

Proyecto – Fase II

Objetivos:

➤ Objetivo General:

Desarrollar una solución de software ERP para la empresa Ninty destinado para trabajar en el sector de las PYMES.

➤ Objetivos Específicos:

- Diseñar la función de usuarios para permitirles la gestión del sistema.
- Diseñar la función de la tienda de módulos para agregarlos a la aplicación.
- Diseñar los módulos que estarán disponibles en la tienda.
- Plantear la propuesta del módulo extra que será implementado en la tienda virtual.
- Definir los requerimientos del sistema.
- Crear la base de datos del proyecto.
- Conocer más sobre los sistemas ERP.

Alcance del proyecto:

Se pretende desarrollar una aplicación basada en los fundamentos de los sistemas ERP destinada para uso de las PYMES que busca ser una solución para agilizar y automatizar los procesos de las empresas contratantes, también busca incorporar nuevas tecnologías a las empresas ayudando en la gestión de tareas y facilitando su realización ahorrando tiempo y dándole mejor control sobre sus operaciones. De este modo las PYMES podrán realizar sus procesos de manera más eficiente con lo cual tendrán un crecimiento mayor en un tiempo menor. Será una aplicación en la cual los propios clientes podrán elegir los módulos que desean cuando sea necesario y pudiendo descartar aquellos que no les son de utilidad.

Panorama general:

Este proyecto tiene por objeto diseñar una herramienta de gestión que se adecúe a cada negocio por medio de módulos que tendrán diversas funciones específicas programadas que podrán ayudar a la empresa en diversos aspectos, el mismo cliente podrá elegir los módulos que desee para formar su propio

sistema. Más adelante el cliente podrá reemplazar los módulos que desee, quitar algunos y agregar nuevos, según las necesidades que la empresa vaya presentando.

La aplicación estará compuesta por múltiples módulos donde cada uno ejecutará determinadas funciones para la gestión de la empresa que los haya seleccionado. Los módulos principales de la aplicación son:

- Módulo usuarios: donde se encuentran los distintos usuarios que poseerá el sistema, como lo son los administradores del sistema, del servicio y los usuarios que harán uso del ERP.
- Tienda de módulos: tienda virtual donde se podrán agregar nuevos módulos y eliminar aquellos que quedaron obsoletos, dentro de los módulos que se tienen se encuentran: gestión de inventario, gestión de ventas, gestión de compras, facturación, reclutamiento, gestión de flotas, blog, eventos, CRM y el módulo propuesto: 'Marketing', para la gestión de acciones promocionales.

Cabe destacar que el sistema se irá actualizando, los módulos que se consideren menos útiles serán eliminados y también se irán agregando nuevos que podrán ser de utilidad para la empresa.

Requerimientos iniciales:

Requerimientos funcionales:

➤ Funciones del módulo usuario:

Ref #	Función	Categoría
R1.1	Habilita o deshabilita los módulos del sistema.	Evidente
R1.2	Registra el control de los clientes que han contratado los servicios.	Oculto
R1.3	Registrar tarjeta de crédito del contratante.	Evidente
R1.4	Registrar los pagos de los servicios prestados.	Oculto
R1.5	Agregar o eliminar módulos al ERP contratado.	Evidente
R1.6	Crear nuevos usuarios del sistema.	Evidente
R1.7	Gestionar las funciones de los módulos.	Evidente

➤ Funciones de la tienda de módulos:

Ref #	Función	Categoría
R2.1	Agregar módulos a la tienda virtual.	Evidente
R2.2	Eliminar módulos no funcionales.	Evidente

➤ Funciones de gestión de inventario:

Ref #	Función	Categoría
R3.1	Debitar los productos luego de una venta.	Oculto
R3.2	Registrar creación, eliminación o modificación de los productos.	Evidente

➤ Funciones de gestión de ventas:

Ref #	Función	Categoría
R4.1	Gestionar las ventas realizadas por la empresa.	Oculto

➤ Funciones de gestión de compras:

Ref #	Función	Categoría
R5.1	Gestionar las compras realizadas por la empresa.	Evidente
R5.2	Realizar un análisis de ingresos.	Evidente

➤ Funciones de facturación:

Ref #	Función	Categoría
R6.1	Realizar la facturación luego de una venta.	Evidente

➤ Funciones de reclutamiento:

Ref #	Función	Categoría
R7.1	Mostrar información del personal.	Evidente
R7.2	Agendar y calendarizar entrevistas de trabajo	Oculto

➤ Funciones de gestión de flotas:

Ref #	Función	Categoría
R8.1	Asignación de viajes	Oculto

➤ Funciones del blog:

Ref #	Función	Categoría
R9.1	Publicación de anuncios.	Evidente
R9.2	Recibir comentarios.	Evidente

➤ Funciones de eventos:

Ref #	Función	Categoría
-------	---------	-----------

R10.1	Crear eventos en el calendario.	Evidente
R10.2	Manejar estados.	Oculto

➤ Funciones de CRM:

Ref #	Función	Categoría
R11.1	Proyectar oportunidades de negocio.	Evidente
R11.2	Mover oportunidades.	Evidente

➤ Funciones de Marketing:

Ref #	Función	Categoría
R12.1	Crear campañas de promoción.	Evidente
R12.2	Realizar seguimiento de clientes.	Oculto

Atributos del sistema:

Atributo	Desarrollo
Seguridad	El sistema garantizará la seguridad de los datos que manejen los usuarios dentro de los distintos módulos que adquieran.
Tiempo de ejecución	Toda la funcionalidad del sistema y transacciones que se ejecuten responderán, en su punto más crítico, en un máximo de 5 segundos.
Estabilidad	Se buscará que el sistema posea una cantidad mínima de fallas además de poder operar adecuadamente con los usuarios.
Facilidad de uso	Sencillez de interpretación de las distintas interfaces del sistema para que los usuarios no tengan complicaciones en su uso y en realizar las tareas que necesiten.
Respuesta de operaciones	El sistema deberá presentar mensajes de error que permitan al usuario identificar el error ocurrido durante sus operaciones.
Aprendizaje sencillo	El usuario será capaz de aprender el funcionamiento del sistema en un plazo menor a 2 horas.
Eficiencia de datos	Las modificaciones que se realicen en la base de datos serán actualizados en un máximo de 5 segundos.
Integridad del sistema	La información del sistema será protegida contra la corrupción y la inconsistencia de los datos

Cientes:

Cliente	Descripción	Módulos
Administrador del sistema	Es quien gestiona los módulos que se presentan y además quien lleva un control sobre los clientes que han contratado el servicio de ERP además de llevar el control sobre los pagos de la aplicación.	Módulo Usuario, Tienda de módulos, sub-módulos presentes en la tienda.
Administrador de servicios	Es quien contrata los servicios del ERP, podrá agregar o eliminar módulos al sistema contratado y además puede crear nuevos usuarios para que utilicen los demás módulos contratados.	Módulo Usuario, tienda de módulos, sub-módulos presentes en la tienda.
Usuario registrado	Es creado por el administrador de servicios y quien gestiona las funciones que ofrece cada módulo en específico.	Sub-módulos presentes en la tienda.

Glosario:

Para una mejor comprensión del proyecto se ha elaborado una lista con algunos términos que pueden llegar a provocar una confusión a los usuarios y que por ello se les ha colocado en la siguiente tabla junto a su significado:

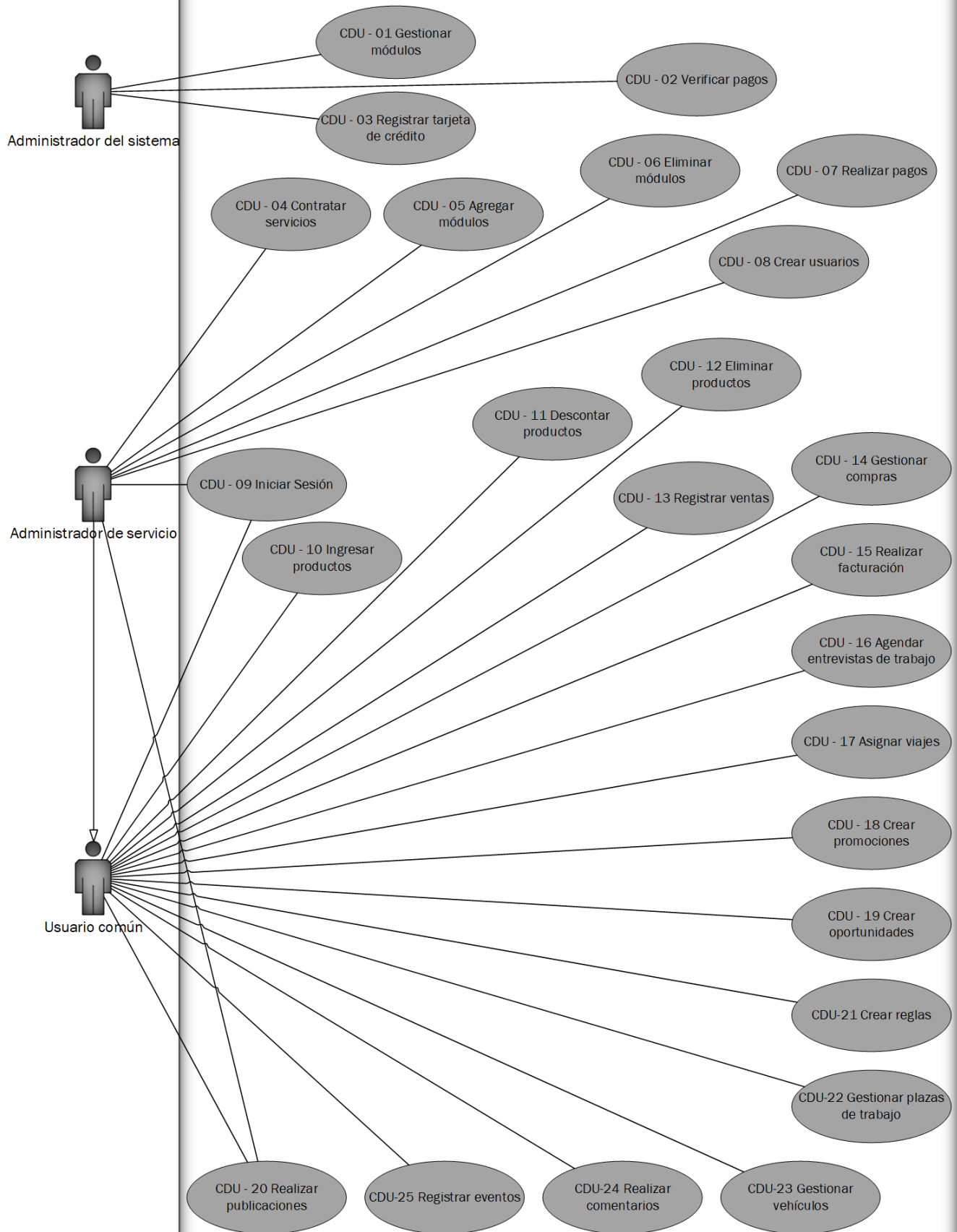
Término	Categoría/Modelo	Descripción
Blog	Requerimientos	Página web que se actualiza regularmente y que suele tratar contenido de interés.
Boolean	Entidad-relación	Tipo de dato que representa valores que pueden ser verdadero o falso.
Char	Entidad-relación	Tipo de dato que representa a un carácter alfanumérico.
Corrupción	Requerimientos	Errores en los datos que se producen durante la transmisión o la recuperación respecto a los datos originales.
Credencial	Entidad-relación	Prueba de autenticación que se le da a una persona.
CRM	Requerimientos	‘Customer Relationship Management’, gestión de relaciones con clientes.
Date	Conceptual	Tipo de dato que almacena valores de fechas.
Double	Entidad-relación	Tipo de dato que representa valores decimales.

ERP	Requerimientos	‘Enterprise Resource Planning’, sistema de planificación de recursos empresariales.
INT	Entidad-relación	Tipo de dato que representa valores enteros.
Interfaz	Requerimientos	Conexión funcional que permite la visualización de información en la pantalla de la computadora.
Marketing	Requerimientos	Técnicas que tienen como objeto mejorar la comercialización de un producto.
Módulo	Requerimientos	Es una porción de un programa que realiza determinadas tareas.
String	Conceptual	Tipo de dato que representa una serie de caracteres
Password	Casos de uso	Contraseña para ingreso a la plataforma.
Plataforma	Casos de uso	Software que se utiliza como base para ejecutar otras aplicaciones compatibles con este.
PYME	Requerimientos	Siglas de Pequeña Y Mediana Empresa
Usuario	Requerimientos	Persona que utiliza un servicio.
Varchar	Entidad-relación	Tipo de dato que almacena una serie de caracteres.

Diagrama de casos de uso:

➤ Alto nivel:

Sistema de planificación de recursos empresariales para PYMES



Nombre del caso de uso:	CDU-01 Gestionar módulos
Actores:	Administrador del sistema
Tipo:	Primario
Descripción:	Permite que el administrador del ERP pueda habilitar o deshabilitar los módulos.

Nombre del caso de uso:	CDU-02 Verificar pagos
Actores:	Administrador del sistema
Tipo:	Primario
Descripción:	Permite que el administrador del ERP verifique los pagos recibidos por los servicios prestados.

Nombre del caso de uso:	CDU-03 Registrar tarjeta de crédito
Actores:	Administrador del sistema
Tipo:	Primario
Descripción:	Al efectuar una contratación con una tarjeta nueva, el administrador deberá registrarla anteriormente para que así el cliente pueda realizar los demás pagos.

Nombre del caso de uso:	CDU-04 Contratar servicios
Actores:	Administrador de servicios
Tipo:	Primario
Descripción:	Por medio de ello nuevas empresas podrán contratar los servicios que ofrece Ninty.

Nombre del caso de uso:	CDU-05 Agregar módulos
Actores:	Administrador de servicios
Tipo:	Secundario
Descripción:	Permite que el administrador cliente agregue módulos para la gestión de su empresa.

Nombre del caso de uso:	CDU-06 Eliminar módulos
Actores:	Administrador de servicios

Tipo:	Secundario
Descripción:	Permite que el administrador cliente elimine aquellos módulos que ya no son de su utilidad.

Nombre del caso de uso:	CDU-07 Realizar pagos
Actores:	Administrador de servicios
Tipo:	Primario
Descripción:	Por este módulo se realiza el pago por los servicios prestados.

Nombre del caso de uso:	CDU-08 Crear usuarios
Actores:	Administrador de servicios
Tipo:	Primario
Descripción:	Por este módulo es que el administrador podrá crear nuevos usuarios que podrán gestionar los demás módulos adquiridos.

Nombre del caso de uso:	CDU-09 Iniciar sesión
Actores:	Administrador de servicios y usuario común
Tipo:	Primario
Descripción:	Permite que los usuarios que compraron la aplicación puedan acceder al portal del sistema y así gestionar los módulos presenta.

Nombre del caso de uso:	CDU-10 Ingresar productos
Actores:	Usuario común
Tipo:	Primario
Descripción:	Permite que los usuarios ingresen productos al inventario.

Nombre del caso de uso:	CDU-11 Descontar productos
Actores:	Usuario común
Tipo:	Primario
Descripción:	Por este módulo el usuario podrá descontar los productos que se encuentren en el inventario.

Nombre del caso de uso:	CDU-12 Eliminar productos
--------------------------------	---------------------------

Actores:	Usuario común
Tipo:	Secundario
Descripción:	Módulo con el cual el usuario eliminará los productos del inventario.

Nombre del caso de uso:	CDU-13 Registrar ventas
Actores:	Usuario común
Tipo:	Primario
Descripción:	Permite que los usuarios registren las ventas realizadas.

Nombre del caso de uso:	CDU-14 Gestionar compras
Actores:	Usuario común
Tipo:	Primario
Descripción:	Permite que los usuarios administren la información de compras realizadas.

Nombre del caso de uso:	CDU-15 Realizar facturación
Actores:	Usuario común
Tipo:	Primario
Descripción:	Módulo que recoge la información de las ventas realizadas y la ordena para ser impresa y entregar una copia al cliente, además envía la información recopilada al módulo de ventas.

Nombre del caso de uso:	CDU-16 Agendar entrevistas de trabajo
Actores:	Usuario común
Tipo:	Primario
Descripción:	Módulo con el cual se agendan y calendarizan las entrevistas de trabajo para personal que se necesite contratar.

Nombre del caso de uso:	CDU-17 Asignar viajes
Actores:	Usuario común
Tipo:	Primario

Descripción:	Con este módulo se tendrá más control sobre el personal que realice viajes para entrega de producto.
---------------------	--

Nombre del caso de uso:	CDU-18 Crear promociones
Actores:	Usuario común
Tipo:	Primario
Descripción:	Función del módulo extra con el cual se podrán gestionar las promociones con el motivo de atraer más clientela.

Nombre del caso de uso:	CDU-19 Crear oportunidades
Actores:	Usuario común
Tipo:	Primario
Descripción:	Con esta funcionalidad se crearán las oportunidades destinadas a los clientes para buscar hacer negocio.

Nombre del caso de uso:	CDU-20 Realizar publicaciones
Actores:	Usuario común y Administrador de servicios
Tipo:	Secundario
Descripción:	Función con la cual los usuarios podrán publicar anuncios en el blog de la empresa.

Nombre del caso de uso:	CDU-21 Crear reglas
Actores:	Usuario común
Tipo:	Primario
Descripción:	Permite la creación de reglas para el manejo de inventarios

Nombre del caso de uso:	CDU-22 Gestionar plazas de trabajo
Actores:	Usuario común
Tipo:	Primario
Descripción:	Por el cual se podrá administrar la información respecto a las plazas que posea la empresa.

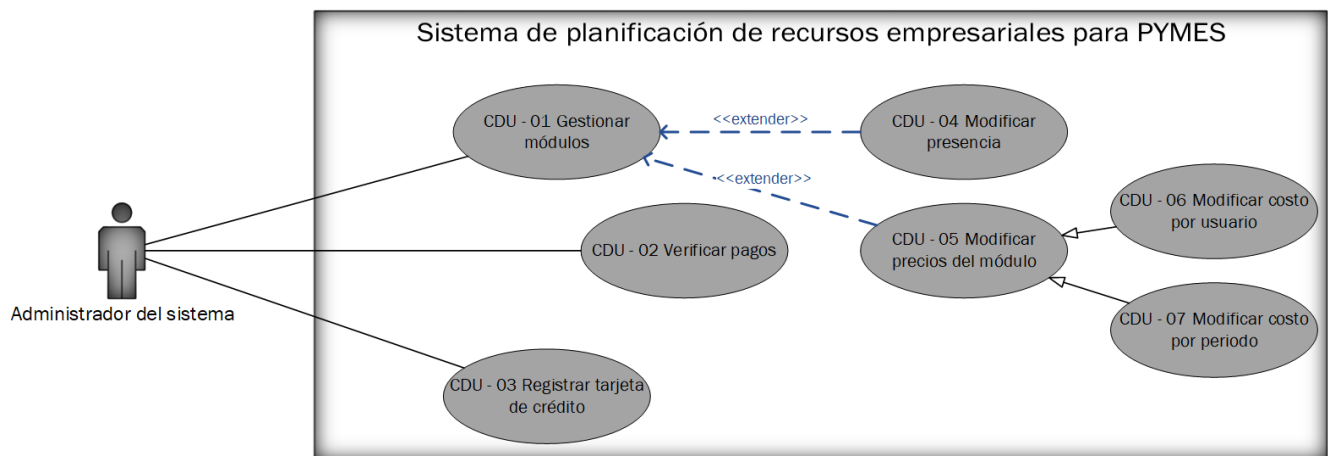
Nombre del caso de uso:	CDU-23 Gestionar vehículos
--------------------------------	----------------------------

Actores:	Usuario común
Tipo:	Primario
Descripción:	Módulo en el cual se manejará la información de los vehículos que posea la empresa.

