Capítulo I

El Proyecto

1.1 Título del proyecto

Mejoramiento en el manejo de la información y tiempo de atención al cliente mediante la gestión de reservas de pedidos en línea en “My Burguer”.

1.2 Área de trabajo

Desarrollo de Sistemas de Información.

1.3 Responsables del proyecto

Carrera de Ingeniería Informática – Taller III – Grupo 2

Marcia Milenka Andrade Llanos

1.4 Entidades asociadas

- U.A.J.M.S. “Universidad Autónoma Juan Misael Saracho”
- Facultad de Ciencias y Tecnología.
- “My Burguer”

1.5 Duración del proyecto

La duración del proyecto es de 10 meses aproximadamente

1.6 Grupo responsable del proyecto

Marcia Milenka Andrade Llanos – Estudiante de la Carrera de Ingeniería Informática.

2. Personal vinculado al proyecto

2.1 Director responsable del proyecto

Apellido Paterno	**Apellido Materno**	**Nombre**	**C.I.**
Andrade	Llanos	Marcia Milenka	7177285 Tja.
Profesión	**Carrera o Unidad**	**Facultad:**	
Estudiante	Ingeniería Informática	Ciencias y Tecnología	
Telf. Oficina	**Celular**	**Correo electrónico**	**Firma**
	79254690	marcia.andrade.llanos@gmail.com	

Tabla 1. Director responsable del proyecto

2.2 Participantes equipo de trabajo

Categoría	**Nombres y Apellidos**	**Profesión**	**C.I.**	**Firma**
TUTOR				
ASESOR	Luis Delfín Briceño Gordy	Gerente “My Burguer”	5049145	

2.3 Equipo de trabajo de: Empresas/Instituciones/Organizaciones Participantes/Cooperantes

Nombre: Hamburguesería “My Burguer”			
Dirección: Avenida Font, B/ Juan XXIII		**Telef. Oficina:**	
Nombre y Apellidos	**Cargo**	**C.I.**	**Firma**
Luis Delfín Briceño Gordy	Gerente “My Burguer”	5049145	

3. Descripción del proyecto

3.1 Resumen ejecutivo del proyecto

A medida que la ciudad se moderniza, la población experimenta transformaciones en los estilos de vida. El momento de la comida rápida se ha establecido como una necesidad moderna de una manera silenciosa pero fuerte en la ciudad. La urgencia de realizar más tareas durante el día lleva a la persona común a minimizar los tiempos, uno de ellos es el tiempo de la alimentación.

Actualmente en nuestro entorno, factores como la globalización y la aceleración del desarrollo tecnológico, están cambiando la forma de hacer negocios. Las empresas se están viendo obligadas a reestructurar sus procesos internos, con el fin de mejorar sus tiempos de respuesta y ser más eficientes, para lo cual están utilizando herramientas tecnológicas de información que mejoren o incrementen su productividad.

El proyecto se centrará en mejorar el manejo de la información en cuanto se refiere a la atención al cliente y gestión de reservas en línea de comida rápida, sector donde los sistemas informáticos se están empezando a utilizar cada vez con mayor intensidad, teniendo una gran acogida en el mercado actual.

“My Burguer” es una empresa dedicada a la venta de comida rápida en la ciudad de Tarija, actualmente cuenta con varias áreas de trabajo y de organización, pero el proyecto se centrará en optimizar el manejo de la información y tiempo de atención al cliente.

Toda empresa comienza de manera básica, con la organización de su información de forma manual, siendo muchas veces un método eficiente pero con el tiempo estas funciones se comienzan a multiplicar y es necesario dar un gran paso y migrar hacia la nueva tecnología, adquiriendo sistemas de información más modernos y capaces de procesar grandes volúmenes de información, cubriendo así las nuevas necesidades de la empresa. Actualmente la forma de administrar la información en el sector mencionado no satisface las necesidades y los requerimientos de la empresa, dado

que el mercado local tiene un elevado índice de crecimiento, con necesidades alimenticias.

Dado que actualmente el volumen de información se maneja de forma manual, en la mayoría de los casos da como resultado retrasos al momento de realizar la planificación efectiva de la empresa, resultando en un perjuicio al desempeño de algunos empleados.

“My Burguer” suele incurrir en problemas de organización de los pedidos realizados por clientes debido a un registro manual inadecuado sobre la información de los productos y un flujo inconveniente de los pedidos, lo que provoca que existan quejas y demoras en la atención al cliente; de manera similar este método de almacenamiento de la información afecta a la organización en el armado de los menús, por esto es que pocas personas conocen los productos que ofrece, reduciendo significativamente el abanico de gustos y detectar las nuevas necesidades del mercado y por ende los ingresos; todos estos problemas perjudican de manera significativa el manejo de información provocando retrasos en los procesos y funciones que afectan al rendimiento empresarial y la atención a los clientes del mercado de “My Burguer”.

