caso es “TTB012_MB”, y después ingresar a la carpeta “FrontEnd”, ahí ingresaremos el comando “pm2 start --name vistas npm -- start”.

```
root@ip-172-31-44-90:/home/ubuntu/TT/TTB012_MB/... x carlosh@charly-manjaro:~/Escritorio/TTMB x

root@ip-172-31-44-90:/home/ubuntu/TT# cd TTB012_MB/
root@ip-172-31-44-90:/home/ubuntu/TT/TTB012_MB# cd FrontEnd/
root@ip-172-31-44-90:/home/ubuntu/TT/TTB012_MB/FrontEnd# sudo pm2 start --name vistas npm -- start
[PM2] Starting /usr/bin/npm in fork_mode (1 instance)
[PM2] Done.



| id | name   | mode | o | status | cpu | memory |
|----|--------|------|---|--------|-----|--------|
| 0  | vistas | fork | 0 | online | 0%  | 14.1mb |



root@ip-172-31-44-90:/home/ubuntu/TT/TTB012_MB/FrontEnd# cd ..
root@ip-172-31-44-90:/home/ubuntu/TT/TTB012_MB# cd BackEnd
root@ip-172-31-44-90:/home/ubuntu/TT/TTB012_MB/BackEnd# ls
BackEnd  database          package.json      server.js
Documents node_modules     registerServiceWorker.js  yarn-error.log
MB.png    nohup.out        route
Models   package-lock.json routes.js
root@ip-172-31-44-90:/home/ubuntu/TT/TTB012_MB/BackEnd# sudo pm2 start --name login npm -- start
[PM2] Starting /usr/bin/npm in fork_mode (1 instance)
[PM2] Done.



| id | name   | mode | o | status | cpu | memory |
|----|--------|------|---|--------|-----|--------|
| 1  | login  | fork | 0 | online | 0%  | 9.4mb  |
| 0  | vistas | fork | 0 | online | 0%  | 44.0mb |


```

Instalación de React JS.

Para instalar react usamos la terminal de comandos y tecleamos el comando `npm install -g npm@latest`, esto va actualizar los paquetes de npm.

```
C:\Windows\system32\cmd.exe

mrc01\AppData\Roaming\npm\npm -> C:\Users\mrc01\AppData\Roaming\npm\node_modules\npm\bin\npm-cli.js
mrc01\AppData\Roaming\npm\npx -> C:\Users\mrc01\AppData\Roaming\npm\node_modules\npm\bin\npx-cli.js
.2
packages from 96 contributors in 6.715s
mrc01>
```

Ahora vamos a instalar el interprete de consola que nos va instalar React facilmente e incluirá el webpack para la compilación y minificación del proyecto. usamos el siguiente comando “`npm install -g create-react-app`”.

```
C:\Windows\system32\cmd.exe

C:\Users\mrc01>npm cache clean --force
npm WARN using --force. Recommended protections disabled.

C:\Users\mrc01>npm install -g create-react-app
npm WARN "trenpm" is not recognized as an internal or external command,
program or executable file.
npm WARN Could not find "trenpm" in the system's PATH.

C:\Users\mrc01>npm install -g create-react-app
npm WARN deprecated tar@2.2.2: This version of tar is no longer supported, and will not receive security updates. Please upgrade asap.

added 67 packages, and audited 68 packages in 2s
1 package requires funding
  run 'npm fund' for details
  2 high severity vulnerabilities
    Some issues need review, and may require choosing a different dependency.
    Run 'npm audit' for details.

C:\Users\mrc01>
```

Requisitos de los sistemas

Para utilizar e instalar (servidores) la aplicación del metrobus se necesita lo siguientes requerimientos mínimos en los distintos dispositivos a utilizar:

Dispositivos móviles.

- Sistema operativo : Android.
- RAM: Mínimo de 512 MB, se recomiendan 2 GB.
- Procesador : Procesador Intel Atom® Z2520 1.2 GHz, o superior.
- Almacenamiento: Entre 850 MB y 1,2 GB, dependiendo de la versión de idioma.
- Software : Navegador de internet y lector de documentos ofimáticos.



PC o Laptop.

- Sistema operativo*: Windows 7, 8, 8.1, 10 o una versión posterior.
- Procesador: procesador Intel Pentium 4 o superior compatible con SSE3
- Espacio en disco: 350 MB.
- Memoria RAM: 512 MB.
- Conexión a internet mínimo 30MB

* Los servidores deben tener Windows Server 2008 R2, Windows Server 2012, Windows Server 2012 R2 o Windows Server 2016.



Errores técnicos comunes

Hardware y/o Software

Otros

SECCIÓN 9

ANEXOS

- Unidades de transporte
- Base de datos
- 00 Descripción por tabla
- 00 Casos de uso
- 00 Descripción de casos de uso
- Estaciones y mantenimiento
- Base de datos
- 00 Descripción por tabla

Unidades de transporte

Base de datos.

Descripción por tabla.

A continuación se presenta de forma individual cada tabal del modelo ER, si se necesita mayor resolución o detalle en el modelo completo presentado en la sección 8 del manual.

Estaciones y mantenimiento

Base de datos.

Descripción por tabla.

A continuación se presenta de forma individual cada tabal del modelo ER, si se necesita mayor resolución o detalle en el modelo completo presentado en la sección 8 del manual.