Nombre del caso de uso:	CDU-24 Realizar comentarios
Actores:	Usuario común
Tipo:	Primario
Descripción:	Módulo por el cual los usuarios podrán realizar comentarios en las publicaciones que se coloquen en el blog.

Nombre del caso de uso:	CDU-25 Registrar eventos
Actores:	Usuario común
Tipo:	Primario
Descripción:	Permite que el usuario registre las futuras actividades que la empresa tendrá.

➤ Explicación expandida:



Nombre del caso de uso:	CDU-01 Gestionar módulos
Actores:	Administrador del sistema
Propósito	Permitir que el administrador pueda gestionar los módulos del sistema.

Tipo:	Primario, Real
Resumen:	Con este módulo el administrador del ERP puede modificar la información presente en los módulos que posee.
Referencias cruzadas:	CDU-04, CDU-05
Flujo normal:	
<ol style="list-style-type: none"> 1. Sistema: muestra la interfaz de inicio del administrador. 2. Usuario: selecciona el botón ‘Gestionar módulos’. 3. Sistema: muestra los módulos que posee el sistema. 4. Usuario: selecciona los módulos que desee modificar. 5. Usuario: selecciona el botón ‘Guardar cambios’. 6. Sistema: muestra los módulos con los cambios efectuados. 	
Flujo alterno:	
<ol style="list-style-type: none"> 1.1. Sistema: no muestra la interfaz de inicio. 1.2. Usuario: selecciona ‘Refrescar página’. 	
Flujo alterno #2:	
<ol style="list-style-type: none"> 5.1. Sistema: no muestra los cambios efectuados. 5.2. Usuario: selecciona de nuevo los módulos y modifica su información. 5.3. Usuario: vuelve a seleccionar el botón ‘Guardar cambios’. 	

Nombre del caso de uso:	CDU-02 Verificar pagos
Actores:	Administrador del sistema
Propósito	Permitir visualizar la información del pago de servicios.
Tipo:	Primario, Real
Resumen:	Permite que el administrador del ERP verifique los pagos recibidos por los servicios prestados.
Referencias cruzadas:	
Flujo normal:	

<ol style="list-style-type: none"> 1. Sistema: muestra la interfaz de inicio del administrador. 2. Usuario: selecciona el botón ‘Pagos’. 3. Sistema: muestra la información de los pagos que se han realizado.
Flujo alterno:
<ol style="list-style-type: none"> 1.1. Sistema: no muestra la interfaz de inicio. 1.2. Usuario: selecciona ‘Refrescar página’.
Flujo alterno #2:
<ol style="list-style-type: none"> 2.1. Sistema: no muestra la información de los pagos. 2.2. Usuario: selecciona ‘Refrescar página’.

Nombre del caso de uso:	CDU-03 Registrar tarjeta de crédito
Actores:	Administrador del sistema
Propósito	Registrar las tarjetas de crédito de los usuarios para que puedan efectuar pagos con ellas.
Tipo:	Primario
Resumen:	Al efectuar una contratación con una tarjeta nueva, el administrador deberá registrarla anteriormente para que así el cliente pueda realizar los demás pagos.
Referencias cruzadas:	
Flujo normal:	
<ol style="list-style-type: none"> 1. Sistema: muestra la interfaz de inicio del administrador. 2. Usuario: selecciona el botón ‘Registro de tarjeta. 3. Sistema: muestra el formulario para registro de tarjeta. 4. Usuario: ingresa los campos que se requieren. 5. Usuario: hace “clic” en el botón ‘Registrar’. 6. Sistema: muestra mensaje de registro exitoso. 	
Flujo alterno:	
<ol style="list-style-type: none"> 1.1. Sistema: no muestra la interfaz de inicio. 1.2. Usuario: selecciona ‘Refrescar página’. 	
Flujo alterno #2:	

3.1. Sistema: no muestra la información de los pagos.

3.2. Usuario: selecciona 'Refrescar página'.

Nombre del caso de uso:	CDU-04 Modificar presencia
Actores:	Administrador del sistema
Propósito	Permitir que el administrador pueda habilitar o deshabilitar módulos.
Tipo:	Primario, Real
Resumen:	Con este módulo el administrador del ERP puede modificar la presencia de un módulo en el sistema habilitándolo o deshabilitándolo.
Referencias cruzadas:	CDU-01
Flujo normal:	
<ol style="list-style-type: none">1. Sistema: muestra la interfaz de inicio del administrador.2. Usuario: selecciona el botón 'Gestionar módulos'.3. Sistema: muestra los módulos que posee el sistema.4. Usuario: selecciona los módulos que desee habilitar o deshabilitar.5. Usuario: selecciona el botón 'Guardar cambios'.6. Sistema: muestra un cuadro de diálogo solicitando un mensaje para mostrar a los clientes las razones de la modificación.7. Usuario: introduce el mensaje.8. Usuario: selecciona 'Aceptar'.9. Sistema: muestra los módulos con los cambios efectuados.	
Flujo alterno #1:	
<ol style="list-style-type: none">1.1. Sistema: no muestra la interfaz de inicio.1.2. Usuario: selecciona 'Refrescar página'.	
Flujo alterno #2:	
<ol style="list-style-type: none">3.1. Sistema: no muestra los módulos del sistema.3.2. Usuario: selecciona 'Refrescar página'.	
Flujo alterno #3:	
<ol style="list-style-type: none">5.1. Sistema: no muestra los cambios efectuados.5.2. Usuario: selecciona de nuevo los módulos y modifica su información.5.3. Usuario: vuelve a seleccionar el botón 'Guardar cambios'.	

Flujo alterno #4:	
8.1. Sistema:	no muestra el mensaje introducido por el usuario.
8.2. Usuario:	introduce nuevamente el mensaje de error.

Nombre del caso de uso:	CDU-05 Modificar precios del módulo
Actores:	Administrador del sistema
Propósito	Permitir que el administrador pueda cambiar los precios de cada módulo.
Tipo:	Secundario, Real
Resumen:	Con este módulo el administrador del ERP puede modificar los precios que tengan colocados los módulos.
Referencias cruzadas:	CDU-01, CDU-06, CDU-07

Flujo normal:	
1.	Sistema: muestra la interfaz de inicio del administrador.
2.	Usuario: selecciona el botón ‘Gestionar módulos’.
3.	Sistema: muestra los módulos que posee el sistema.
4.	Usuario: selecciona la opción ‘Precios’.
5.	Sistema: muestra los precios que tenga cada módulo.
6.	Usuario: modifica los precios que desea.
7.	Usuario: selecciona el botón guardar cambios.
8.	Sistema: muestra los módulos con los cambios efectuados.

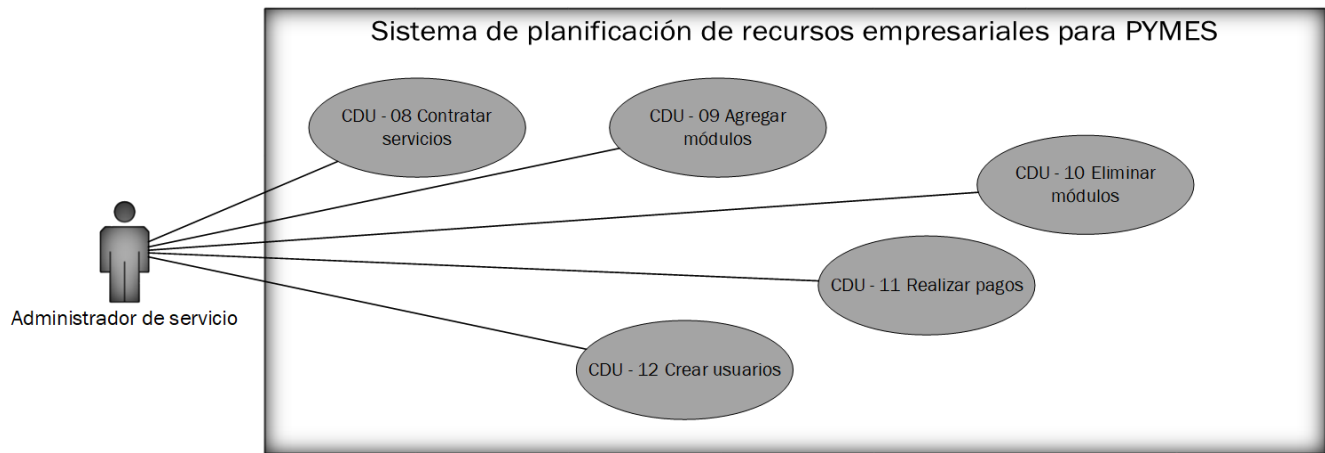
Flujo alterno:	
1.1.	Sistema: no muestra la interfaz de inicio.
1.2.	Usuario: selecciona ‘Refrescar página’.

Flujo alterno #2:	
3.1.	Sistema: no muestra los módulos del sistema.
3.2.	Usuario: selecciona ‘Refrescar página’.

Flujo alterno #3:	
7.1.	Sistema: no muestra los cambios efectuados.
7.2.	Usuario: modifica de nuevo los precios.
7.3.	Usuario: vuelve a seleccionar el botón ‘Guardar cambios’.

Nombre del caso de uso:	CDU-06 Modificar costo por usuario
Actores:	Administrador del sistema
Propósito	Permitir que el administrador pueda cambiar los precios de cada módulo por cada usuario.
Tipo:	Secundario, Real
Resumen:	Con este módulo el administrador del ERP puede modificar los precios que tengan colocados los módulos dependiendo de los usuarios que se le asignen.
Referencias cruzadas:	CDU-05
Flujo normal:	
<ol style="list-style-type: none"> 1. Sistema: muestra la interfaz de inicio del administrador. 2. Usuario: selecciona el botón ‘Gestionar módulos’. 3. Sistema: muestra los módulos que posee el sistema. 4. Usuario: selecciona la opción ‘Precios’. 5. Sistema: muestra los precios que tenga cada módulo. 6. Usuario: selecciona la opción ‘Costo por usuario’. 7. Usuario: modifica los costos que desea. 8. Usuario: selecciona el botón guardar cambios. 9. Sistema: muestra los módulos con los cambios efectuados. 	
Flujo alterno:	
<ol style="list-style-type: none"> 1.1. Sistema: no muestra la interfaz de inicio. 1.2. Usuario: selecciona ‘Refrescar página’. 	
Flujo alterno #2:	
<ol style="list-style-type: none"> 3.1. Sistema: no muestra los módulos del sistema. 3.2. Usuario: selecciona ‘Refrescar página’. 	
Flujo alterno #3:	
<ol style="list-style-type: none"> 7.1. Sistema: no muestra los cambios efectuados. 7.2. Usuario: modifica de nuevo los precios. 7.3. Usuario: vuelve a seleccionar el botón ‘Guardar cambios’. 	

Nombre del caso de uso:	CDU-07 Modificar costo por periodo
Actores:	Administrador del sistema
Propósito	Permitir que el administrador pueda cambiar los precios de cada módulo dependiendo del periodo de tiempo que se contrate.
Tipo:	Secundario, Real
Resumen:	Con este módulo el administrador del ERP puede modificar los precios que tengan colocados los módulos dependiendo del periodo de tiempo que se contrate.
Referencias cruzadas:	CDU-05
Flujo normal:	
<ol style="list-style-type: none"> 1. Sistema: muestra la interfaz de inicio del administrador. 2. Usuario: selecciona el botón ‘Gestionar módulos’. 3. Sistema: muestra los módulos que posee el sistema. 4. Usuario: selecciona la opción ‘Precios’. 5. Sistema: muestra los precios que tenga cada módulo. 6. Usuario: selecciona la opción ‘Costo por periodo.’ 7. Usuario: modifica los costos que desea. 8. Usuario: selecciona el botón guardar cambios. 9. Sistema: muestra los módulos con los cambios efectuados. 	
Flujo alterno:	
<ol style="list-style-type: none"> 1.1. Sistema: no muestra la interfaz de inicio. 1.2. Usuario: selecciona ‘Refrescar página’. 	
Flujo alterno #2:	
<ol style="list-style-type: none"> 3.1. Sistema: no muestra los módulos del sistema. 3.2. Usuario: selecciona ‘Refrescar página’. 	
Flujo alterno #3:	
<ol style="list-style-type: none"> 7.1. Sistema: no muestra los cambios efectuados. 7.2. Usuario: modifica de nuevo los precios. 7.3. Usuario: vuelve a seleccionar el botón ‘Guardar cambios’. 	



Nombre del caso de uso:	CDU-08 Contratar servicios
Actores:	Administrador de servicios
Propósito	Permitir a los clientes contratar servicios.
Tipo:	Primario, Real
Resumen:	Por medio de ello nuevas empresas podrán contratar los servicios que ofrece Ninty.
Referencias cruzadas:	
Flujo normal:	
<ol style="list-style-type: none"> 1. Sistema: muestra la página de inicio de la plataforma de módulos. 2. Usuario: selecciona la cantidad de usuarios que utilizarán el servicio. 3. Usuario: selecciona los módulos que desea agregar al servicio. 4. Usuario: selecciona el periodo de cobro. 5. Usuario: selecciona el botón ‘Mostrar desglose’. 6. Sistema: muestra la información seleccionada por el usuario en forma de factura. 7. Usuario: selecciona el botón ‘Comprar’. 8. Sistema: muestra mensaje de transacción completada. 	
Flujo alterno:	
<ol style="list-style-type: none"> 1.1. Sistema: no muestra la interfaz de la plataforma de módulos. 1.2. Usuario: selecciona ‘Refrescar página’. 	
Flujo alterno #2:	

5.1. Sistema: no muestra el desglose de información.
5.2. Usuario: ingresa de nuevo los usuarios.
5.3. Usuario: selecciona de nuevo los módulos.
5.4. Usuario: selecciona de nuevo el botón ‘Mostrar desglose’.
Flujo alterno #3:
7.1. Sistema: muestra un mensaje de ‘Error de transacción’.
7.2. Sistema: redirige a la página de inicio de la plataforma de módulos.

Nombre del caso de uso:	CDU-09 Agregar módulos
Actores:	Administrador de servicios
Propósito	Agregar nuevos módulos a la lista de servicios contratados.
Tipo:	Secundario, Real
Resumen:	Permite que el administrador cliente agregue módulos para la gestión de su empresa.
Referencias cruzadas:	
Flujo normal:	
1. Sistema: muestra la página de inicio de la plataforma de módulos. 2. Usuario: selecciona los nuevos módulos que se agregarán al servicio. 3. Usuario: selecciona el botón ‘Mostrar desglose’. 4. Sistema: muestra la información seleccionada por el usuario en forma de factura. 5. Usuario: selecciona el botón ‘Agregar’. 6. Sistema: muestra mensaje de transacción completada.	
Flujo alterno:	
1.1. Sistema: no muestra la interfaz de la plataforma de módulos. 1.2. Usuario: selecciona ‘Refrescar página’.	
Flujo alterno #2:	
3.1. Sistema: no muestra el desglose de información. 3.2. Usuario: selecciona de nuevo los módulos. 3.3. Usuario: selecciona de nuevo el botón ‘Mostrar desglose’.	
Flujo alterno #3:	

- 6.1. Sistema: muestra un mensaje de ‘Error de transacción’.
- 6.2. Sistema: redirige a la página de inicio de la plataforma de módulos.

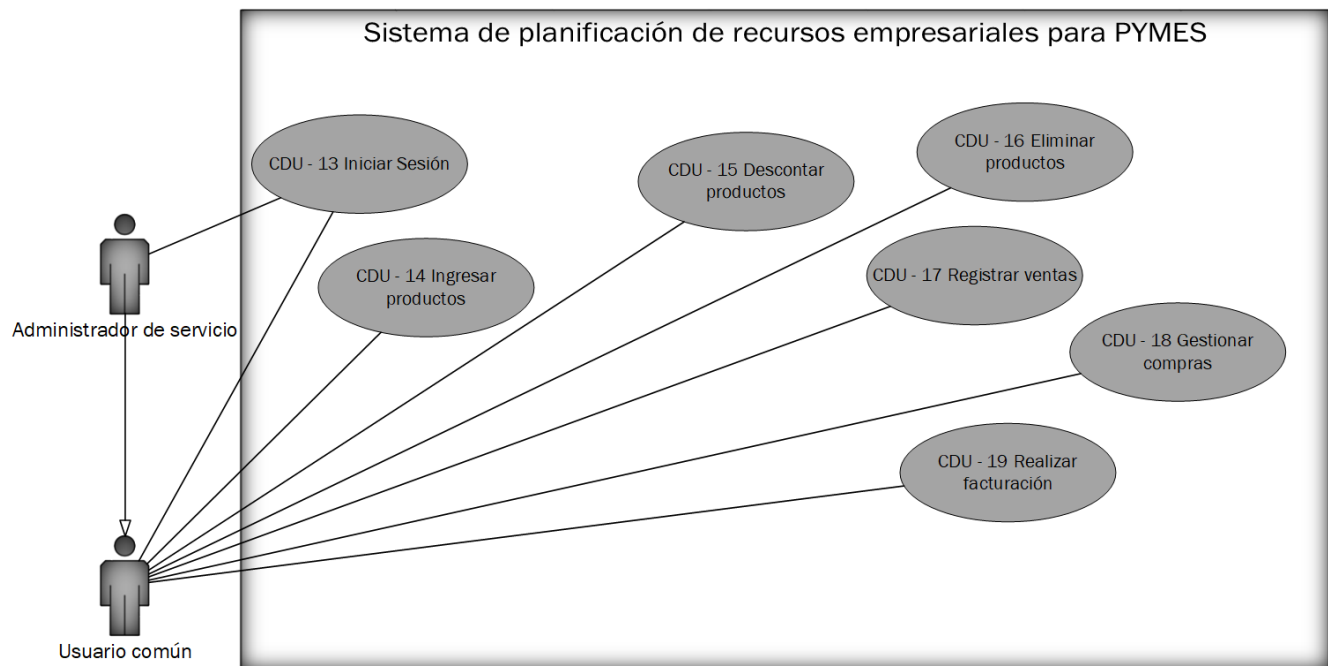
Nombre del caso de uso:	CDU-10 Eliminar módulos
Actores:	Administrador de servicios
Propósito	Eliminar módulos de la lista de servicios contratados.
Tipo:	Secundario, Real
Resumen:	Permite que el administrador cliente elimine aquellos módulos que ya no son de su utilidad.
Referencias cruzadas:	
Flujo normal:	
<ol style="list-style-type: none"> 1. Sistema: muestra la página de inicio de la plataforma de módulos. 2. Usuario: selecciona el botón ‘Eliminar módulos’. 3. Sistema: redirecciona a la página de módulos del usuario. 4. Usuario: desmarca los módulos que ya no serán parte de su sistema. 5. Usuario: selecciona el botón ‘Confirmar’. 6. Sistema: muestra mensaje de transacción completada. 	
Flujo alterno:	
<ol style="list-style-type: none"> 1.1. Sistema: no muestra la interfaz de la plataforma de módulos. 1.2. Usuario: selecciona ‘Refrescar página’. 	
Flujo alterno #2:	
<ol style="list-style-type: none"> 2.1 Sistema: no muestra la interfaz de la página de módulos del usuario. 2.2 Usuario: selecciona ‘Refrescar página’. 	
Flujo alterno #3:	
<ol style="list-style-type: none"> 5.1 Sistema: muestra mensaje de error de transacción. 5.2 Sistema: retorna a la página de inicio de la plataforma de módulos. 	