Toda la información se encuentra registrada en hojas manuscritas, la cuál podría ser almacenada en un sistema informático capaz de optimizar espacio físico, acceso a los datos vía internet, publicitar sus productos en línea, facilitar horarios de reservas en línea, reducir los tiempos de atención a los clientes internos y externos, contribuyendo de esta manera a la fluidez en el crecimiento empresarial, ofreciendo una presentación visual de sus productos por la web, con la facilidad de reservar pedidos dentro de un horario determinado por la empresa; además una fuente de información rápida y oportuna con la finalidad de coadyuvar la comunicación interna y la manipulación de los datos que se manejan de manera interna, obtener e implementar reportes, accesos vía internet o intranet y desde cualquier plataforma, ya sea, Windows, Linux y otros, de esta forma reducir tiempos en los procesos de atención y satisfacer las necesidades tanto del propietario como del personal de la

empresa que necesiten acceder de manera remota y satisfacer los gustos de los clientes mediante presentaciones visuales de los productos que se ofrecen.

La capacitación al personal sobre el manejo de la información basándose en herramientas de soporte de la empresa, favorecerá al manejo y administración de recursos informáticos debido a que tendrán una comprensión más clara sobre las reglas y funciones que el sistema exige, logrando de esta manera que la empresa crezca tanto vertical como horizontalmente.

El lanzamiento publicitario del sistema en las redes sociales favorecerá al crecimiento del mercado de “My Burguer”.

Caso de estudio

Actualmente la empresa “My Burguer” no cuenta con sistema informático que contribuya al manejo de información y seguimiento de reservas.

Actividades previstas para los integrantes del equipo de investigación

Responsable	**Actividades**
Director Marcia Milenka Andrade Llanos	Conducción del proyecto informático desde su concepción original hasta el lanzamiento del producto final. Tareas: • Definir el proyecto y evaluar sus necesidades. • Redactar las especificaciones del proyecto. • Calcular el costo del proyecto. • Realizar un seguimiento e informes del progreso del proyecto, en términos de calidad, costo y plazos de entrega.
Tutor	• Orientar al estudiante sobre las pautas a seguir para desarrollar el trabajo de investigación de grado en base a los reglamentos establecidos. • Establecer el tema de investigación en base a las Líneas de Investigación existentes. • Guiar en la forma de obtención y registro de la revisión de la información bibliográfica: texto, revistas científicas, acceso

	electrónico a base de datos en bibliotecas virtuales, etc. • Asesorar oportunamente las consultas. • Velar por el cumplimiento de las actividades y cronograma indicados en el proyecto • Exigir la presentación del avance del proyecto. • Revisión y visto bueno de la documentación presentada a la universidad.
Investigador 1 Marcia Milenka Andrade Llanos	• Recopilación de datos de la empresa "My Burguer"
Asesor Luis Delfín Briceño Gordy	• Facilitar la información necesaria para el análisis de requerimientos. • Facilitar acceso a la documentación autorizada. • Asistencia en cuanto a las necesidades informáticas de la Empresa. • Revisión y visto bueno sobre el avance del proyecto. • Asistencia en configuraciones y demás correspondientes a los medios de almacenamiento informático de la empresa. • Experto en el manejo de documentos y requisitos empresariales. • Experto en las necesidades y requerimientos de reportes. • Orientar sobre los fallos actuales sobre el manejo de la información. • Experto en el manejo de la documentación interna.

Tabla 2. Actividad prevista para los integrantes

3.2 Descripción, fundamentación y justificación del proyecto (qué y porqué)

El mundo actual es sumamente complejo y exigente; por esto, enmarcadas en los márgenes de exigencia, calidad y atención de las nuevas generaciones, las empresas competitivas de hoy en día han comprendido y aprendido que el éxito hay que buscarlo en el manejo de una exitosa relación con los clientes, más allá de la rentabilidad por producto o líneas de producto. El éxito, y por ende la

rentabilidad empresarial, vienen del conocimiento del cliente, de reconocer su valor actual y potencial y de saber qué productos le gustan, de escuchar sus quejas y sugerencias y saber cómo utilizarlas en beneficio de la compañía, en definitiva se trata de poder conocer y predecir el comportamiento actual y futuro de los clientes para garantizar su completa y plena satisfacción para finalmente conseguir su lealtad hacia la empresa.

En este contexto, las nuevas tecnologías son el vehículo que le permite a una empresa competir más eficazmente. Tener la información disponible, allí en donde se la necesite y en el momento en el que se la necesite, se ha convertido en una variable estratégica para la competitividad. Esta variable afecta, entre otros parámetros, a la optimización de recursos, la satisfacción de los clientes, la optimización del tiempo, y la capacidad de adaptación al cambio, en este sentido es que las empresas modernas adoptan nuevas tecnologías y nuevos sistemas que les permitan mejorar sus estrategias de mercado.

Estos sistemas se explican perfectamente cuando a desempeño organizacional se refiere, siendo el fin, cumplir los objetivos que se propone cada organización, convirtiéndose el sistema de información, en su medio principal para resolver tareas administrativas, funcionales y de producción, constituyendo la información en la fuente de partida de todos los procesos dentro de las entidades, que tienen proyecciones de crecimiento cualitativo y cuantitativamente basándose en resultados oportunos a través de sistemas de información.

El propósito del proyecto “Optimizar el manejo de información y tiempo de atención al cliente mediante la gestión de reservas de pedidos en ‘My Burguer’” tiene como finalidad contribuir al mejoramiento del mercado. Dentro la organización se busca mejorar y/o facilitar las diversas tareas o funciones que desempeñan el personal de la empresa como ser seguimiento de reservas de los pedidos de clientes, armar el menú de comida, publicitar los productos que ofrece; reportes diarios, semanales, mensuales y anuales, según sea la

necesidad. El proyecto persigue de manera general brindar una herramienta que pueda optimizar los procesos mencionados anteriormente.

Los problemas por los que pasa la empresa “My Burguer” actualmente son:

- **Registro inadecuado sobre la información de los productos.-** Todo el registro de los productos es de forma manual.
- **Flujo inadecuado de los pedidos.-** La información solicitada por los empleados durante el proceso de atención al cliente desde la reserva, preparación y entrega final del pedido se registra de forma manual en hojas sueltas.
- **Inadecuada organización de los pedidos realizados por los clientes.-** Los pedidos suelen incurrir en demoras de tiempo, equivocaciones en los productos seleccionados, perdidas de las ordenes de clientes.
- **Organización ineficiente en el armado del menú.-** El menú es armado en hojas manuscritas lo que provocan demoras debido a que la información de los productos es difícilmente asequible.
- **Escasa información sobre productos ofertados.-** La población desconoce las características de los productos que la empresa ofrece.
- **Realizar reportes conlleva tiempo.-** Dado que la generación de reportes son estáticos e incompletos, se pierde demasiado tiempo en el armado de algunos que son requeridos por la gerencia.
- **Demoras frecuentes en atención al cliente.-** Provocado por el procesamiento lento de la información.
- **Marketing insuficiente.-** Imposibilidad de inserción de Tecnologías de la Información al marketing empresarial.

En consecuencia a lo mencionado, los métodos tradicionales de papeleo supone gastos importantes en horas hombre para su distribución.

Lo que se pretende es que el sistema pueda ser utilizado vía internet para recopilar, almacenar, procesar los datos transformándolos en información valiosa facilitando la comunicación y mejorando exponencialmente todas las áreas afectadas en la estructura organizacional dentro del sector tomado en cuenta en el proyecto que es el manejo de la información y gestión de reservas en línea.

Con el objeto de solucionar los tiempos de atención al cliente y obtención de la información, surge la necesidad de este proyecto incorporando las siguientes soluciones para su desarrollo:

- **Registro computarizado de productos en línea.-** Los productos ofertados por la empresa se registran en el sistema vía web.
- **Registro de reservas de pedidos computarizado.-** Las reservas de pedidos de los clientes se registran en el sistema y están presentes para su visualización y utilización durante el flujo de procesos de reserva, preparación y entrega del pedido.
- **Presentación y reserva de pedidos en línea.-** Toda la información de reservas puede ser accedida vía internet según corresponda a los diferentes estados que puede tener un pedido:
    - Reservado.- Es el pedido que se encuentra esperando la hora para ser entregado, esta hora será indicada por el cliente, la cual tendrá que cumplir un tiempo de anticipación especificado como política de la empresa; y como forma de garantizar la preparación del pedido el sistema tiene una opción flexible que puede ser modificada según sean las políticas de la empresa, en caso de no cumplir dicha política, el pedido no pasará del estado reservado a ordenado.

- Ordenado.- Es el pedido que pasa a ser ordenado en cocina al momento que el usuario encargado (recepcionista) permite éste paso cumpliendo las políticas de efectivización de la reserva.
- Listo.- Es el pedido que está listo para ser entregado al cliente según haya sido su solicitud de tipo de entrega: en local o para llevar.
- Entregado.- Es el pedido que ya fue entregado al cliente.
- Anulado.- Es la reserva que fue anulada por el mismo cliente, administrador del sistema, recepcionista o algún proceso incompleto.

- **Interfaz de interacción según el rol del empleado.-** Las interfaces de usuario varían según el rol del empleado:
  - Administrador del sistema.- Posee privilegios sobre todas las funcionalidades del sistema.
  - Recepcionista.- Controla todos los estados de los pedidos.
  - Chef.- Encargado de la administración de los productos y control del estado listo de los pedidos.
  - Mesero.- Encargado de la entrega de pedidos consumidos en el local.
  - Cliente.- Realiza la reserva de su pedido en línea.
- **Reportes automatizados en línea.-** Los reportes serán obtenidos en línea.
- **Armado de menús y organización de productos.-** Todos los productos estarán registrados en el sistema dentro un grupo específico en el cual podrán ser intercambiados según las preferencias del cliente. En estos grupos los usuarios con privilegios correspondientes podrán indicar que productos pueden ser intercambiados y cuáles no. Posteriormente, el armado de combos se realizará eficientemente con una interfaz amigable con los grupos de productos; cada combo será compuesto por dos o más

productos, su precio total será calculado por la suma de los precios de cada producto que lo compone menos su descuento asignado.