Nombre del caso de uso:	CDU-11 Realizar pagos
Actores:	Administrador de servicios

Propósito	Realizar el pago de los módulos adquiridos.
Tipo:	Primario, Real
Resumen:	Por este módulo se realiza el pago por los servicios que la empresa Ninty ha prestado.
Referencias cruzadas:	
Flujo normal:	
<ol style="list-style-type: none"> 1. Sistema: muestra la página de inicio del administrador de servicios. 2. Usuario: selecciona el botón ‘Realizar pagos’. 3. Sistema: redirecciona a la página de pagos. 4. Usuario: selecciona los módulos a pagar. 5. Usuario: selecciona la tarjeta a utilizar. 6. Usuario: selecciona el botón ‘Confirmar’. 7. Sistema: muestra mensaje de transacción completada. 	
Flujo alterno:	
<ol style="list-style-type: none"> 1.1. Sistema: no muestra la interfaz de la página de inicio del administrador de servicios. 1.2. Usuario: selecciona ‘Refrescar página’. 	
Flujo alterno #2:	
<ol style="list-style-type: none"> 3.1. Sistema: no muestra la interfaz de la página de pagos. 3.2. Usuario: selecciona ‘Refrescar página’. 	
Flujo alterno #3:	
<ol style="list-style-type: none"> 6.1. Sistema: muestra mensaje de error de transacción. 6.2. Sistema: retorna a la página de inicio del administrador. 	

Nombre del caso de uso:	CDU-12 Crear usuarios
Actores:	Administrador de servicios
Propósito	Crear usuarios para que administren los módulos adquiridos.
Tipo:	Primario, Real

Resumen:	Por este módulo es que el administrador podrá crear nuevos usuarios que podrán gestionar los demás módulos adquiridos.
Referencias cruzadas:	
Flujo normal:	
<ol style="list-style-type: none"> 1. Sistema: muestra la página de inicio del administrador de servicios. 2. Usuario: selecciona el botón ‘Crear usuarios’. 3. Sistema: redirecciona a la página de creación de usuarios. 4. Sistema: muestra los campos a llenar. 5. Usuario: llena los campos de creación de usuarios. 6. Usuario: selecciona el botón ‘Crear usuario’. 7. Sistema: muestra mensaje de creación exitosa. 	
Flujo alterno:	
<ol style="list-style-type: none"> 1.1. Sistema: no muestra la interfaz de la página de inicio del administrador de servicios. 1.2. Usuario: selecciona ‘Refrescar página’. 	
Flujo alterno #2:	
<ol style="list-style-type: none"> 3.1. Sistema: no muestra la interfaz de la página de creación de usuarios. 3.2. Usuario: selecciona ‘Refrescar página’. 	
Flujo alterno #3:	
<ol style="list-style-type: none"> 6.1. Sistema: muestra mensaje de error de transacción. 6.2. Sistema: retorna a la página de inicio del administrador. 	



Nombre del caso de uso:	CDU-13 Iniciar sesión
Actores:	Administrador de servicios y usuario común
Propósito	Permitir que el usuario inicie sesión
Tipo:	Primario, Real
Resumen:	Permite que los usuarios que compraron la aplicación puedan acceder al portal del sistema y así gestionar los módulos presenta.
Referencias cruzadas:	
Flujo normal:	
<ol style="list-style-type: none"> 1. Sistema: muestra los campos para ingreso de “nombre de usuario” y “contraseña”. 2. Usuario: ingresa el nombre de usuario correspondiente. 3. Sistema: solicita el ingreso del password. 4. Usuario: introduce el password. 5. Usuario: hace “clic” en el botón ‘Ingresar’. 6. Sistema: muestra la página de inicio para los usuarios registrados. 	
Flujo alterno:	
<ol style="list-style-type: none"> 5.1 Sistema: sale mensaje de error de credenciales. 5.2 Usuario: ingresa de nuevo los campos para inicio de sesión. 	

Nombre del caso de uso:	CDU-14 Ingresar productos
Actores:	Usuario común
Propósito	Agregar productos al inventario.
Tipo:	Primario, Real
Resumen:	Permite que los usuarios ingresen productos al inventario.
Referencias cruzadas:	
Flujo normal:	
<ol style="list-style-type: none"> 1. Sistema: muestra la página de inicio del usuario común. 2. Usuario: selecciona el módulo de inventario. 3. Sistema: redirecciona a la página del inventario. 4. Usuario: selecciona el botón ‘Agregar productos’. 5. Usuario: introduce los campos. 6. Usuario: selecciona el botón ‘Confirmar’. 7. Sistema: muestra mensaje de creación exitosa. 	
Flujo alterno:	
<ol style="list-style-type: none"> 6.1.Sistema: muestra mensaje de error al agregar productos. 6.2.Sistema: limpia los campos de ‘Agregar productos’. 6.3.Usuario: introduce los campos. 	

Nombre del caso de uso:	CDU-15 Descontar productos
Actores:	Usuario común
Propósito	Descontar productos del inventario
Tipo:	Primario, Real
Resumen:	Por este módulo el usuario podrá descontar los productos que se encuentren en el inventario.
Referencias cruzadas:	
Flujo normal:	

1. Sistema: muestra la página de inicio del usuario común.
2. Usuario: selecciona el módulo de inventario.
3. Sistema: redirecciona a la página del inventario.
4. Usuario: selecciona el botón 'Descontar productos'.
5. Usuario: selecciona los productos a descontar.
6. Usuario: selecciona el botón 'Confirmar'.
7. Sistema: muestra mensaje de descuento exitoso.

Flujo alterno:

- 6.1.Sistema: muestra mensaje de error al descontar productos.
- 6.2.Sistema: retorna a la página de inicio del usuario común.

Nombre del caso de uso:	CDU-16 Eliminar productos
Actores:	Usuario común
Propósito	Eliminar productos del inventario.
Tipo:	Secundario, Real
Resumen:	Módulo con el cual el usuario eliminará los productos del inventario.
Referencias cruzadas:	

Flujo normal:

1. Sistema: muestra la página de inicio del usuario común.
2. Usuario: selecciona el módulo de inventario.
3. Sistema: redirecciona a la página del inventario.
4. Usuario: selecciona el botón 'Eliminar productos'.
5. Usuario: selecciona los productos a eliminar.
6. Usuario: selecciona el botón 'Confirmar'.
7. Sistema: muestra mensaje de borrado exitoso.

Flujo alterno:

- 6.1.Sistema: muestra mensaje de error al borrar productos.
- 6.2.Sistema: retorna a la página de inicio del usuario común.

Nombre del caso de uso:	CDU-17 Registrar ventas
--------------------------------	-------------------------

Actores:	Usuario común
Propósito	Registrar las ventas realizadas.
Tipo:	Primario, Real
Resumen:	Permite que los usuarios registren las ventas realizadas.
Referencias cruzadas:	
Flujo normal:	
<ol style="list-style-type: none"> 1. Sistema: muestra la página de inicio del usuario común. 2. Usuario: selecciona el módulo de ventas. 3. Sistema: redirecciona a la página de registro de ventas. 4. Usuario: selecciona el botón ‘Registrar ventas’. 5. Usuario: introduce los campos. 6. Usuario: selecciona el botón ‘Confirmar’. 7. Sistema: muestra mensaje de registro exitosa. 	
Flujo alterno:	
<ol style="list-style-type: none"> 6.1.Sistema: muestra mensaje de error al registrar venta. 6.2.Sistema: limpia los campos de ‘Registrar ventas’. 6.3.Usuario: introduce los campos. 	

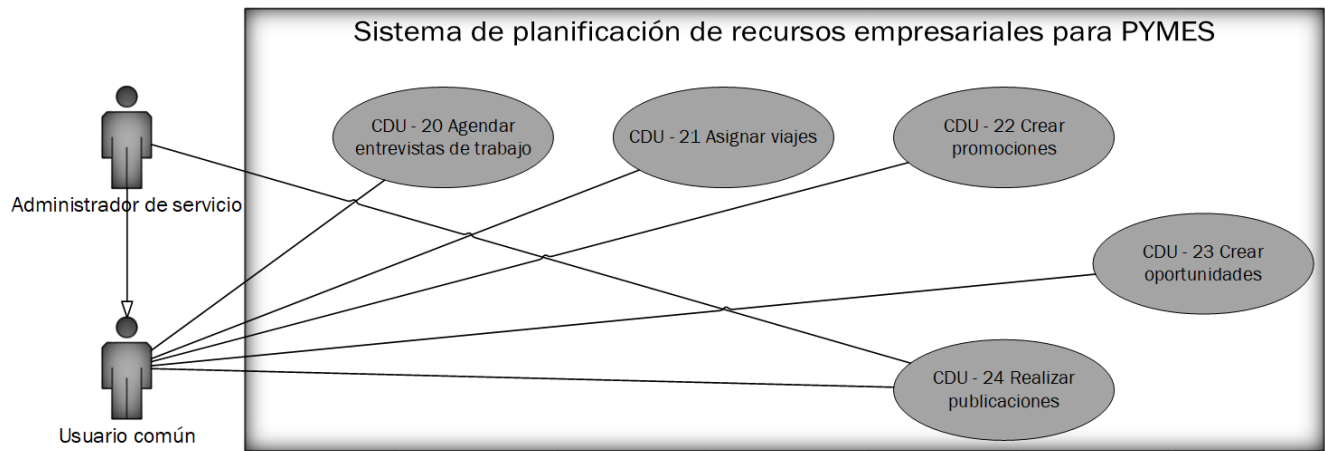
Nombre del caso de uso:	CDU-18 Gestionar compras
Actores:	Usuario común
Propósito	Administrar las compras
Tipo:	Primario, Real
Resumen:	Permite que los usuarios administren la información de compras realizadas.
Referencias cruzadas:	
Flujo normal:	

1. Sistema: muestra la página de inicio del usuario común.
2. Usuario: selecciona el módulo de compras.
3. Sistema: redirecciona a la página de compras.
4. Sistema: muestra la información de las compras realizadas.

Flujo alterno:

- 3.1.Sistema: no muestra información de las compras.
- 3.2.Usuario: selecciona refrescar la página

Nombre del caso de uso:	CDU-19 Realizar facturación
Actores:	Usuario común
Propósito	Registrar la información de los productos vendidos
Tipo:	Primario, Real
Resumen:	Módulo que recoge la información de las ventas realizadas y la ordena para ser impresa y entregar una copia al cliente, además envía la información recopilada al módulo de ventas.
Referencias cruzadas:	
Flujo normal:	
<ol style="list-style-type: none"> 1. Sistema: muestra la página de inicio del usuario común. 2. Usuario: selecciona el módulo de facturación. 3. Sistema: redirecciona a la página de facturación. 4. Sistema: muestra la información de los productos vendidos. 	
Flujo alterno:	
<ol style="list-style-type: none"> 3.1.Sistema: no muestra información de las ventas. 3.2.Usuario: selecciona refrescar la página 	



Nombre del caso de uso:	CDU-20 Agendar entrevistas de trabajo
Actores:	Usuario común
Propósito	Calendarizar las entrevistas de trabajo
Tipo:	Primario, Real
Resumen:	Módulo con el cual se agendan y calendarizan las entrevistas de trabajo para personal que se necesite contratar.
Referencias cruzadas:	
Flujo normal:	
1. Sistema: muestra la página de inicio del usuario común. 2. Usuario: selecciona el módulo de reclutamiento. 3. Sistema: redirecciona a la página de reclutamiento. 4. Sistema: muestra la información de las entrevistas a realizar. 5. Usuario: selecciona la información a modificar. 6. Usuario: selecciona el botón 'Confirmar'. 7. Sistema: muestra mensaje de registro exitoso.	
Flujo alterno:	
6.1.Sistema: muestra mensaje de error de registro. 6.2.Sistema: retorna a la página de reclutamiento.	

Nombre del caso de uso:	CDU-21 Asignar viajes
Actores:	Usuario común

Propósito	Asignar viajes para el transporte de productos
Tipo:	Primario, Real
Resumen:	Con este módulo se tendrá más control sobre el personal que realice viajes para entrega de producto.
Referencias cruzadas:	
Flujo normal:	
<ol style="list-style-type: none"> 1. Sistema: muestra la página de inicio del usuario común. 2. Usuario: selecciona el módulo de gestión de flotas. 3. Sistema: redirecciona a la página de gestión de flotas. 4. Sistema: muestra la información de los viajes a realizar. 5. Usuario: selecciona la opción ‘Asignación’. 6. Sistema: muestra formulario de ingreso. 7. Usuario: selecciona la información a ingresar. 8. Usuario: selecciona el botón ‘Confirmar’. 9. Sistema: muestra mensaje de registro exitoso. 	
Flujo alterno:	
<ol style="list-style-type: none"> 8.1.Sistema: muestra mensaje de error de asignación. 8.2.Sistema: retorna a la página de gestión de flotas. 	

Nombre del caso de uso:	CDU-22 Crear promociones
Actores:	Usuario común
Propósito	Gestionar las promociones a presentar.
Tipo:	Primario, Esencial
Resumen:	Función del módulo extra con el cual se podrán gestionar las promociones con el motivo de atraer más clientela.
Referencias cruzadas:	
Flujo normal:	

1. Sistema: muestra la página de inicio del usuario común.
2. Usuario: selecciona el módulo de marketing.
3. Sistema: redirecciona a la página de marketing.
4. Sistema: muestra la información de las promociones presentes.
5. Usuario: selecciona la opción 'Ingresar promoción'.
6. Sistema: muestra formulario de ingreso.
7. Usuario: introduce los campos de descripción.
8. Usuario: selecciona el botón 'Confirmar'.
9. Sistema: muestra mensaje de registro exitoso.

Flujo alterno:

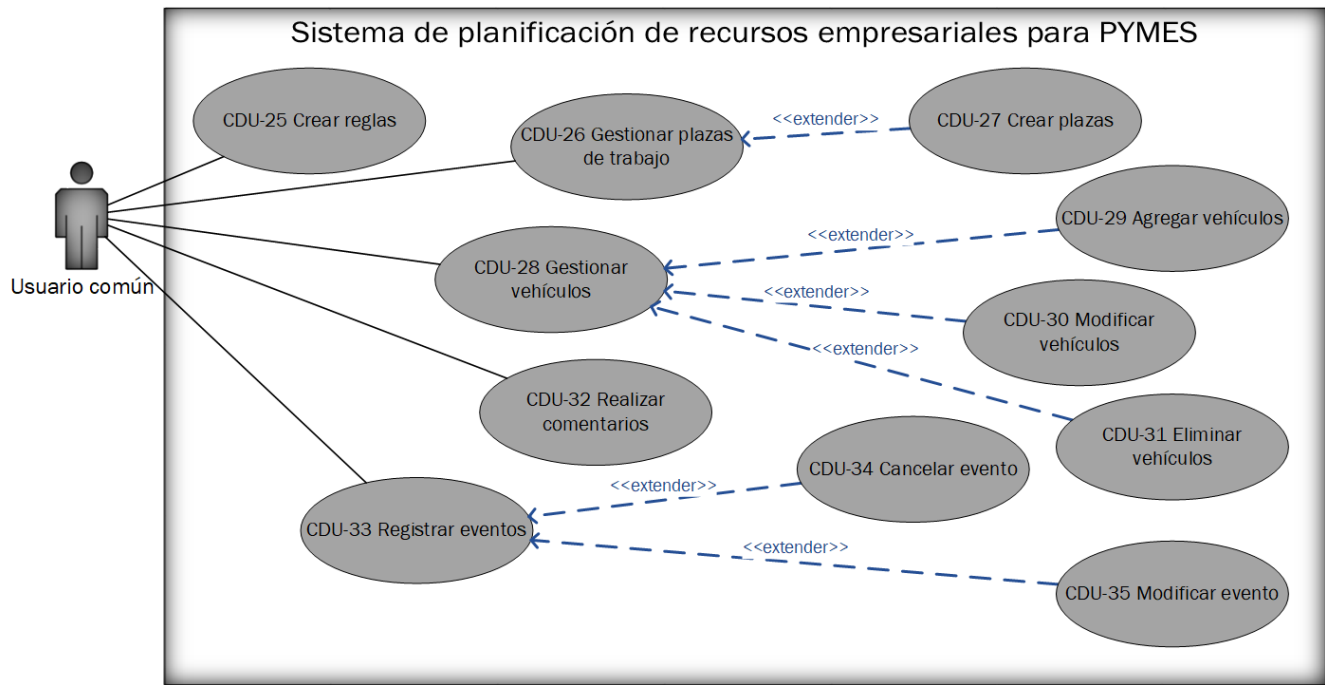
- 8.1.Sistema: muestra mensaje de error de registro de promoción.
- 8.2.Sistema: retorna a la página de marketing.

Nombre del caso de uso:	CDU-23 Crear oportunidades
Actores:	Usuario común
Propósito	Proyectar oportunidades de negocios para sus clientes.
Tipo:	Primario
Resumen:	Con esta funcionalidad se crearán las oportunidades destinadas a los clientes para buscar hacer negocio.
Referencias cruzadas:	
Flujo normal:	
<ol style="list-style-type: none"> 1. Sistema: muestra la página de inicio del usuario común. 2. Usuario: selecciona el módulo de CRM. 3. Sistema: redirecciona a la página de CRM. 4. Sistema: muestra la información de las oportunidades. 5. Usuario: selecciona la opción 'Ingresar oportunidad'. 6. Sistema: muestra formulario de ingreso. 7. Usuario: introduce los campos de descripción. 8. Usuario: selecciona el botón 'Confirmar'. 9. Sistema: muestra mensaje de registro exitoso. 	

Flujo alterno:	
8.1.Sistema: muestra mensaje de error de registro de oportunidad.	
8.2.Sistema: retorna a la página de CRM.	

Nombre del caso de uso:	CDU-24 Realizar publicaciones
Actores:	Usuario común y Administrador de servicios
Propósito	Secundario
Tipo:	Secundario, Real
Resumen:	Función con la cual los usuarios podrán publicar anuncios en el blog de la empresa.
Referencias cruzadas:	

Flujo normal:	
<ol style="list-style-type: none"> 1. Sistema: muestra la página de inicio de usuario. 2. Usuario: selecciona el módulo del blog. 3. Sistema: redirecciona a la página del blog. 4. Sistema: muestra el formulario de publicación 5. Usuario: ingresa los campos del formulario. 6. Usuario: selecciona el botón ‘Publicar’. 7. Sistema: muestra mensaje de registro exitoso. 	
Flujo alterno:	
6.1.Sistema: muestra mensaje de error de publicación.	
6.2.Sistema: retorna a la página de inicio.	



Nombre del caso de uso:	CDU-25 Crear reglas
Actores:	Usuario común
Propósito	Crear reglas para manejar productos
Tipo:	Primario, Real
Resumen:	Permite la creación de reglas para el manejo de productos en el inventario.
Referencias cruzadas:	
Flujo normal:	
<ol style="list-style-type: none"> 1. Sistema: muestra la página de inicio de usuario. 2. Usuario: selecciona el módulo de inventario. 3. Sistema: redirecciona a la página de inventario. 4. Usuario: selecciona el botón 'Reglas'. 5. Sistema: muestra las reglas y el formulario de ingreso de nuevas reglas. 6. Usuario: completa el formulario 7. Usuario: selecciona el botón 'Agregar regla'. 8. Sistema: muestra la nueva regla agregada. 	
Flujo alterno:	

7.1.Sistema: muestra mensaje de error en la creación de regla nueva.
7.2.Sistema: retorna a la página de inventario.
Flujo alterno #2:
7.1.Sistema: no muestra la nueva regla agregada.
7.2.Usuario: selecciona refrescar la página.

Nombre del caso de uso:	CDU-26 Gestionar plazas de trabajo
Actores:	Usuario común
Propósito	Administrar las plazas de trabajo.
Tipo:	Primario, Real
Resumen:	Por el cual se podrá administrar la información respecto a las plazas que posea la empresa.
Referencias cruzadas:	CDU-27
Flujo normal:	
<ol style="list-style-type: none"> 1. Sistema: muestra la página de inicio de usuario. 2. Usuario: selecciona el módulo de Reclutamiento. 3. Sistema: redirecciona a la página de reclutamiento. 4. Usuario: selecciona la opción 'Plazas'. 5. Sistema: muestra la información de las plazas de la empresa. 	
Flujo alterno:	
<ol style="list-style-type: none"> 4.1.Sistema: no muestra la información de las plazas. 4.2.Usuario: refresca la página. 	