- **Interfaz de presentación de productos en línea.-** Todos los productos y combos ofrecidos por la empresa estarán presentes en un portal web.
- **Sistema multiplataforma para el manejo de la información en atención al cliente y gestión de reservas en línea "My Burguer".-** Es el componente 1 del proyecto que pretende mejorar el manejo de información y tiempo de atención al cliente mediante la gestión de reservas en línea.
- **Programa capacitación.-** Se realizará una capacitación al personal de la empresa ya que son los usuarios finales y beneficiarios quienes van a utilizar el sistema.
- **Presentación del sistema para publicar en las redes sociales.-** Se lanzarán anuncios publicitarios del sistema en las redes sociales.
- **Capacitación y sociabilización del sistema.-** Es el componente 2 del proyecto, la capacitación permitirá que el usuario final pueda utilizar de manera óptima y correcta todas las funcionalidades del sistema. La sociabilización ayudará a que más personas conozcan los productos que se ofrecen y la oportunidad de realizar reservas de pedidos en línea en "My Burguer".
- **Optimizar la atención en general de los clientes.-** Con el manejo de información adecuado los procesos involucrados en toda la atención al cliente se realizarán de forma eficiente.
- **Presentación visual de la empresa "My Burguer" y de los productos que ofrece en la web.-** Las personas podrán conocer a la empresa y sus productos mediante el portal web del sistema.

3.2.1 Análisis de causas del problema

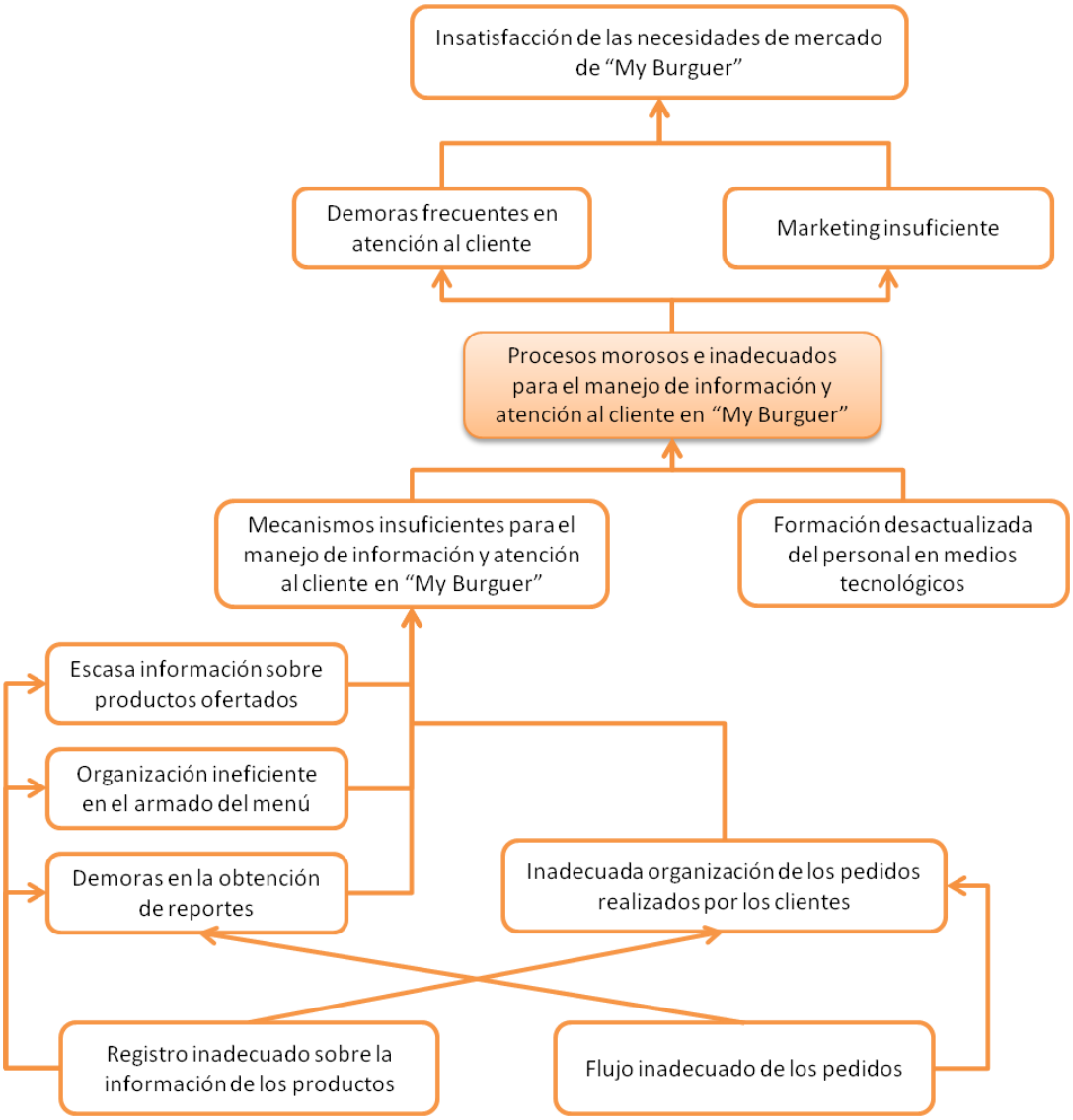


Figura 1. Árbol de problemas

3.2.2 Análisis de objetivos

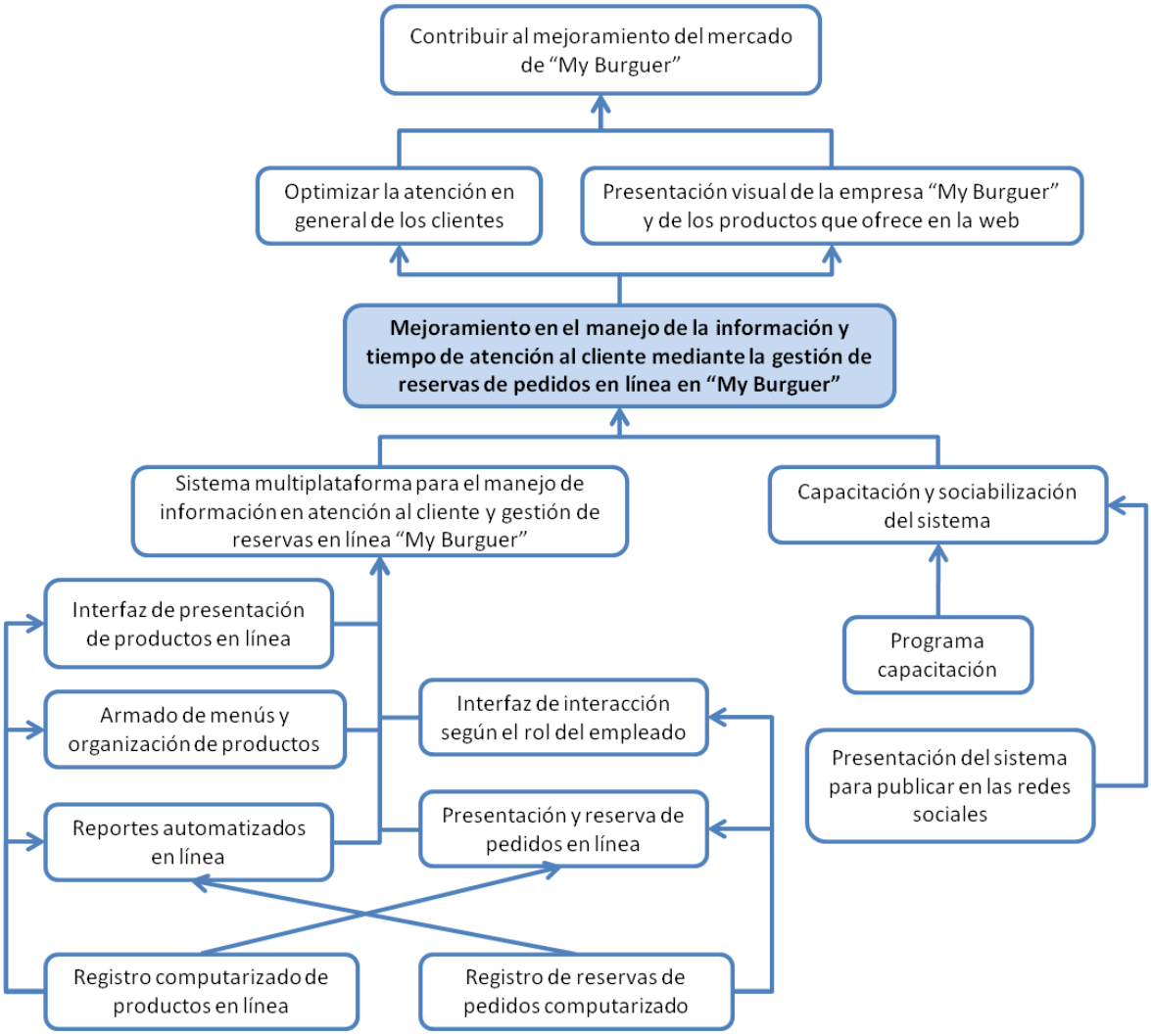


Figura 2. Árbol de objetivos

3.2.3 Situación planteada con y sin proyecto

Situación sin proyecto	Situación con proyecto
- Registro inadecuado sobre la información de los productos. - Flujo inadecuado de los pedidos. - Inadecuada organización de los pedidos realzados por lo clientes. - Demoras en la obtención de reportes. - Organización ineficiente en el armado del menú. - Escasa información sobre productos ofertados. - Mecanismos insuficientes para el manejo de información y atención al cliente en “My Burguer”. - Inadecuada formación del personal en medios TIC. - Procesos morosos e inadecuados para el manejo de información y atención al cliente en “My Burguer”. - Demoras frecuentes en atención al cliente. - Marketing insuficiente. - Insatisfacción de las necesidades de mercado de “My Burguer”.	- Registro computarizado de productos en línea. - Registro de reservas de pedidos computarizado. - Presentación y reserva de pedidos en línea. - Interfaz de interacción según el rol del empleado. - Reportes automatizados en línea. - Armado de menús y organización de productos. - Interfaz de presentación de productos en línea. - Sistema multiplataforma para el manejo de información en atención al cliente y gestión de reservas en línea de “My Burguer”. - Presentación del sistema para publicar en las redes sociales. - Programa capacitación. - Capacitación y sociabilización del sistema. - Optimizar el manejo de información y tiempo de atención al cliente mediante la gestión de reservas de pedidos en “My Burguer”.