Nombre del caso de uso:	CDU-27 Crear plazas
Actores:	Usuario común
Propósito	Crear nuevas plazas de trabajo.
Tipo:	Primario, Real

Resumen:	Por el cual el usuario podrá crear las plazas que se necesiten en la empresa.
Referencias cruzadas:	CDU-26
Flujo normal:	
<ol style="list-style-type: none"> 1. Sistema: muestra la página de inicio de usuario. 2. Usuario: selecciona el módulo de Reclutamiento. 3. Sistema: redirecciona a la página de reclutamiento. 4. Usuario: selecciona la opción ‘Plazas’. 5. Sistema: muestra la información de las plazas de la empresa. 6. Usuario: selecciona la opción ‘Crear nueva plaza’. 7. Sistema: despliega formulario de ingreso de plazas. 8. Usuario: completa los campos. 9. Usuario: selecciona el botón ‘Crear’. 10. Sistema: muestra mensaje de creación exitosa. 	
Flujo alterno:	
<ol style="list-style-type: none"> 9.1.Sistema: mensaje de error de creación de plaza. 9.2.Usuario: vuelve a completar los campos del formulario. 	

Nombre del caso de uso:	CDU-28 Gestionar vehículos
Actores:	Usuario común
Propósito	Gestionar la información de los vehículos
Tipo:	Primario, Real
Resumen:	Módulo en el cual se manejará la información de los vehículos que posea la empresa.
Referencias cruzadas:	CDU-29, CDU-30, CDU-31
Flujo normal:	

1. Sistema: muestra la página de inicio de usuario.
2. Usuario: selecciona el módulo de gestión de flotas.
3. Sistema: redirecciona a la página de gestión de flotas.
4. Usuario: selecciona la opción ‘Vehículos’.
5. Sistema: muestra la información de los vehículos de la empresa.

Flujo alterno:

- 5.1.Sistema: no muestra la información de los vehículos.
- 5.2.Usuario: refresca la página

Nombre del caso de uso:	CDU-29 Agregar vehículos
Actores:	Usuario común
Propósito	Agregar nuevos vehículos
Tipo:	Primario, Real
Resumen:	Módulo en el cual se agregará información de nuevos vehículos.
Referencias cruzadas:	CDU-28
Flujo normal:	
<ol style="list-style-type: none"> 1. Sistema: muestra la página de inicio de usuario. 2. Usuario: selecciona el módulo de gestión de flotas. 3. Sistema: redirecciona a la página de gestión de flotas. 4. Usuario: selecciona la opción ‘Vehículos’. 5. Sistema: muestra la información de los vehículos de la empresa. 6. Usuario: selecciona la opción ‘Agregar’. 7. Sistema. Muestra formulario para agregar información. 8. Usuario: llena los campos. 9. Usuario: presiona el botón ‘Confirmar’. 10. Sistema. Muestra mensaje exitoso. 	
Flujo alterno:	
<ol style="list-style-type: none"> 9.1.Sistema: muestra mensaje de error. 9.2.Usuario: introduce de nuevo los datos. 	

Nombre del caso de uso:	CDU-30 Modificar vehículos
Actores:	Usuario común
Propósito	Modificar información de vehículos
Tipo:	Primario, Real
Resumen:	Módulo en el cual se modificará la información de los vehículos que tenga la empresa.
Referencias cruzadas:	CDU-28
Flujo normal:	
<ol style="list-style-type: none"> 1. Sistema: muestra la página de inicio de usuario. 2. Usuario: selecciona el módulo de gestión de flotas. 3. Sistema: redirecciona a la página de gestión de flotas. 4. Usuario: selecciona la opción ‘Vehículos’. 5. Sistema: muestra la información de los vehículos de la empresa. 6. Usuario: selecciona la opción ‘Modificar información’. 7. Sistema. Muestra los registros de vehículos. 8. Usuario: selecciona los registros a modificar. 9. Usuario: modifica los datos que desee. 10. Sistema: guarda la información modificada. 	
Flujo alterno:	
<ol style="list-style-type: none"> 10.1 Sistema: muestra mensaje de error en la modificación. 10.2 Usuario: modifica de nuevo los datos. 	

Nombre del caso de uso:	CDU-31 Eliminar vehículos
Actores:	Usuario común
Propósito	Elimina los vehículos que estén registrados.
Tipo:	Primario, Real
Resumen:	Módulo en el cual se eliminará la información de los vehículos que tenga la empresa.

Referencias cruzadas:	CDU-28
Flujo normal:	
<ol style="list-style-type: none"> 1. Sistema: muestra la página de inicio de usuario. 2. Usuario: selecciona el módulo de gestión de flotas. 3. Sistema: redirecciona a la página de gestión de flotas. 4. Usuario: selecciona la opción ‘Vehículos’. 5. Sistema: muestra la información de los vehículos de la empresa. 6. Usuario: selecciona la opción ‘Eliminar información’. 7. Sistema. Muestra los registros de vehículos. 8. Usuario: elimina los registros que desee. 9. Sistema: guarda la información modificada. 	
Flujo alterno:	
<ol style="list-style-type: none"> 9.1.Sistema: muestra mensaje de error en la eliminación de datos. 9.2.Usuario: modifica de nuevo los datos. 	

Nombre del caso de uso:	CDU-32 Realizar comentarios
Actores:	Usuario común
Propósito	Realizar un comentario en una publicación.
Tipo:	Primario, Real
Resumen:	Módulo por el cual los usuarios podrán realizar comentarios en las publicaciones que se coloquen en el blog.
Referencias cruzadas:	
Flujo normal:	
<ol style="list-style-type: none"> 1. Sistema: muestra la página de inicio de usuario. 2. Usuario: selecciona el apartado del blog. 3. Sistema: redirecciona a la página del blog. 4. Sistema: muestra las publicaciones del blog. 5. Usuario: selecciona una publicación. 6. Sistema: muestra la información de la publicación. 7. Usuario: selecciona ‘Comentar’. 	

8. Usuario: escribe el comentario.
9. Usuario: envía el comentario.
10. Sistema: muestra la publicación con el comentario actualizado.
Flujo alterno:
9.1.Sistema: muestra mensaje de error al efectuar publicación de comentario.
9.2.Usuario: redacta de nuevo el comentario.

Nombre del caso de uso:	CDU-33 Registrar eventos
Actores:	Usuario común
Propósito	Registrar actividades en el blog.
Tipo:	Primario, Real
Resumen:	Permite que el usuario registre las futuras actividades que la empresa tendrá.
Referencias cruzadas:	CDU-34, CDU-35
Flujo normal:	
1. Sistema: muestra la página de inicio de usuario. 2. Usuario: selecciona el módulo de Eventos. 3. Sistema: redirecciona a la página de Eventos. 4. Sistema: muestra el formulario de publicación 5. Usuario: ingresa los campos del formulario. 6. Usuario: selecciona el botón ‘Publicar’. 7. Sistema: muestra mensaje de registro exitoso.	
Flujo alterno:	
6.1.Sistema: muestra mensaje de error al efectuar publicación de evento. 6.2.Usuario: redacta de nuevo el evento.	

Nombre del caso de uso:	CDU-34 Cancelar evento
Actores:	Usuario común
Propósito	Cancelar actividades en el blog.

Tipo:	Primario, Real
Resumen:	Permite que el usuario cancele algún evento que fue registrado en el blog.
Referencias cruzadas:	CDU-33
Flujo normal:	
<ol style="list-style-type: none"> 1. Sistema: muestra la página de inicio de usuario. 2. Usuario: selecciona el módulo de Eventos. 3. Sistema: redirecciona a la página de Eventos. 4. Usuario: selecciona el botón ‘Cancelar eventos’. 5. Sistema: muestra el registro de eventos. 6. Usuario: selecciona los eventos a cancelar. 7. Usuario: selecciona ‘Finalizar’. 8. Sistema: muestra mensaje de eliminación exitosa. 	
Flujo alterno:	
<ol style="list-style-type: none"> 7.1.Sistema: muestra mensaje de error al efectuar eliminación. 7.2.Usuario: selecciona de nuevo los eventos. 	

Nombre del caso de uso:	CDU-35 Modificar evento
Actores:	Usuario común
Propósito	Modificar actividades en el blog.
Tipo:	Primario, Real
Resumen:	Permite que el usuario modifique algún evento que fue registrado en el blog.
Referencias cruzadas:	CDU-33
Flujo normal:	
<ol style="list-style-type: none"> 1. Sistema: muestra la página de inicio de usuario. 2. Usuario: selecciona el módulo de Eventos. 3. Sistema: redirecciona a la página de Eventos. 4. Usuario: selecciona el botón ‘Modificar eventos’. 5. Sistema: muestra el registro de eventos. 	

- | |
|---|
| <p>6. Usuario: selecciona los eventos a modificar.</p> <p>7. Usuario: selecciona ‘Modificar’.</p> <p>8. Sistema: muestra mensaje de modificación exitosa.</p> |
| <p>Flujo alterno:</p> |
| <p>7.3.Sistema: muestra mensaje de error al efectuar modificaciones.</p> <p>7.4.Usuario: edita de nuevo los eventos.</p> |

Flujo alterno:

7.3.Sistema: muestra mensaje de error al efectuar modificaciones.

7.4.Usuario: edita de nuevo los eventos.
--

Modelo conceptual:

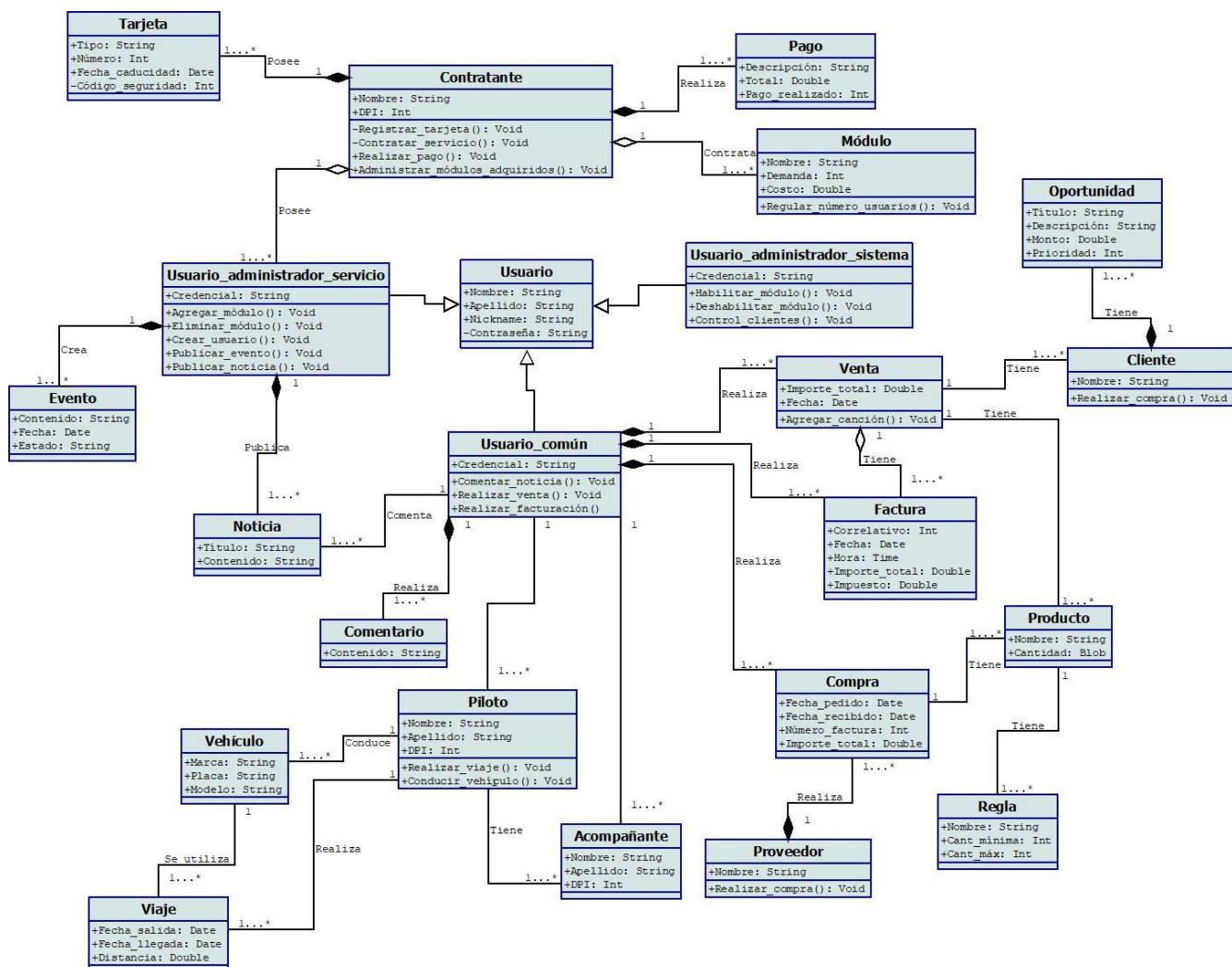


Diagrama entidad-relación:

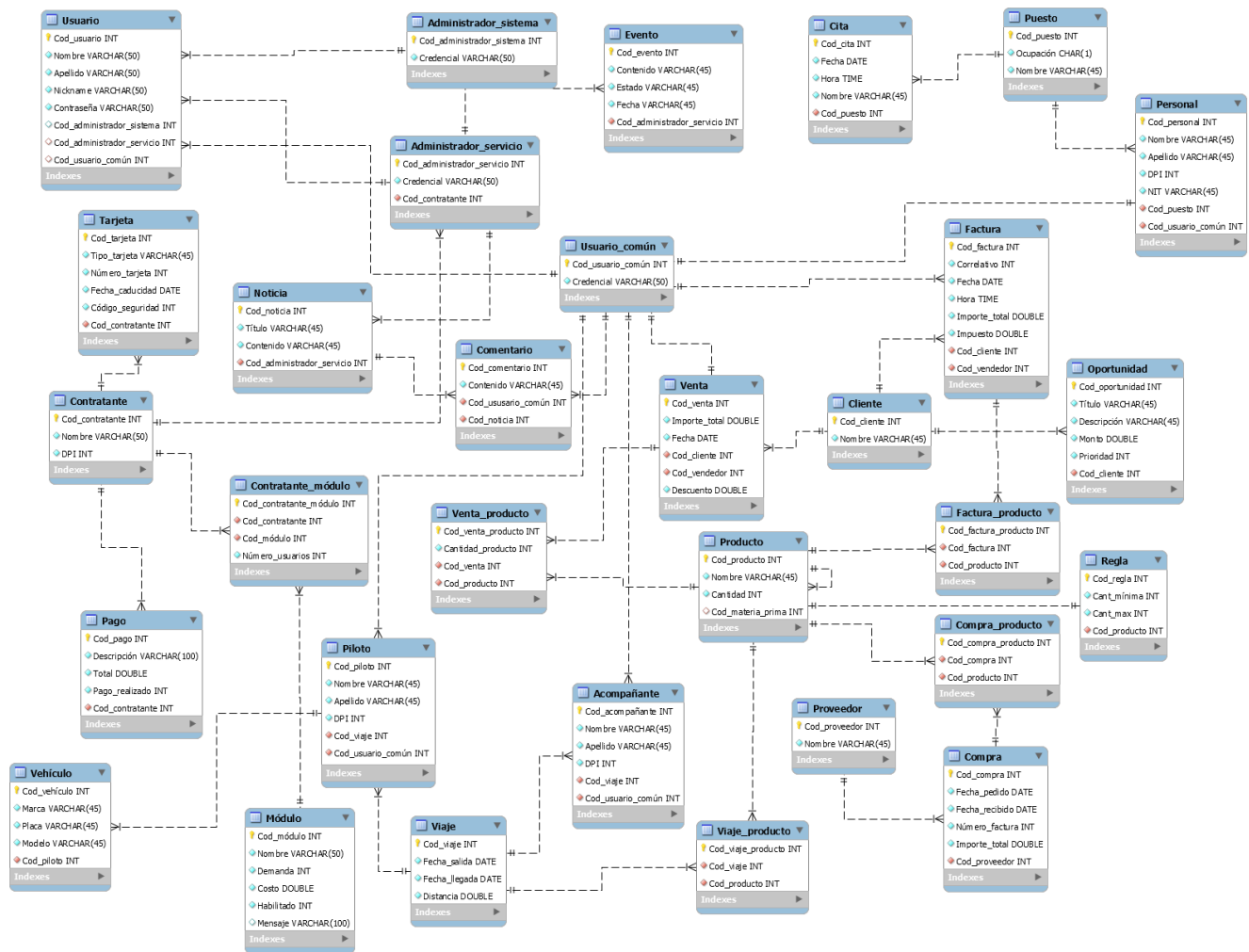


Diagrama de secuencias:

Diagrama de secuencias – Registrar contratantes

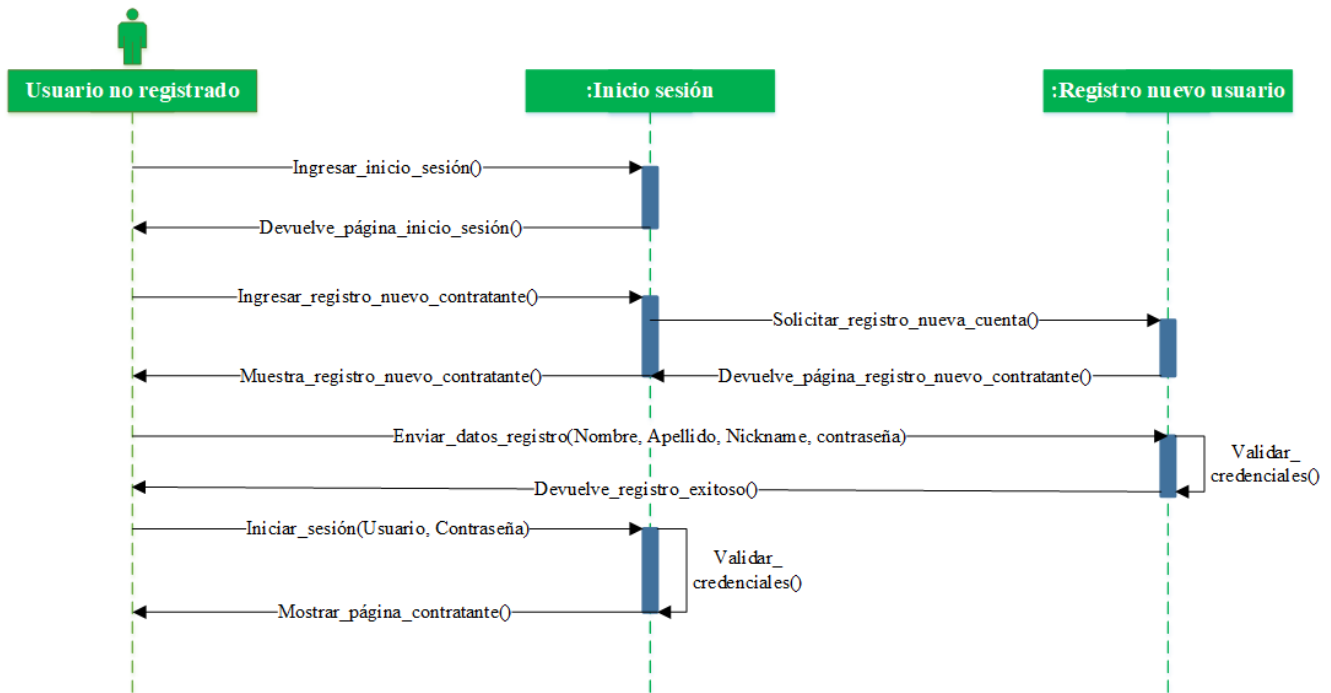


Diagrama de secuencias – Registro de tarjeta

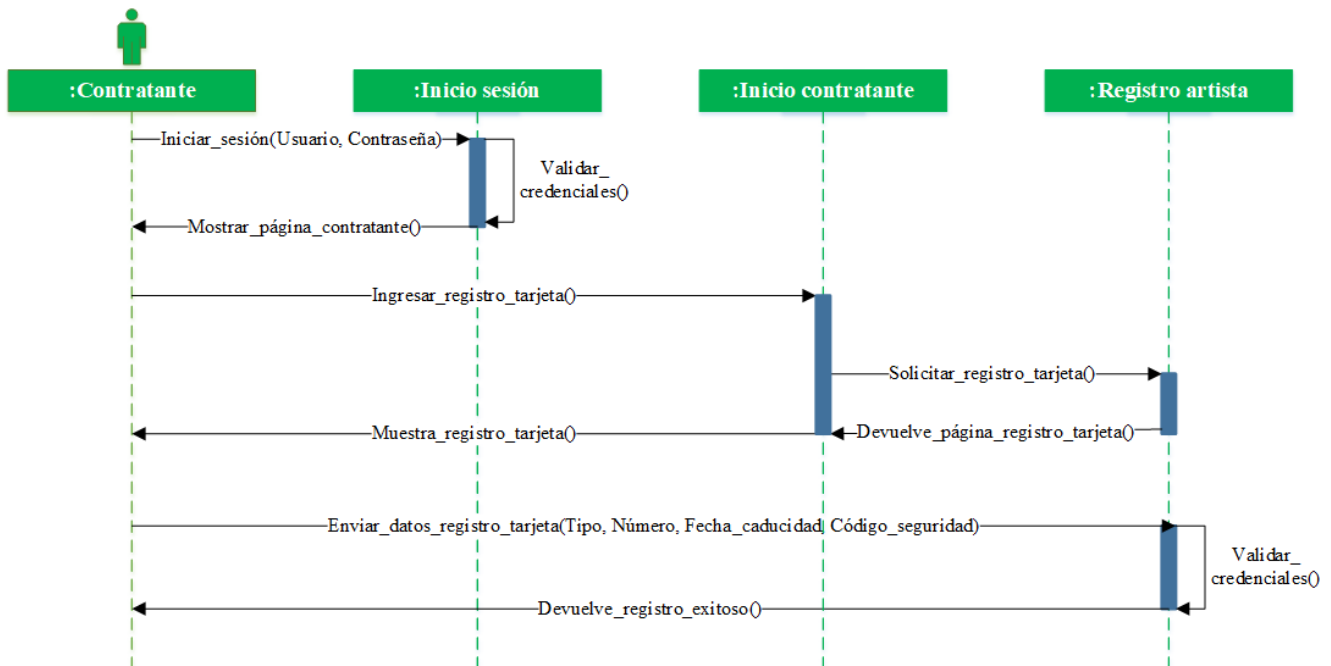


Diagrama de secuencias – Contratar servicios

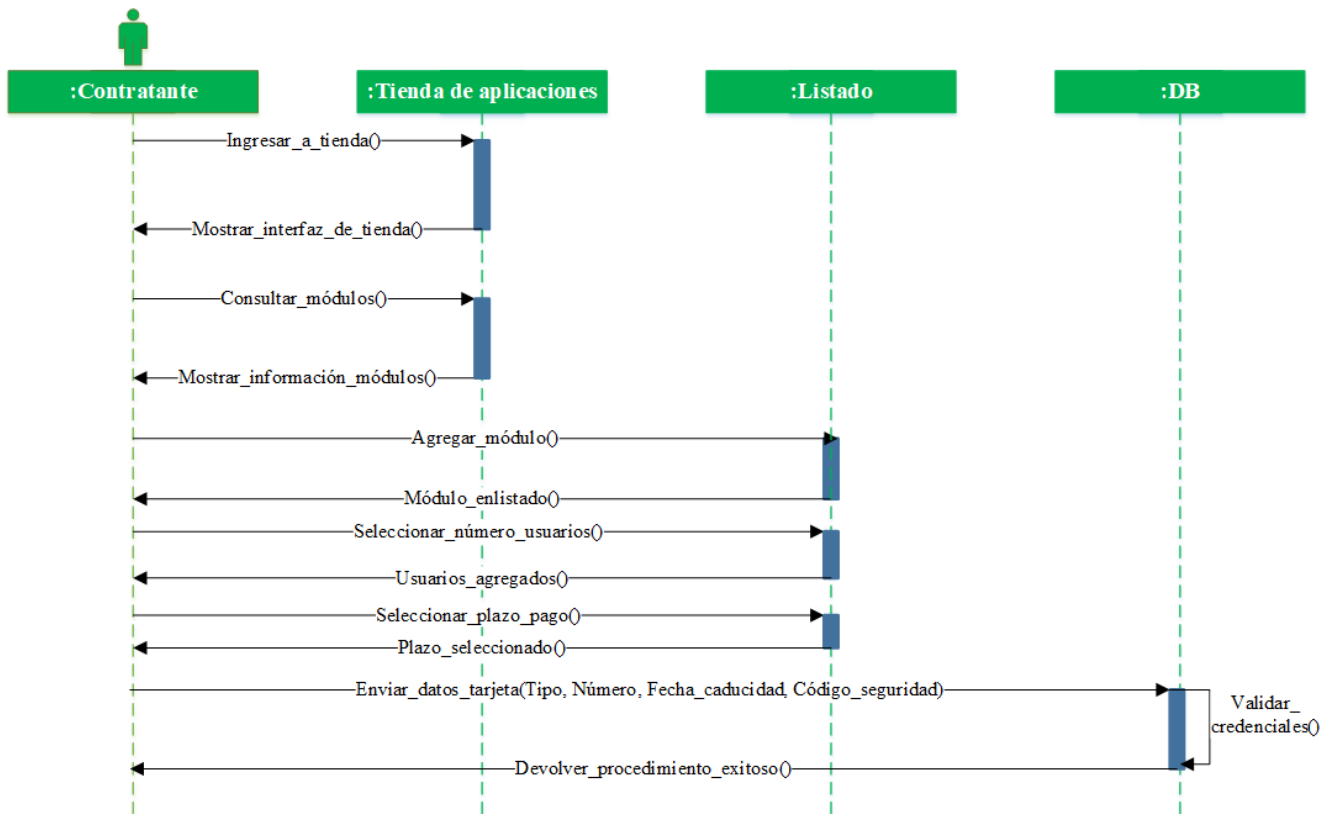


Diagrama de secuencias – Módulo Administración

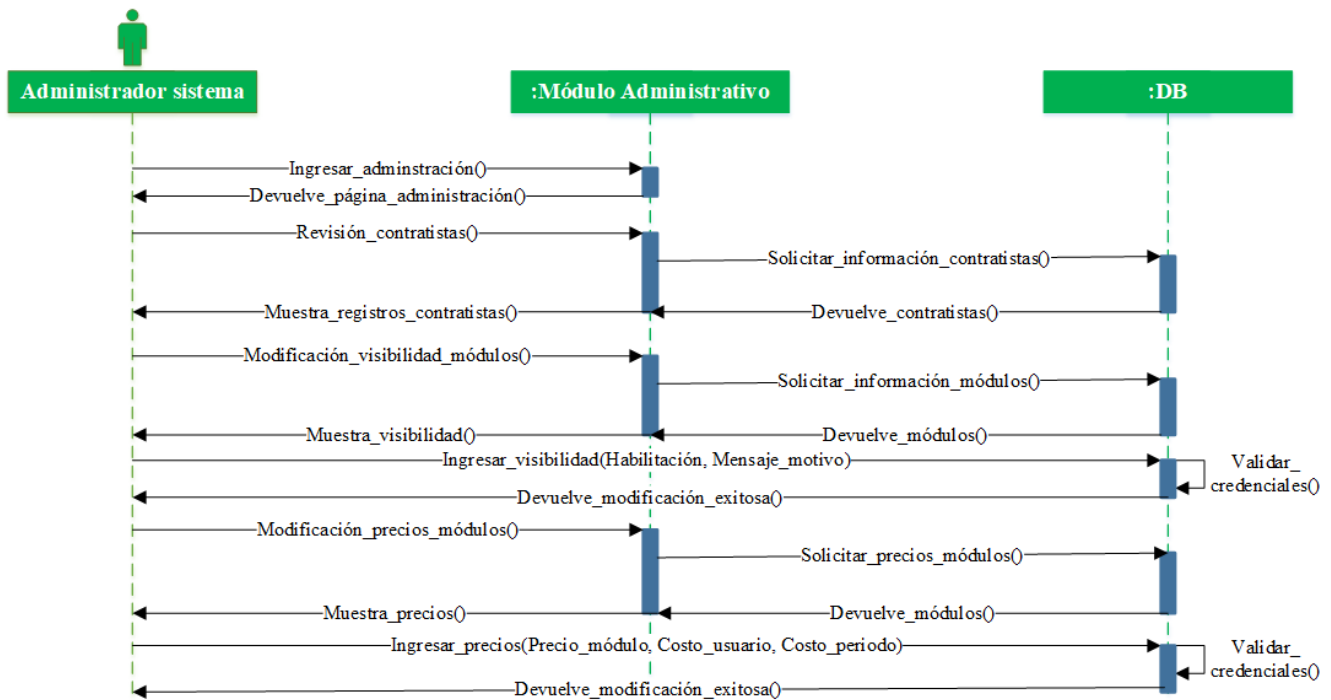


Diagrama de secuencias – Agregar productos

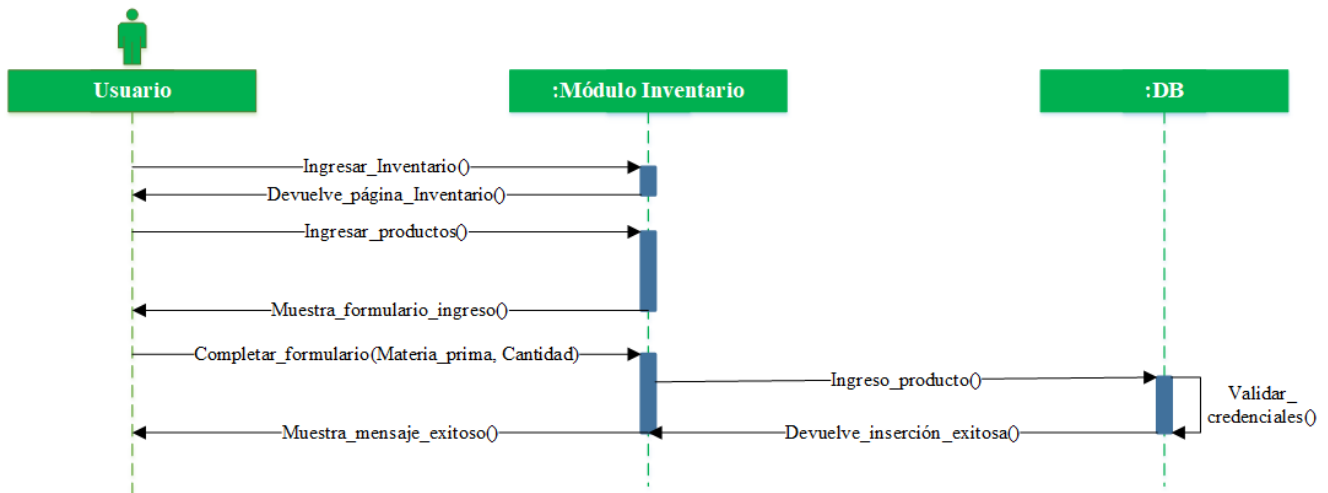


Diagrama de secuencias – Crear reglas

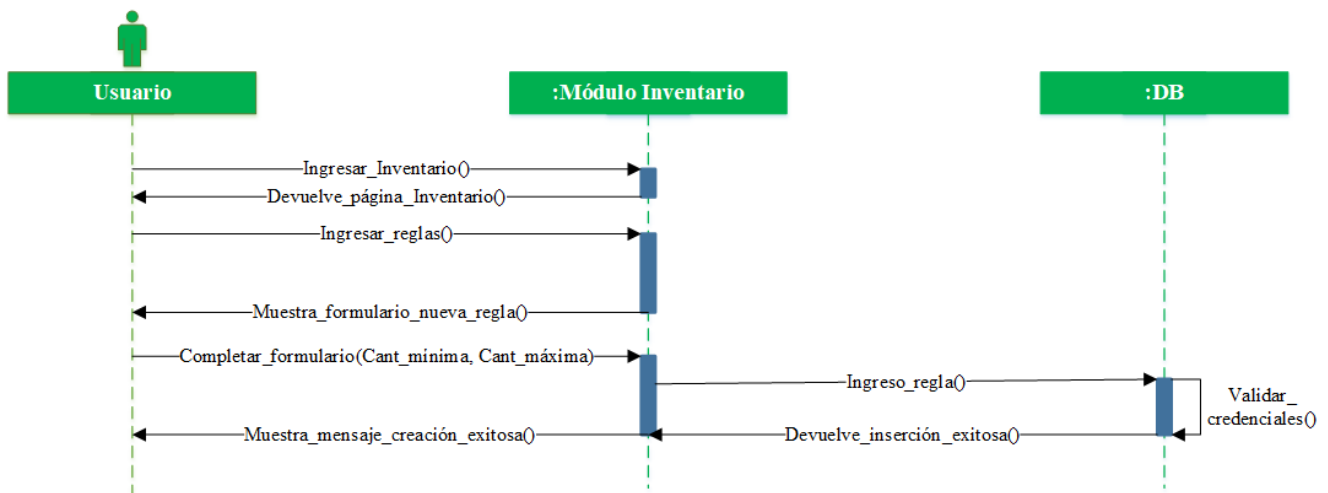


Diagrama de secuencias – Módulo de gestión de ventas

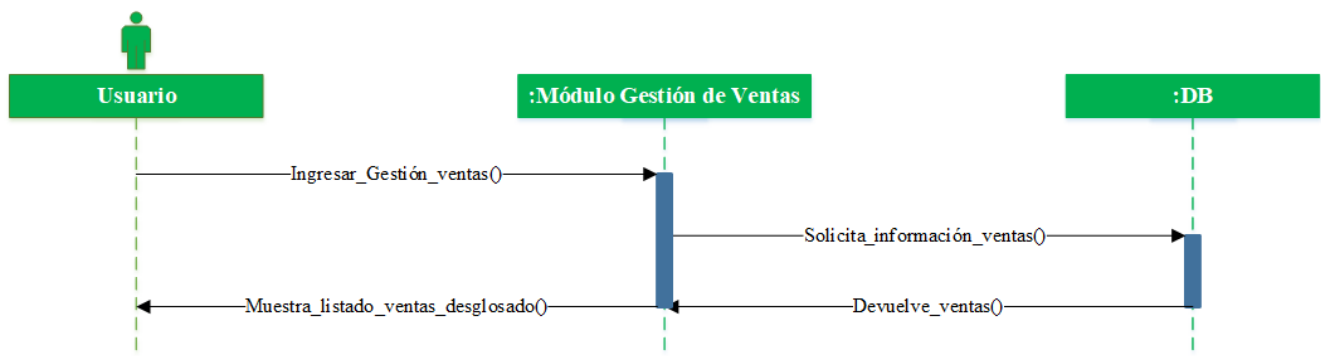


Diagrama de secuencias – Módulo de gestión de compras