	- Presentación visual de la empresa "My Burguer y de los productos que ofrece en la web". - Optimizar la atención en general de los clientes. - Contribuir al mejoramiento del mercado de pedidos "My Burguer".

Tabla 3. Situación planteada con y sin proyecto

3.3 Objetivos

3.3.1 Objetivo general

Optimizar el manejo de la información y atención al cliente y mediante gestión de reservas de pedidos en línea "My Burguer".

3.3.2 Objetivos específicos

- Desarrollar un sistema multiplataforma para el manejo de información en atención al cliente y gestión de reservas en línea de "My Burguer"..
- Desarrollar un programa de capacitación y sociabilización para el uso adecuado del sistema.

3.4 Marco lógico del proyecto

Resumen Narrativo del Proyecto	Indicadores	Medios de Verificación	Supuestos
Fin Contribuir al mejoramiento del mercado en "My Burguer".	Al final del 1° año de ejecución del proyecto, la afluencia de clientes se incrementó en un 30% denotando un incremento en los ingresos de la empresa de "My Burguer". **Categoría: Eficacia**	Informe otorgado por la gerencia del incremento en volumen de ventas.	El funcionamiento de la empresa es normal. Cuentan con medios tecnológicos suficientes para la implementación del sistema.
Objetivo General (Propósito) Optimizar el manejo de la información y atención al cliente y mediante gestión de reservas de pedidos en línea "My Burguer".	Al final de la ejecución del proyecto se mejoró al menos en un 39,85% los tiempos de atención al cliente en la empresa "My Burguer". **Formula: 1-(Tiempo invertido en la atención al cliente con el proyecto / Tiempo invertido en la atención al cliente sin el proyecto) * 100** **Categoría: Eficiencia**	Resultado de las estadísticas realizadas por la Gerencia con relación al tiempo de respuesta en atención al cliente.	La Gerencia de "My Burguer" apoya el desarrollo y ejecución del proyecto **Mejoramiento en el manejo de la información y atención al cliente mediante la gestión**

	Al final de la ejecución del proyecto se redujo el tiempo de acceso a las listas de productos en al menos un 93,42%. **Formula: (1-(Tiempo invertido en el acceso a las listas de productos con el proyecto / Tiempo invertido en el acceso a las listas de productos sin el proyecto)) * 100** **Categoría: Eficiencia** Al final de la ejecución del proyecto se redujo el tiempo de acceso a las listas de precios en al menos un 78,79%. **Formula: (1-(Tiempo invertido en el acceso a las listas de precios con el proyecto / Tiempo invertido en el acceso a las listas de precios sin el proyecto)) * 100** **Categoría: Eficiencia** Al final de la ejecución del proyecto se redujo el tiempo de armado de combos dentro un periodo	Resultado de las estadísticas realizadas por la Gerencia con relación al tiempo invertido en el manejo de la información.	**de reservas de pedidos en línea en "My Burguer".**

	mensual en al menos un 87,50%. **Formula: (1-(Tiempo invertido en el armado de combos mensual con el proyecto / Tiempo invertido en el armado de combos mensual sin el proyecto)) * 100** **Categoría: Eficiencia** Al final de la ejecución del proyecto se redujo el tiempo de asignaciones de grupos intercambiables dentro un periodo mensual en al menos un 83,33%. **Formula: (1-(Tiempo invertido en asignaciones de grupos intercambiables mensual con el proyecto / Tiempo invertido en asignaciones de grupos intercambiables mensual sin el proyecto)) * 100** **Categoría: Eficiencia** Al final de la ejecución del proyecto se redujo el tiempo de acceso a las listas de reservas diarias por cliente en al menos un 62,16%.		

	Formula: (1-(Tiempo invertido en el acceso a listas de reservas diarias por cliente con el proyecto / Tiempo invertido en el acceso a listas de reservas diarias por cliente sin el proyecto)) * 100 **Categoría: Eficiencia** Al final de la ejecución del proyecto se redujo el tiempo de acceso al detalle de preparación y contenidos de productos en al menos un 60%. **Formula: (1-(Tiempo invertido en el acceso al detalle de preparación y contenidos de productos con el proyecto / Tiempo invertido en el acceso al detalle de preparación y contenidos de productos sin el proyecto)) * 100** **Categoría: Eficiencia** Al finalizar el proyecto el 100% de información de las reservas de “My Burguer” se encuentra disponible, completa, actualizada y mejorada. **Categoría: Eficacia**		

Objetivos Específicos (Componentes) 1. Sistema multiplataforma para el manejo de información en atención al cliente y gestión de reservas en línea de "My Burguer".	Al finalizar el proyecto se ha desarrollado en un 100% el Sistema multiplataforma para el manejo de información en línea de "My Burguer" de acuerdo a los requisitos establecidos bajo la norma IEEE830.	Documentación del sistema en base a la especificación de requisitos de la norma IEEE 830 aprobada por los docentes de Taller III. Carta de respaldo de la gerencia "My Burguer" respecto a la conformidad y aceptación del sistema.	Información de requerimientos recopilada y confiable. Existe voluntad y compromiso institucional por parte de la empresa para la implantación del sistema.
2. Capacitación y sociabilización del sistema.	Al finalizar el proyecto, se ha implementado en un 80% el programa de capacitación sobre el manejo del sistema informático al personal de "My Burguer"; se realizó una presentación del sistema para promocionar a través de las redes sociales.	Documentación del programa de capacitación. Certificados de capacitación.	Interés y disponibilidad de tiempo del personal de la empresa, para la realización de la capacitación sobre el sistema desarrollado.

Actividades **Componente I** **1. Sistema multiplataforma para el manejo de información en atención al cliente y gestión de reservas en línea de "My Burguer"** 1.1. Especificación de requerimientos del sistema bajo la norma IEEE 830. 1.2. Análisis y diseño del sistema 1.3. Programación del sistema 1.4. Pruebas realizadas al sistema.	En el mes de junio de 2014 se inició la espeficicación de requerimientos del sistema bajo la norma IEEE 830 y hasta la fecha se conluyó en un 100%. En el mes de agosto de 2014 se inició el análisis y diseño del sistema, hasta la fecha se concluyó al 100%. En el mes de octubre de 2014 se inició la programación del sistema y hasta la fecha se completó en un 95%.	Documentación obtenida de todo el análisis y diseño del sistema. Documentos de las pruebas realizadas al sistema.	Disponibilidad de brindar información oportuna por parte del personal y los cooperantes del trabajo del proyecto. Coordinación entre el equipo de trabajo del proyecto y el usuario final. Presupuesto económico disponible para el desarrollo del proyecto.

1.5. Implantación del sistema. **Componente II** **2. Capacitación y sociabilización del sistema.** 2.1. Diseñar y desarrollar el programa de capacitación. 2.2. Diseñar la presentación del sistema para su lanzamiento publicitario en las redes sociales.	Al finalizar la gestion 2015 se realizó las pruebas al sistema en un 100% En diciembre de 2015 se realizó la capacitación del sistema al personal de la empresa en un 75% En enero de 2016 se diseñó la presentación del sistema en un 95% para su publicación en las redes sociales. **Costo componente I: Bs. 17700** **Costo componente II: Bs. 3500** **Total: Bs. 21200**	Documento del plan de Capacitación para el usuario final. Carta de conformidad y respaldo con el desarrollo de la capacitación por parte de la Gerencia de "My Burguer". Fotografías tomadas durante la realización de la capacitación. Encuestas post capacitación.	Asistencia del personal a la capacitación. Durante la capacitación el personal hace preguntas sobre el sistema. Los usuarios del sistema después de la capacitación aportan sugerencias en cuanto a posibles mejoras del sistema las cuales se toman en cuenta para una permisible implementación.

Tabla 4. **Matriz del marco lógico**

3.5 Metodología del proyecto

Para el desarrollo del proyecto se utilizó la metodología de la Matriz de Marco Lógico (MML).

Componente 1.

Para el desarrollo del componente sistema se hará uso de la metodología RUP en todo el proceso de desarrollo del sistema, el cual es un enfoque iterativo que propone una comprensión incremental del problema a través de refinamientos sucesivos e implementación de una solución efectiva a través de varios ciclos. Como parte del enfoque iterativo se encuentra la flexibilidad para acomodarse a nuevos requerimientos o cambios tácticos de los objetos de negocio.

El Lenguaje Unificado de Modelado UML, el cual constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

UML (Lenguaje de Modelado Unificado) es una metodología de Ingeniería de Software basado en una notación gráfica la cual permite: especificar, construir, visualizar y documentar los objetos de un sistema.

Así también se utilizará un modelo entidad-relación que es una herramienta para el modelado de datos que permite representar las entidades relevantes del sistema así como sus interrelaciones y propiedades.

Componente 2.

Para el desarrollo de este componente se realizará la capacitación en el puesto al personal de la empresa, se coordinará una reunión con el gerente y los usuarios del sistema para acordar la fecha y hora de las capacitaciones; la elaboración del material de capacitación estará dirigida a personas con conocimientos básicos en computación; el desarrollo de las capacitaciones se efectuará según lo acordado en las reuniones previamente realizadas.

La capacitación se llevará a cabo en el ambiente de “My Burguer” y cumplirán las siguientes metas:

- Aplicar y describir diversas Herramientas de las Tecnologías de Información y Comunicación (TIC) complementarias para el uso del sistema.
- Se hará una descripción completa sobre el manejo óptimo del Sistema multiplataforma para el manejo de información en línea de “My Burguer”.
- Se aplicará un permanente seguimiento a los participantes para mantener una colaboración activa en el programa de capacitación.
- Se levantará información de Internet sobre temas relacionados con el desarrollo del uso de las Tics en empresas.

Se desarrollarán recursos diapositivas según los contenidos usando las TIC’s para el proceso de E-A.

Para la sociabilización del sistema se publicarán presentaciones del sistema en las redes sociales.