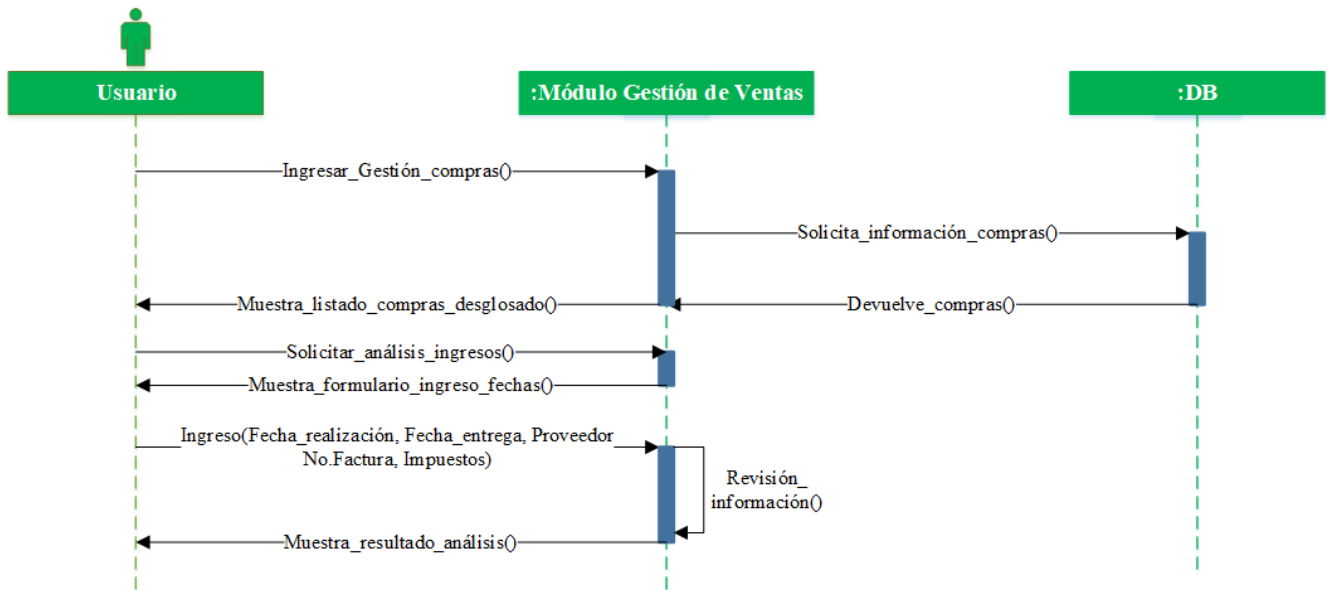


Diagrama de secuencias – Módulo facturación

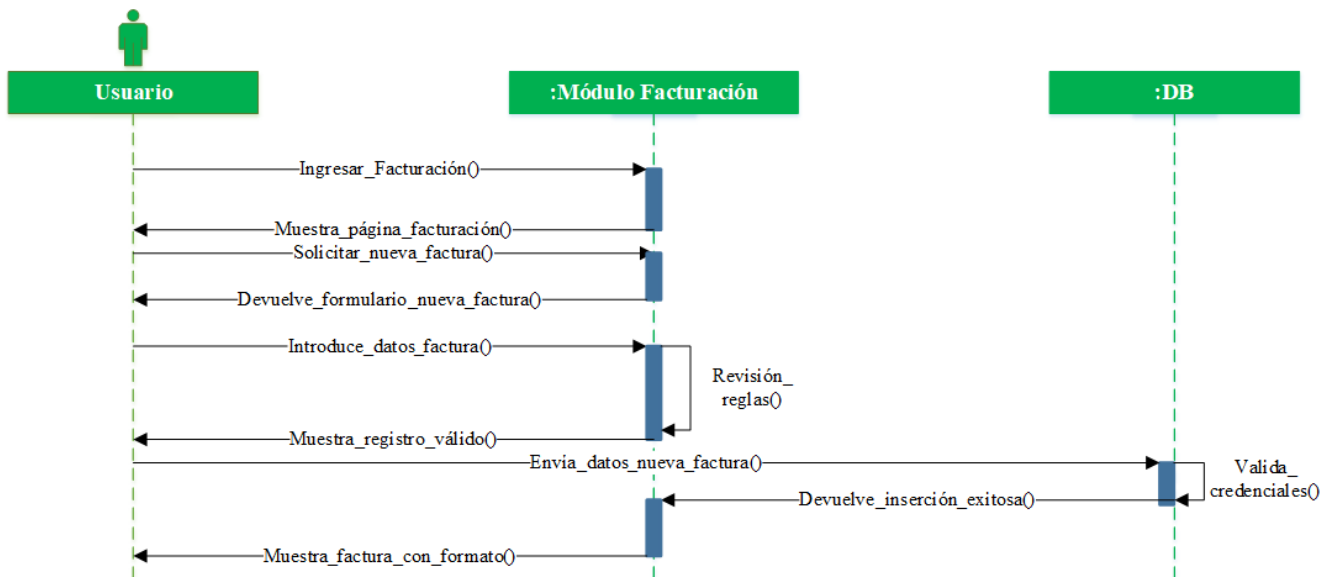


Diagrama de secuencias – Módulo reclutamiento

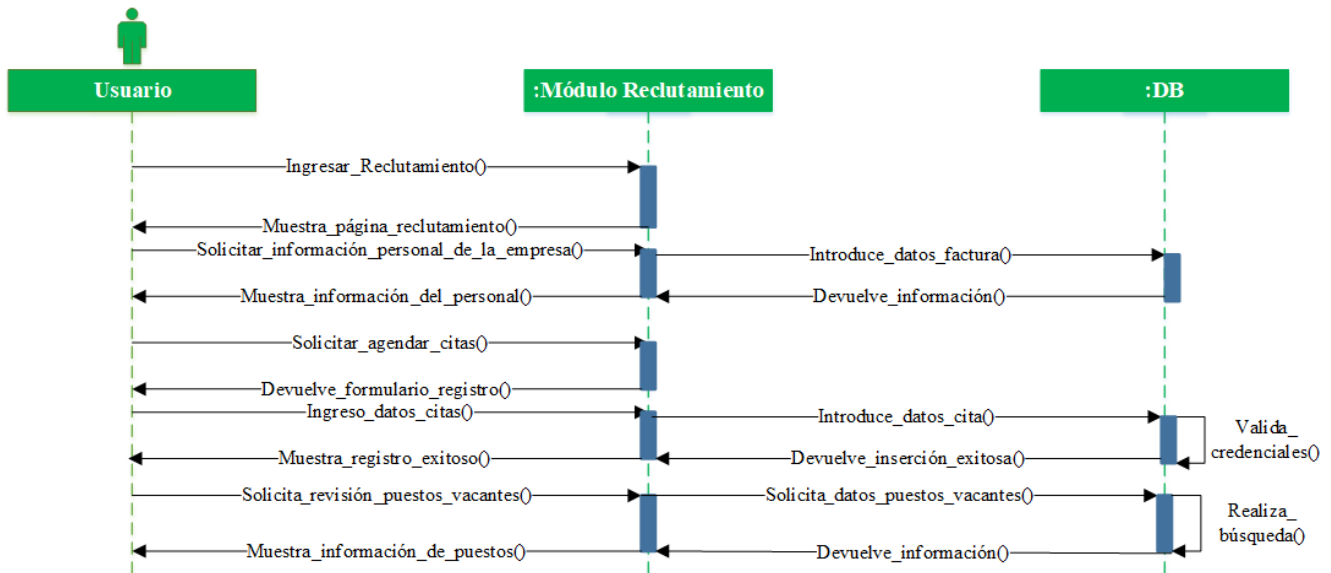


Diagrama de secuencias – Módulo gestión de flotas

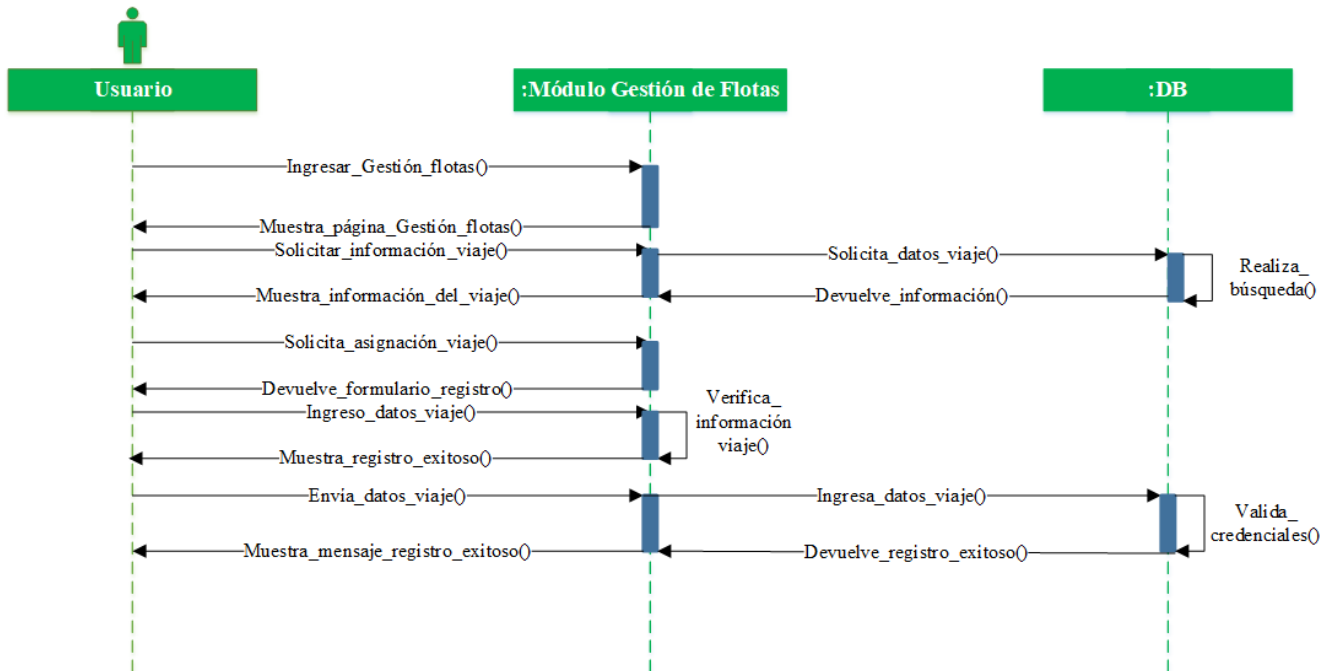


Diagrama de secuencias – Blog

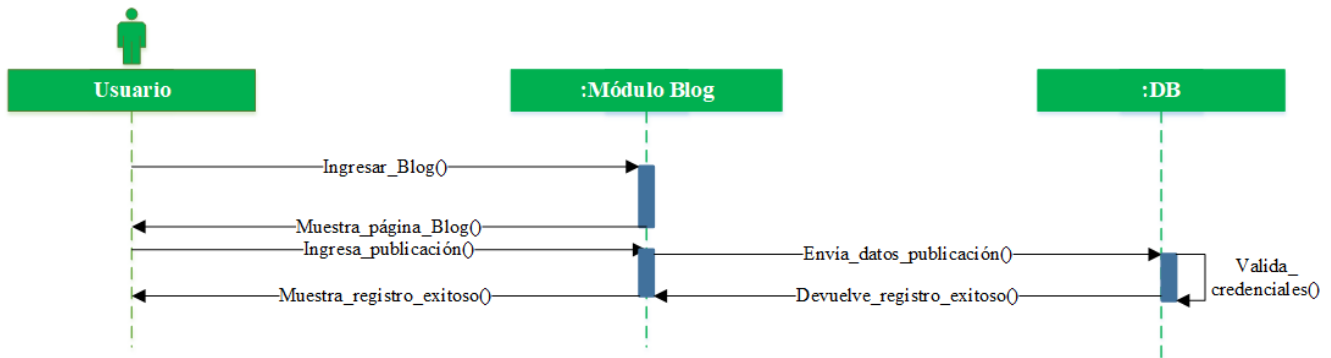


Diagrama de secuencias – Comentar

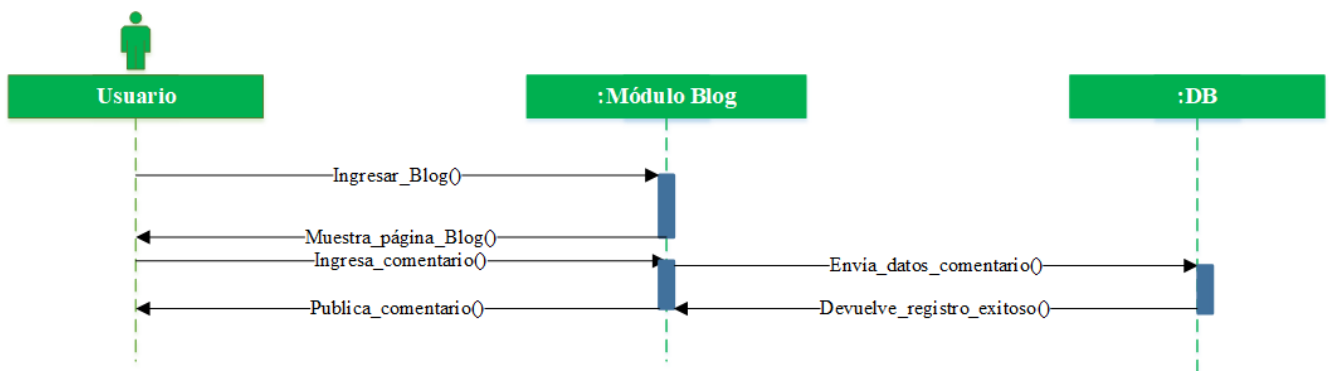


Diagrama de secuencias – Eventos

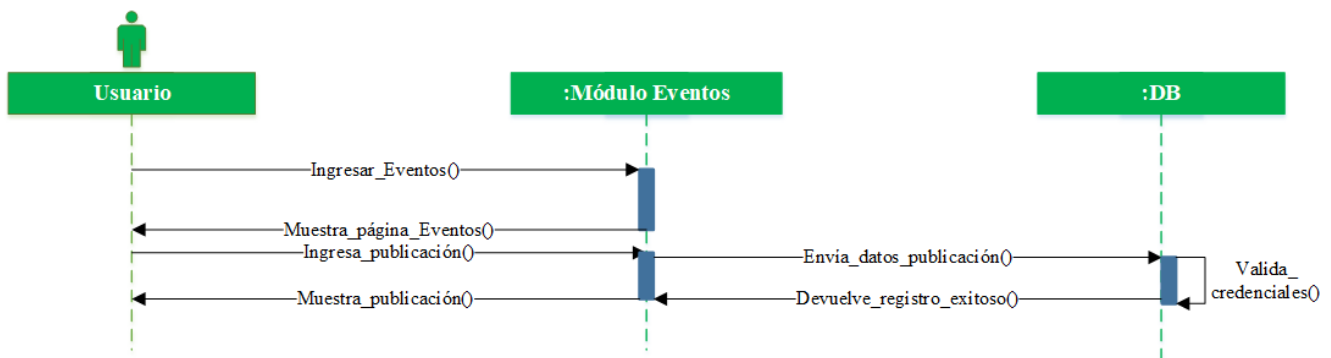
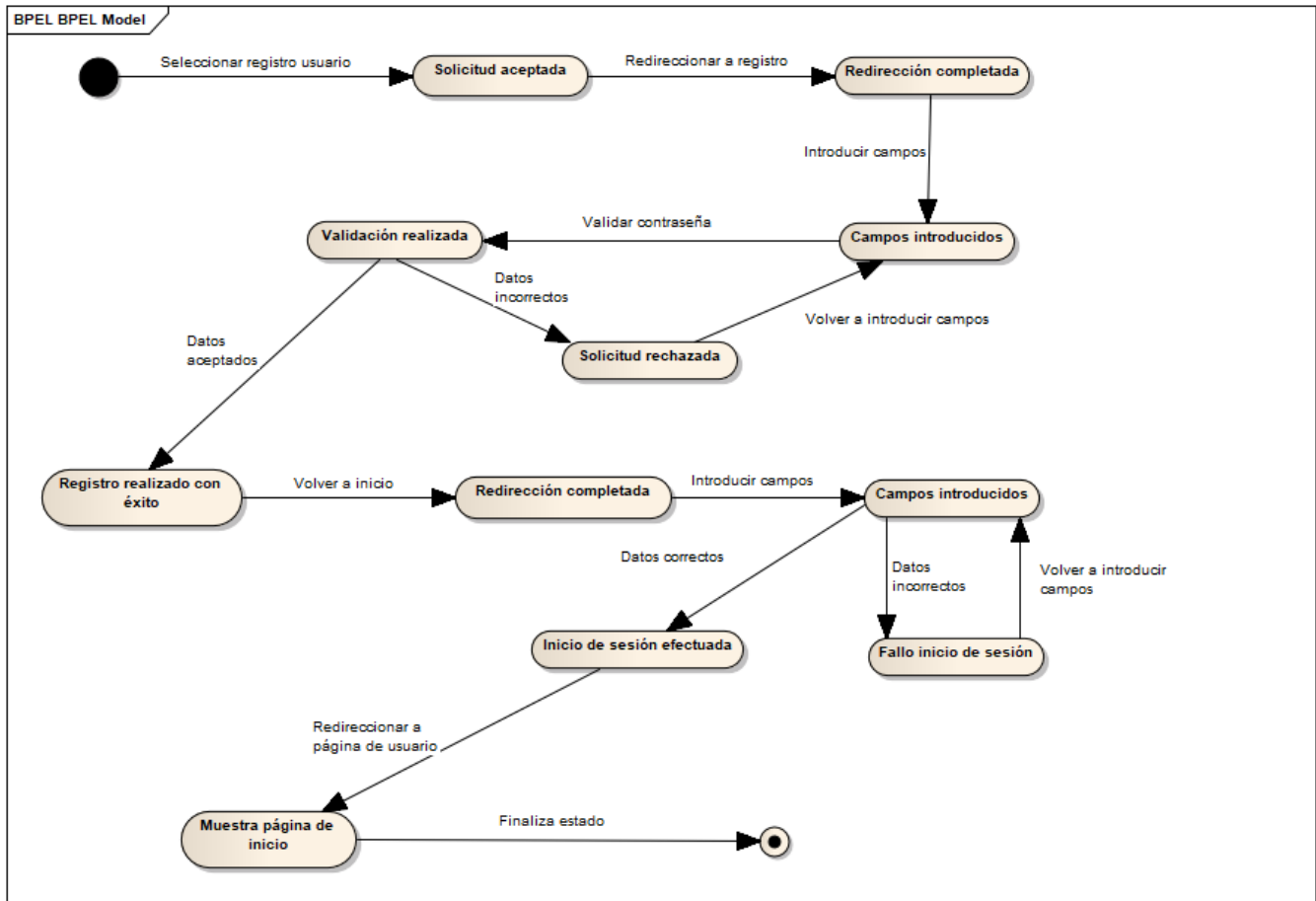
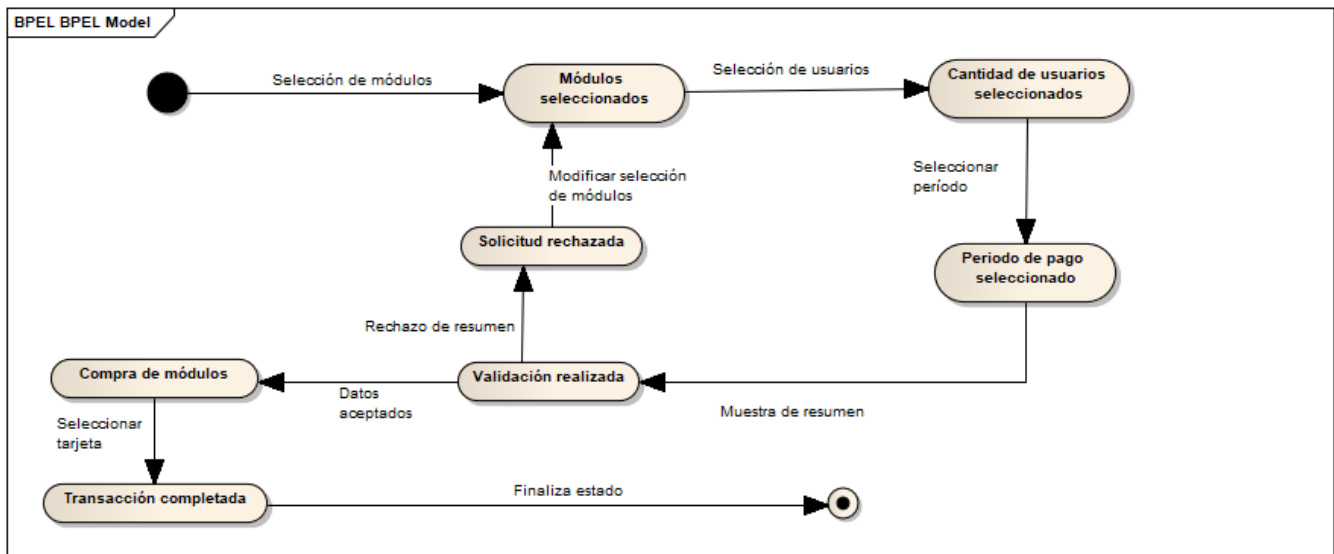


Diagrama de estados:

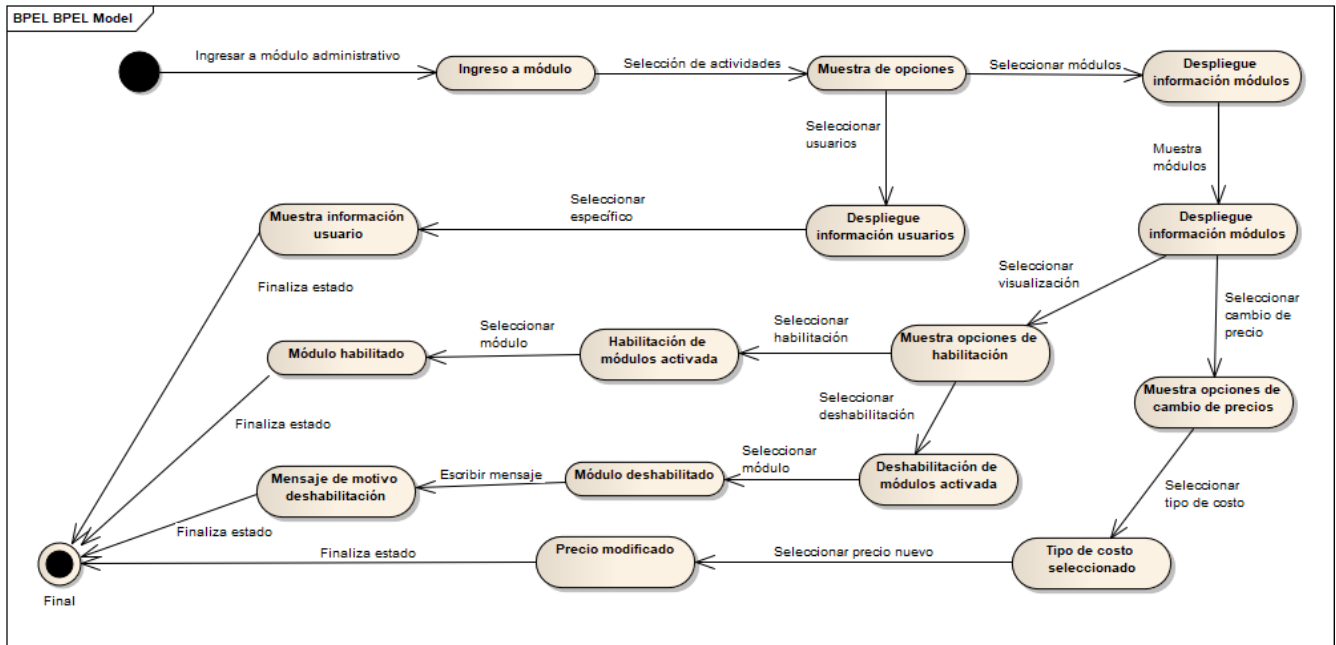
➤ Inicio sesión:



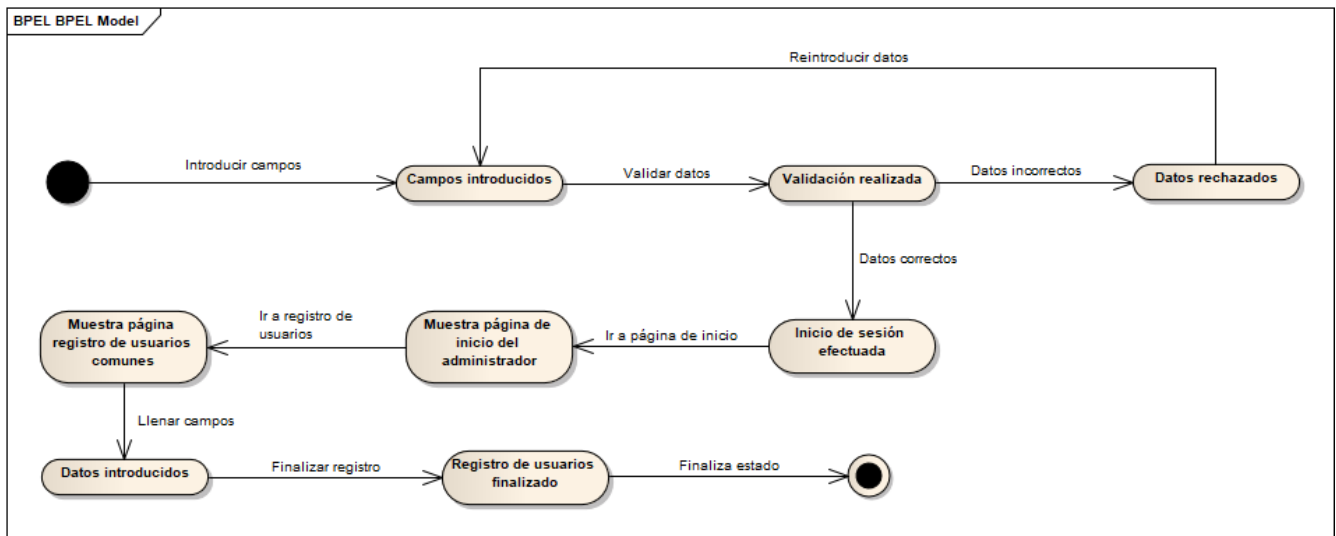
➤ Selección de módulos:



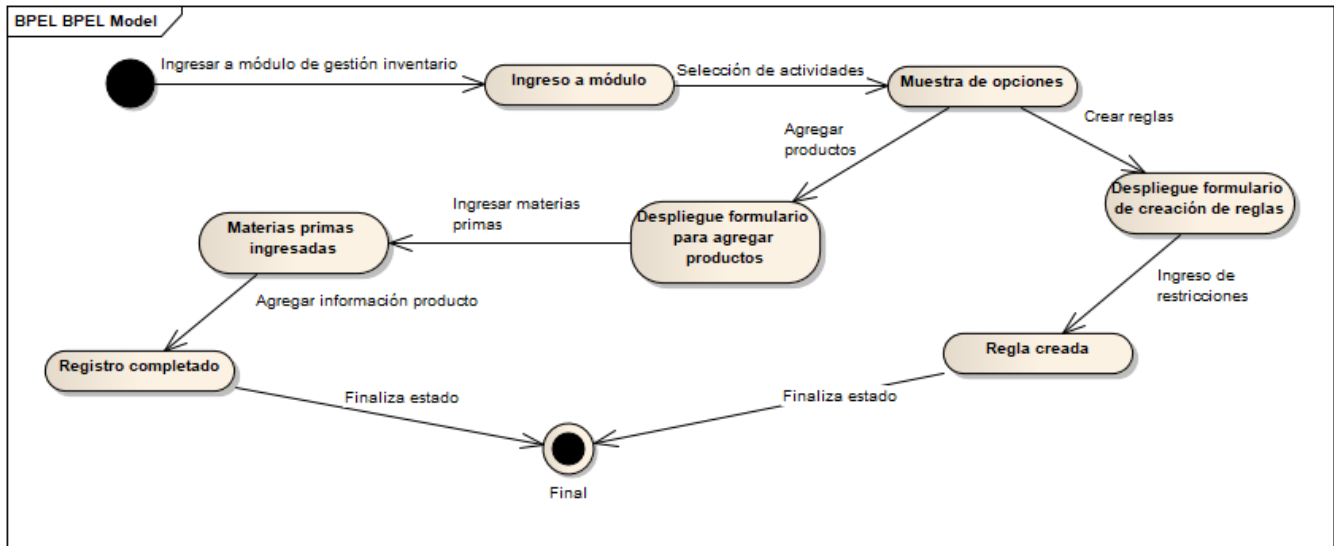
➤ Módulo administrativo:



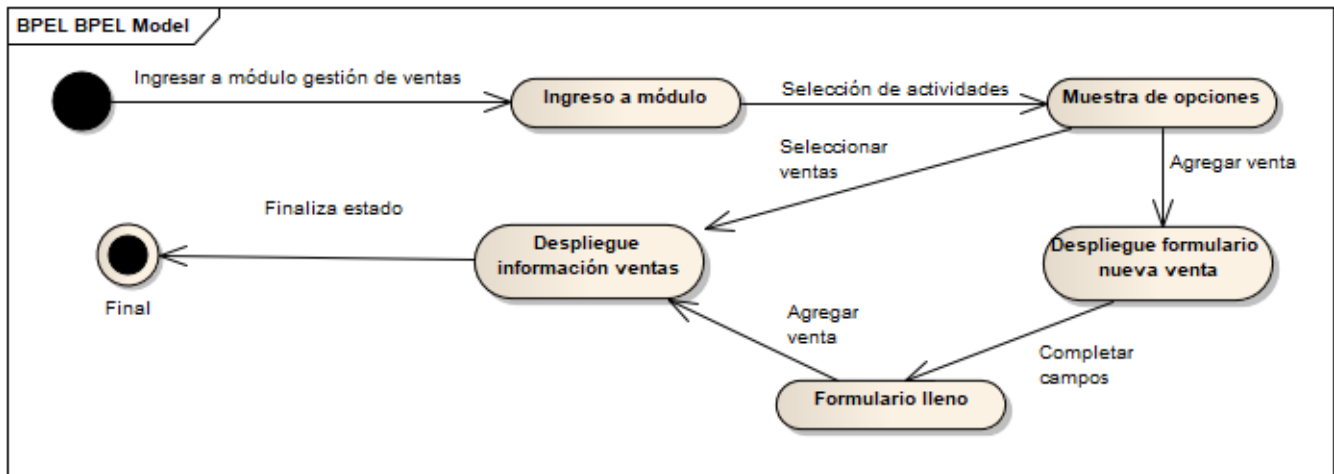
➤ Registro de usuarios:



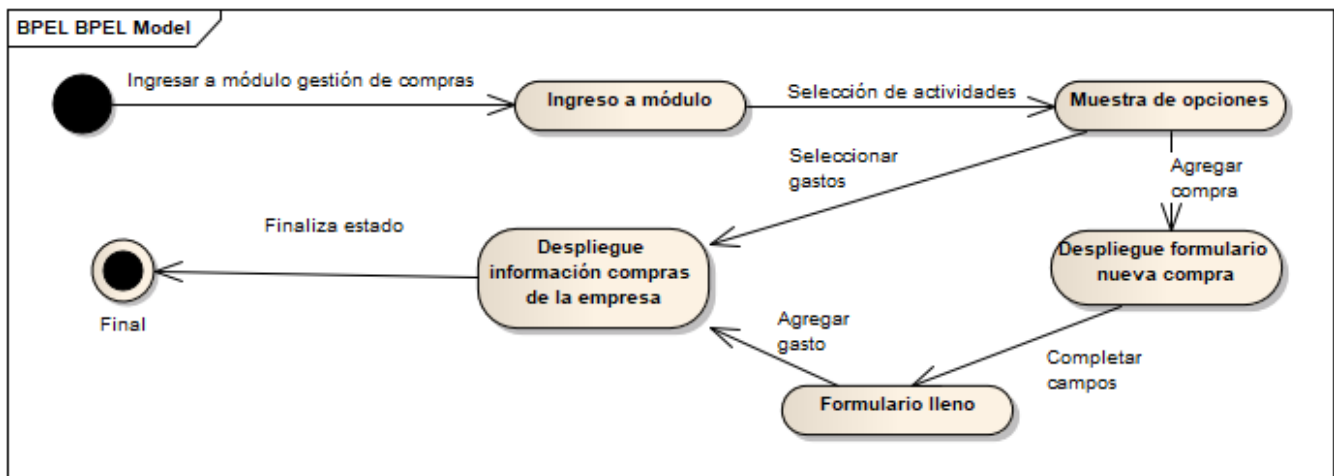
➤ Módulo de gestión de inventario:



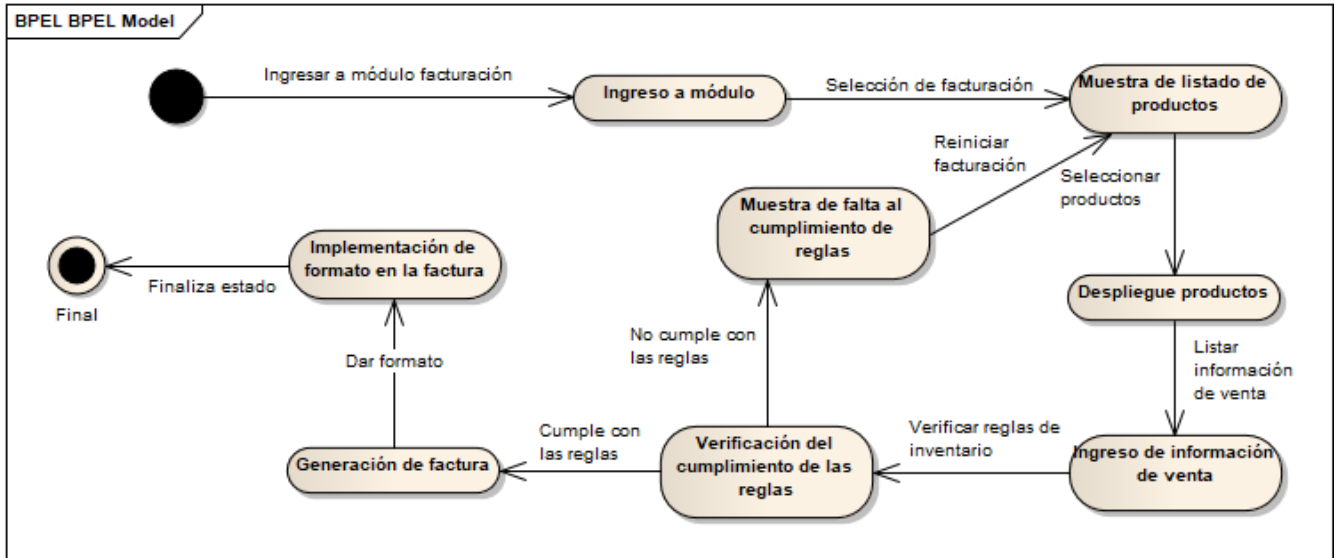
➤ Módulo gestión de ventas:



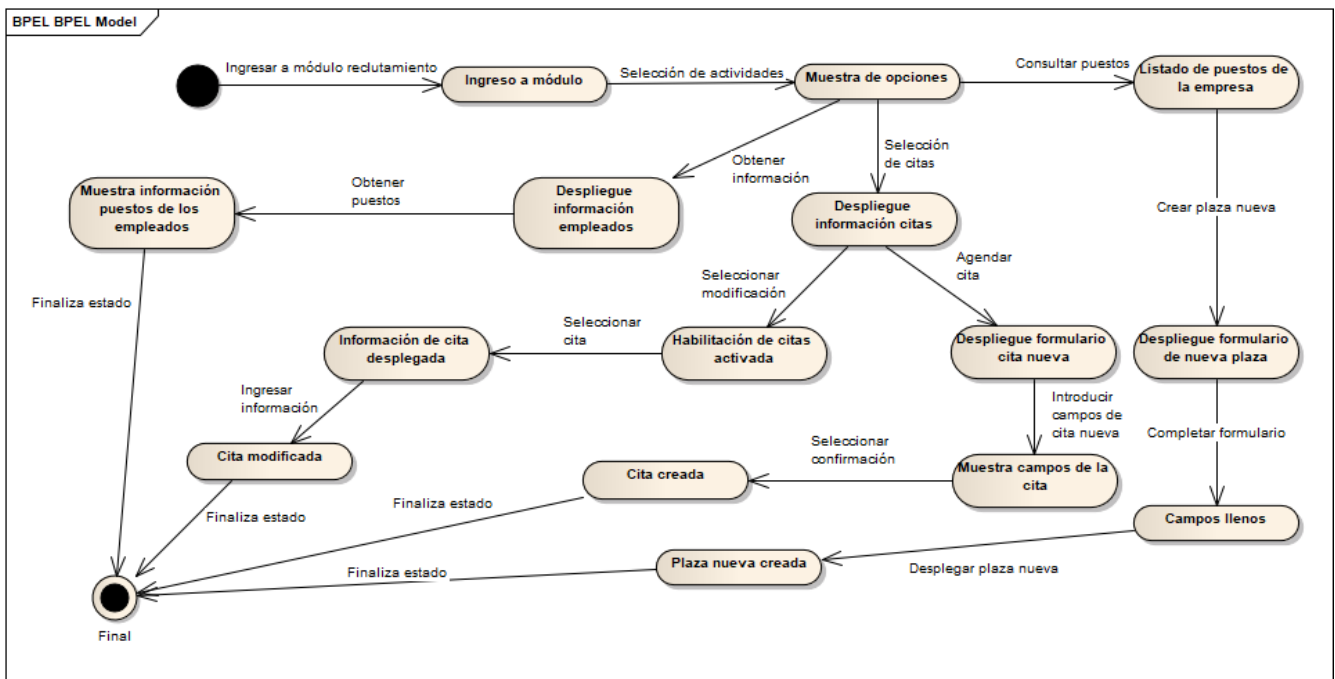
➤ Módulo de gestión de compras:



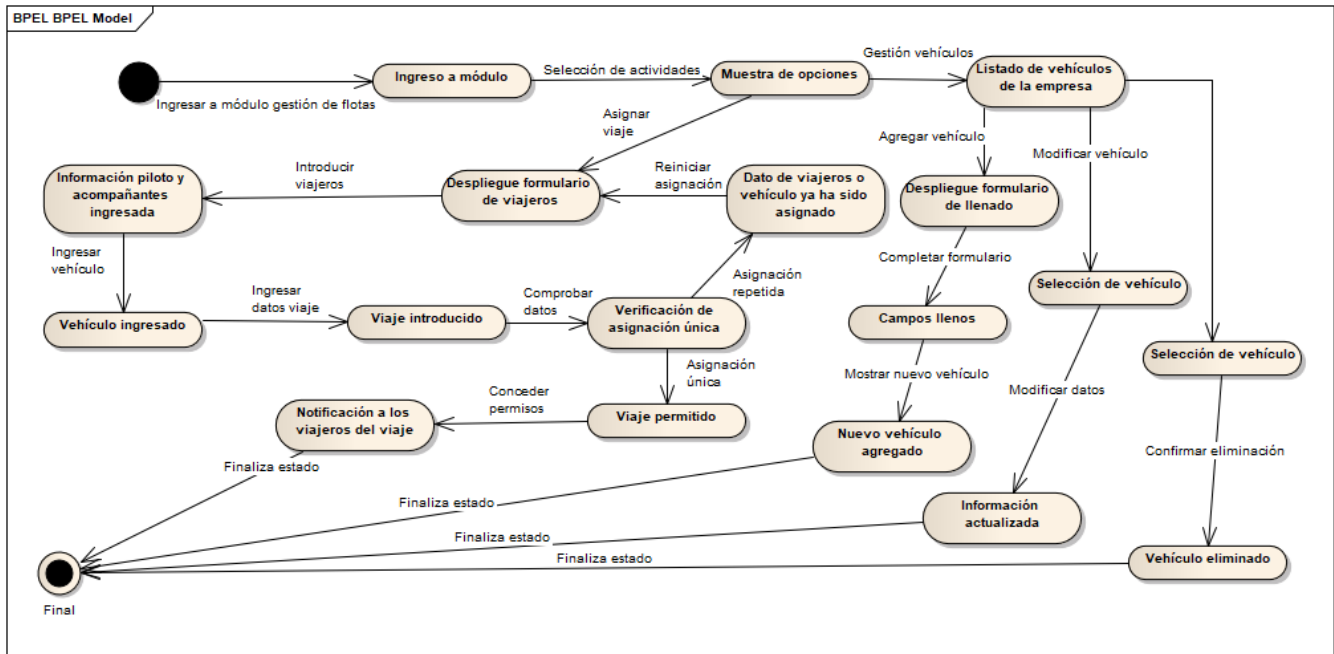
➤ Módulo facturación:



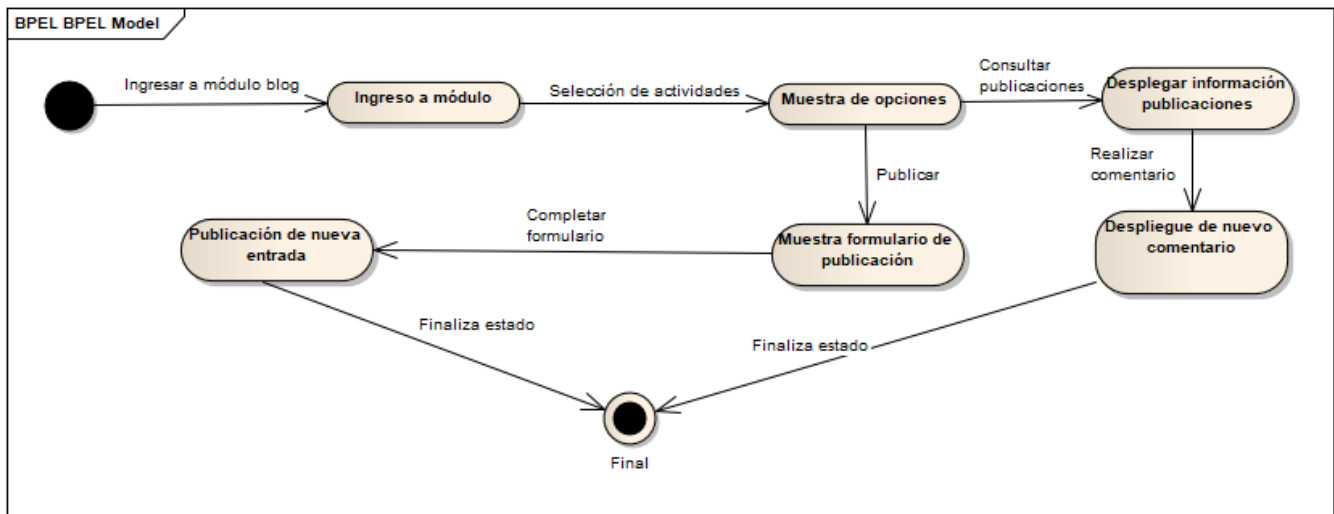
➤ **Módulo reclutamiento:**



➤ **Módulo de gestión de flota:**



➤ Módulo blog:



Arquitectura previa del sistema:

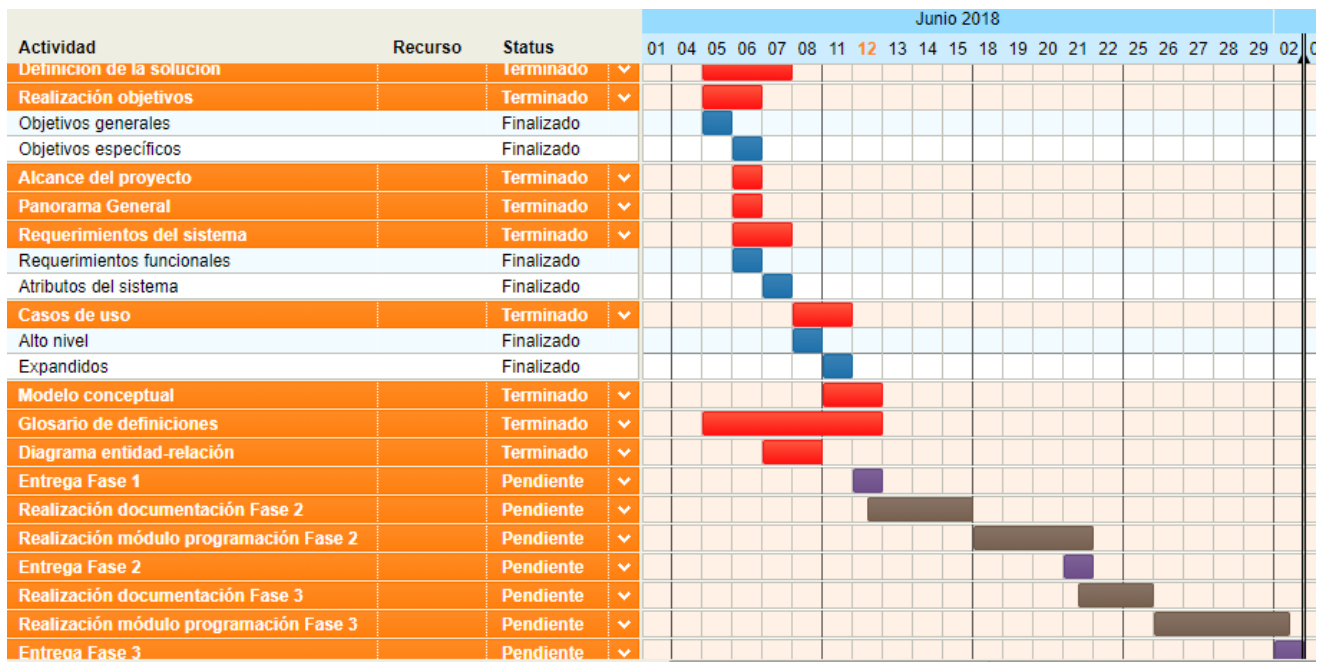
Según la arquitectura de tres capas, la cual está formada por la capa de presentación, la capa de lógica del negocio y la capa de base de datos, este proyecto hace uso de tres lenguajes de programación, cada uno con su respectiva capa.

Para la capa de presentación se hizo uso del lenguaje de programación C# por medio del entorno ASP.NET utilizando el IDE de Visual Studio 2015. Con el uso de web-service, para intercambio de datos entre aplicaciones, se pudo realizó la conexión entre la capa de presentación y la capa de la lógica del negocio.

Para la capa de la lógica del negocio; se utilizó el lenguaje de programación Java por medio del entorno NetBeans versión 8.2. En esta capa se creó la conexión a la base de datos y todas las consultas a la misma. Para poder hacer uso de las consultas se crearon servicios web con el objetivo de entrelazar dos lenguajes de programación.

Por último, la capa de base datos. Para esta capa se utilizó el sistema PostgreSQL con el entorno pgAdmin 4 versión 3.0. En esta capa se creó la base de datos, en la cual se registraría toda la información y se realizarían todas las consultas necesarias para la resolución del problema planteado en el proyecto.

Planificación de proyecto



Script:

```
CREATE TABLE Usuario_común (
    Cod_usuario_común SERIAL PRIMARY KEY NOT NULL,
    Credencial VARCHAR(50) NOT NULL
);
```

```
CREATE TABLE Administrador_sistema (  
    Cod_administrador_sistema SERIAL NOT NULL,  
    Credencial VARCHAR(50) NOT NULL,  
    PRIMARY KEY (Cod_administrador_sistema));
```

```
CREATE TABLE Contratante (  
    Cod_contratante SERIAL NOT NULL,  
    Nombre VARCHAR(50) NOT NULL,  
    DPI INT NOT NULL,  
    PRIMARY KEY (Cod_contratante));
```

```
CREATE TABLE Administrador_servicio (  
    Cod_administrador_servicio SERIAL NOT NULL,  
    Credencial VARCHAR(50) NOT NULL,  
    Cod_contratante INT NOT NULL,  
    PRIMARY KEY (Cod_administrador_servicio),  
    CONSTRAINT Cod_contratante  
        FOREIGN KEY (Cod_contratante)  
        REFERENCES Contratante (Cod_contratante)  
        ON DELETE CASCADE  
        ON UPDATE CASCADE);
```

```
CREATE TABLE Usuario (  
    Cod_usuario SERIAL NOT NULL,  
    Nombre VARCHAR(50) NOT NULL,  
    Apellido VARCHAR(50) NOT NULL,  
    Nickname VARCHAR(50) NOT NULL,  
    Contraseña VARCHAR(50) NOT NULL,  
    Cod_administrador_sistema INT NULL,  
    Cod_administrador_servicio INT NULL,  
    Cod_usuario_común INT NULL,  
    PRIMARY KEY (Cod_usuario),  
    CONSTRAINT Cod_usuario_común
```

```

FOREIGN KEY (Cod_usuario_común)
REFERENCES Usuario_común (Cod_usuario_común)
ON DELETE CASCADE
ON UPDATE CASCADE,
CONSTRAINT Cod_administrador_sistema
FOREIGN KEY (Cod_administrador_servicio)
REFERENCES Administrador_sistema (Cod_administrador_sistema)
ON DELETE CASCADE
ON UPDATE NO ACTION,
CONSTRAINT Cod_administrador_servicio
FOREIGN KEY (Cod_administrador_servicio)
REFERENCES Administrador_servicio (Cod_administrador_servicio)
ON DELETE CASCADE
ON UPDATE CASCADE);

```

```

CREATE TABLE Módulo (
Cod_módulo SERIAL NOT NULL,
Nombre VARCHAR(50) NOT NULL,
Demanda INT NOT NULL,
Costo FLOAT8 NOT NULL,
Habilitado INT NOT NULL,
Mensaje VARCHAR(100) NULL,
PRIMARY KEY (Cod_módulo));

```

```

CREATE TABLE Contratante_módulo (
Cod_contratante_módulo SERIAL NOT NULL,
Cod_contratante INT NOT NULL,
Cod_módulo INT NOT NULL,
Número_usuarios INT NOT NULL,
PRIMARY KEY (Cod_contratante_módulo),
CONSTRAINT Cod_contratante
FOREIGN KEY (Cod_contratante)

```



```
REFERENCES Contratante (Cod_contratante)
ON DELETE CASCADE
ON UPDATE CASCADE,
CONSTRAINT Cod_módulo
FOREIGN KEY (Cod_módulo)
REFERENCES Módulo (Cod_módulo)
ON DELETE CASCADE
ON UPDATE CASCADE);
```

```
CREATE TABLE Pago (
Cod_pago SERIAL NOT NULL,
Descripción VARCHAR(100) NOT NULL,
Total FLOAT8 NOT NULL,
Pago_realizado INT NOT NULL,
Cod_contratante INT NOT NULL,
PRIMARY KEY (Cod_pago),
CONSTRAINT Cod_contratante
FOREIGN KEY (Cod_contratante)
REFERENCES Contratante (Cod_contratante)
ON DELETE CASCADE
ON UPDATE CASCADE);
```

```
CREATE TABLE Tarjeta (
Cod_tarjeta SERIAL NOT NULL,
Tipo_tarjeta VARCHAR(45) NOT NULL,
Número_tarjeta INT NOT NULL,
Fecha_caducidad DATE NOT NULL,
Código_seguridad INT NOT NULL,
Cod_contratante INT NOT NULL,
PRIMARY KEY (Cod_tarjeta),
CONSTRAINT Cod_contratante
FOREIGN KEY (Cod_contratante)
```

```
REFERENCES Contratante (Cod_contratante)
ON DELETE CASCADE
ON UPDATE CASCADE);
```

```
CREATE TABLE Producto (
Cod_producto SERIAL NOT NULL,
Nombre VARCHAR(45) NOT NULL,
Cantidad INT NOT NULL,
Cod_materia_prima INT NULL,
PRIMARY KEY (Cod_producto),
CONSTRAINT Cod_materia_prima
FOREIGN KEY (Cod_materia_prima)
REFERENCES Producto (Cod_producto)
ON DELETE CASCADE
ON UPDATE CASCADE);
```

```
CREATE TABLE Cliente (
Cod_cliente SERIAL NOT NULL,
Nombre VARCHAR(45) NOT NULL,
PRIMARY KEY (Cod_cliente));
```

```
CREATE TABLE Venta (
Cod_venta SERIAL NOT NULL,
Importe_total FLOAT8 NOT NULL,
Fecha DATE NOT NULL,
Cod_cliente INT NOT NULL,
Cod_vendedor INT NOT NULL,
Descuento FLOAT8 NOT NULL,
PRIMARY KEY (Cod_venta),
CONSTRAINT Cod_cliente
FOREIGN KEY (Cod_cliente)
REFERENCES Cliente (Cod_cliente)
ON DELETE CASCADE
```

```
ON UPDATE CASCADE,  
CONSTRAINT Cod_vendedor  
FOREIGN KEY (Cod_vendedor)  
REFERENCES Usuario_común (Cod_usuario_común)  
ON DELETE CASCADE  
ON UPDATE CASCADE);
```

```
CREATE TABLE Venta_producto (  
Cod_venta_producto SERIAL NOT NULL,  
Cantidad_producto INT NOT NULL,  
Cod_venta INT NOT NULL,  
Cod_producto INT NOT NULL,  
PRIMARY KEY (Cod_venta_producto),  
CONSTRAINT Cod_venta  
FOREIGN KEY (Cod_venta)  
REFERENCES Venta (Cod_venta)  
ON DELETE CASCADE  
ON UPDATE CASCADE,  
CONSTRAINT Cod_producto  
FOREIGN KEY (Cod_producto)  
REFERENCES Producto (Cod_producto)  
ON DELETE CASCADE  
ON UPDATE CASCADE);
```

```
CREATE TABLE Proveedor (  
Cod_proveedor SERIAL NOT NULL,  
Nombre VARCHAR(45) NOT NULL,  
PRIMARY KEY (Cod_proveedor));
```

```
CREATE TABLE Compra (  
Cod_compra SERIAL NOT NULL,  
Fecha_pedido DATE NOT NULL,  
Fecha_recibido DATE NOT NULL,
```

Número_factura INT NOT NULL,
Importe_total FLOAT8 NOT NULL,
Cod_proveedor INT NOT NULL,
PRIMARY KEY (Cod_compra),
CONSTRAINT Cod_proveedor
FOREIGN KEY (Cod_proveedor)
REFERENCES Proveedor (Cod_proveedor)
ON DELETE CASCADE
ON UPDATE CASCADE);

CREATE TABLE Factura (
Cod_factura SERIAL NOT NULL,
Correlativo INT NOT NULL,
Fecha DATE NOT NULL,
Hora TIME NOT NULL,
Importe_total FLOAT8 NOT NULL,
Impuesto FLOAT8 NOT NULL,
Cod_cliente INT NOT NULL,
Cod_vendedor INT NOT NULL,
PRIMARY KEY (Cod_factura),
CONSTRAINT Cod_cliente
FOREIGN KEY (Cod_cliente)
REFERENCES Cliente (Cod_cliente)
ON DELETE CASCADE
ON UPDATE CASCADE,
CONSTRAINT Cod_vendedor
FOREIGN KEY (Cod_vendedor)
REFERENCES Usuario_común (Cod_usuario_común)
ON DELETE CASCADE
ON UPDATE CASCADE);

CREATE TABLE Factura_producto (
Cod_factura_producto SERIAL NOT NULL,

```
Cod_factura INT NOT NULL,  
Cod_producto INT NOT NULL,  
PRIMARY KEY (Cod_factura_producto),  
CONSTRAINT Cod_factura  
    FOREIGN KEY (Cod_factura)  
    REFERENCES Factura (Cod_factura)  
    ON DELETE CASCADE  
    ON UPDATE CASCADE,  
CONSTRAINT Cod_producto  
    FOREIGN KEY (Cod_producto)  
    REFERENCES Producto (Cod_producto)  
    ON DELETE CASCADE  
    ON UPDATE CASCADE);
```

```
CREATE TABLE Compra_producto (  
    Cod_compra_producto SERIAL NOT NULL,  
    Cod_compra INT NOT NULL,  
    Cod_producto INT NOT NULL,  
    PRIMARY KEY (Cod_compra_producto),  
    CONSTRAINT Cod_compra  
        FOREIGN KEY (Cod_compra)  
        REFERENCES Compra (Cod_compra)  
        ON DELETE CASCADE  
        ON UPDATE CASCADE,  
    CONSTRAINT Cod_producto  
        FOREIGN KEY (Cod_producto)  
        REFERENCES Producto (Cod_producto)  
        ON DELETE CASCADE  
        ON UPDATE CASCADE);
```

```
CREATE TABLE Puesto (  
    Cod_puesto SERIAL NOT NULL,  
    Ocupación CHAR(1) NOT NULL,
```

Nombre VARCHAR(45) NOT NULL,
PRIMARY KEY (Cod_puesto));

```
CREATE TABLE Personal (  
    Cod_personal SERIAL NOT NULL,  
    Nombre VARCHAR(45) NOT NULL,  
    Apellido VARCHAR(45) NOT NULL,  
    DPI INT NOT NULL,  
    NIT VARCHAR(45) NOT NULL,  
    Cod_puesto INT NOT NULL,  
    Cod_usuario_común INT NOT NULL,  
    PRIMARY KEY (Cod_personal),  
    CONSTRAINT Cod_usuario_común  
        FOREIGN KEY (Cod_usuario_común)  
        REFERENCES Usuario_común (Cod_usuario_común)  
        ON DELETE CASCADE  
        ON UPDATE CASCADE,  
    CONSTRAINT Cod_puesto  
        FOREIGN KEY (Cod_puesto)  
        REFERENCES Puesto (Cod_puesto)  
        ON DELETE CASCADE  
        ON UPDATE CASCADE);
```

```
CREATE TABLE Viaje (  
    Cod_viaje SERIAL NOT NULL,  
    Fecha_salida DATE NOT NULL,  
    Fecha_llegada DATE NOT NULL,  
    Distancia FLOAT8 NOT NULL,  
    PRIMARY KEY (Cod_viaje));
```

```
CREATE TABLE Piloto (  
    Cod_piloto SERIAL NOT NULL,  
    Nombre VARCHAR(45) NOT NULL,
```

```
Apellido VARCHAR(45) NOT NULL,  
DPI INT NOT NULL,  
Cod_viaje INT NOT NULL,  
Cod_usuario_común INT NOT NULL,  
PRIMARY KEY (Cod_piloto),  
CONSTRAINT Cod_viaje  
FOREIGN KEY (Cod_viaje)  
REFERENCES Viaje (Cod_viaje)  
ON DELETE CASCADE  
ON UPDATE CASCADE,  
CONSTRAINT Cod_usuario_común  
FOREIGN KEY (Cod_usuario_común)  
REFERENCES Usuario_común (Cod_usuario_común)  
ON DELETE CASCADE  
ON UPDATE CASCADE);
```

```
CREATE TABLE Vehículo (  
Cod_vehículo SERIAL NOT NULL,  
Marca VARCHAR(45) NOT NULL,  
Placa VARCHAR(45) NOT NULL,  
Modelo VARCHAR(45) NOT NULL,  
Cod_piloto INT NOT NULL,  
PRIMARY KEY (Cod_vehículo),  
CONSTRAINT Cod_piloto  
FOREIGN KEY (Cod_piloto)  
REFERENCES Piloto (Cod_piloto)  
ON DELETE CASCADE  
ON UPDATE CASCADE);
```

```
CREATE TABLE Acompañante (  
Cod_acompañante SERIAL NOT NULL,  
Nombre VARCHAR(45) NOT NULL,  
Apellido VARCHAR(45) NOT NULL,
```

```
DPI INT NOT NULL,  
Cod_viaje INT NOT NULL,  
Cod_usuario_común INT NOT NULL,  
PRIMARY KEY (Cod_acompañante),  
CONSTRAINT Cod_viaje  
    FOREIGN KEY (Cod_viaje)  
    REFERENCES Viaje (Cod_viaje)  
    ON DELETE CASCADE  
    ON UPDATE CASCADE,  
CONSTRAINT Cod_usuario_común  
    FOREIGN KEY (Cod_usuario_común)  
    REFERENCES Usuario_común (Cod_usuario_común)  
    ON DELETE CASCADE  
    ON UPDATE CASCADE);
```

```
CREATE TABLE Viaje_producto (  
    Cod_viaje_producto SERIAL NOT NULL,  
    Cod_viaje INT NOT NULL,  
    Cod_producto INT NOT NULL,  
    PRIMARY KEY (Cod_viaje_producto),  
    CONSTRAINT Cod_viaje  
        FOREIGN KEY (Cod_viaje)  
        REFERENCES Viaje (Cod_viaje)  
        ON DELETE CASCADE  
        ON UPDATE CASCADE,  
    CONSTRAINT Cod_producto  
        FOREIGN KEY (Cod_producto)  
        REFERENCES Producto (Cod_producto)  
        ON DELETE CASCADE  
        ON UPDATE CASCADE);
```

```
CREATE TABLE Noticia (  
    Cod_noticia SERIAL NOT NULL,
```


Título VARCHAR(45) NOT NULL,
Contenido VARCHAR(45) NOT NULL,
Cod_administrador_servicio INT NOT NULL,
PRIMARY KEY (Cod_noticia),
CONSTRAINT Cod_administrador_servicio
FOREIGN KEY (Cod_administrador_servicio)
REFERENCES Administrador_servicio (Cod_administrador_servicio)
ON DELETE CASCADE
ON UPDATE CASCADE);

CREATE TABLE Comentario (
Cod_comentario SERIAL NOT NULL,
Contenido VARCHAR(45) NOT NULL,
Cod_usuario_común INT NOT NULL,
Cod_noticia INT NOT NULL,
PRIMARY KEY (Cod_comentario),
CONSTRAINT Cod_usuario_común
FOREIGN KEY (Cod_usuario_común)
REFERENCES Usuario_común (Cod_usuario_común)
ON DELETE CASCADE
ON UPDATE CASCADE,
CONSTRAINT Cod_noticia
FOREIGN KEY (Cod_noticia)
REFERENCES Noticia (Cod_noticia)
ON DELETE CASCADE
ON UPDATE CASCADE);

CREATE TABLE Evento (
Cod_evento SERIAL NOT NULL,
Contenido VARCHAR(45) NOT NULL,
Estado VARCHAR(45) NOT NULL,
Fecha VARCHAR(45) NOT NULL,
Cod_administrador_servicio INT NOT NULL,

```
PRIMARY KEY (Cod_evento),  
CONSTRAINT Cod_administrador_servicio  
FOREIGN KEY (Cod_administrador_servicio)  
REFERENCES Administrador_servicio (Cod_administrador_servicio)  
ON DELETE CASCADE  
ON UPDATE CASCADE);
```

```
CREATE TABLE Oportunidad (  
Cod_oportunidad SERIAL NOT NULL,  
Título VARCHAR(45) NOT NULL,  
Descripción VARCHAR(45) NOT NULL,  
Monto FLOAT8 NOT NULL,  
Prioridad INT NOT NULL,  
Cod_cliente INT NOT NULL,  
PRIMARY KEY (Cod_oportunidad),  
CONSTRAINT Cod_cliente  
FOREIGN KEY (Cod_cliente)  
REFERENCES Cliente (Cod_cliente)  
ON DELETE CASCADE  
ON UPDATE CASCADE);
```

```
CREATE TABLE Regla (  
Cod_regla SERIAL NOT NULL,  
Cant_mínima INT NOT NULL,  
Cant_max INT NOT NULL,  
Cod_producto INT NOT NULL,  
PRIMARY KEY (Cod_regla),  
CONSTRAINT Cod_producto  
FOREIGN KEY (Cod_producto)  
REFERENCES Producto (Cod_producto)  
ON DELETE CASCADE  
ON UPDATE CASCADE);
```

```
CREATE TABLE Cita (  
    Cod_cita SERIAL NOT NULL,  
    Fecha DATE NOT NULL,  
    Hora TIME NOT NULL,  
    Nombre VARCHAR(45) NOT NULL,  
    Cod_puesto INT NOT NULL,  
    PRIMARY KEY (Cod_cita),  
    CONSTRAINT Cod_puesto  
        FOREIGN KEY (Cod_puesto)  
        REFERENCES Puesto (Cod_puesto)  
    ON DELETE CASCADE  
    ON UPDATE CASCADE);
```