3.6 Descripción y relación de las estrategias con los objetivos

Estrategias	Objetivos Específicos
Desarrollo y planificación del componente sistema utilizando la metodología RUP.	Sistema multiplataforma para el manejo de información en línea de “My Burguer”.
Planificación del programa de capacitación definiendo horarios adecuados para cada grupo de usuarios (Personal de “My Burguer”) y sociabilización del sistema en las redes sociales detallando el proceso de reservas en línea.	Programa de capacitación para el uso del sistema, dirigido al personal involucrado. Sociabilizar el sistema mostrando el proceso de reservas en línea.

Tabla 5. Descripción y relación de las estrategias con los objetivos

3.7 Cronograma de actividades

N º	Actividad	Nº días	M 1	M 2	M 3	M 4	M 5	M 6	M 7	M 8
1.	**Sistema multiplataforma para el manejo de la información en atención al cliente y gestión de reservas en línea de "My Burguer"**	**190**								
	Identificación de requerimientos del sistema bajo la norma IEEE 830	**30**	**X**							
	Análisis y diseño del sistema	**70**		**X**	**X**	**X**				
	Programación del sistema	**60**					**X**	**X**	**X**	
	Pruebas del sistema	**20**								**X**
	Implantación del sistema	**10**								**X**
2	**Capacitación y sociabilización del sistema**	**15**								
	Diseñar y desarrollar el programa de capacitación	**11**							**X**	
	Diseñar la presentación del sistema para su lanzamiento publicitario en las redes sociales	**4**								**X**

Figura 3. Cronograma de actividades

3.8 Resultados esperados

El resultado esperado es la realización de:

- **Sistema multiplataforma para el manejo de la información en atención al cliente y gestión de reservas en línea "My Burguer"**, desarrollado según los requerimientos el cual permitirá optimizar el manejo de información y tiempo de atención al cliente mediante la gestión de reservas de pedidos accedidos desde internet y desde cualquier plataforma.
- **Capacitación y sociabilización del sistema,** mediante la capacitación se espera formar al personal con el manejo de las funcionalidades del sistema, con lo cual se obtiene el máximo provecho del sistema desarrollado; con la sociabilización se espera dar a conocer el servicio de reservas de pedidos en línea y promocionar la empresa "My Burguer".

3.9 Transferencia de resultados

Medios y estrategias para la transferencia de resultados.

Por Convenio: Toda la información es proporcionada por la empresa "My Burguer", mediante un convenio firmado, mediante el cual se permite realizar ingeniería de requerimientos de la empresa.

Grupo de beneficiarios de los resultados

Con el proyecto realizado se beneficiarán los siguientes grupos de involucrados:

- ✓ "My Burguer"
- ✓ Clientes
- ✓ Empleados

Capítulo II

Componentes

4. Componente 1: Sistema multiplataforma para el manejo de la información en atención al cliente y gestión de reservas en línea "My Burguer".

4.1 Marco Teórico

4.1.1 Antecedentes

Una organización es un sistema compuesto por tres elementos: personas, materiales e información. Los sistemas de información, por su parte, surgen como sistemas complejos y abiertos que interactúan con otros sistemas y subsistemas como parte de su actuación. Por los años 90, una de las concepciones más defendidas por la gestión de la información fue que las organizaciones son sistemas de información.

El uso de ciertos conceptos tomados de la teoría de sistemas y del campo de la informática llevó a un alto grado de desarrollo entre los sistemas de información. Aunque existen diversas definiciones, hechas desde diferentes enfoques, sobre los sistemas de información, en su gran mayoría tienen puntos en común. El análisis realizado sobre las definiciones más frecuentes efectuadas en la década de los años 90 revela que constituyen un conjunto integrado de procesos, elementos o componentes que –según las estrategias y necesidades de una organización– recopilan, elaboran y distribuyen la información necesaria.

Un sistema moderno de gestión de información exige la aplicación de nuevas tecnologías de información; sin embargo, la tecnología por sí sola no es suficiente para lograr una buena gestión de información. Son diversos los procesos que conforman los sistemas de gestión de información; ellos generan las entradas y salidas del sistema o de otros procesos relacionados; también pueden identificarse, controlarse, corregirse o actualizarse en la medida en que se producen las transformaciones del entorno y evoluciona la organización, como vía incuestionable para garantizar su calidad, eficiencia y mejora continua. A modo de resumen de este antecedente de marco teórico, puede decirse que los sistemas de gestión de información, en su definición más

general, se refieren al conjunto de todos los componentes necesarios que se interrelacionan, con el objetivo de tramitar y facilitar la información sobre el tema de interés para su consumo en cualquier medio, momento y lugar.

4.1.2 Estudio de la variable independiente: Gestion de la Información

4.1.2.1 "MyBurguer"

"My Burguer" es una empresa en crecimiento dedicada a la venta de comida rápida en la ciudad de Tarija, con el propósito de aumentar clientes satisfaciendo sus necesidades.

4.1.3 Estudio de la variable dependiente: Sistema de Gestion de la Infomación

El manejo constante de información valiosa y cuantiosa de diversos establecimientos académicos que procesan tal información aun basándose en un sistema manual que muchas veces es lento, inseguro e inadecuado ha puesto en claro la necesidad de un Sistema mucho más rápido y confiable es por eso que se propone como solución el desarrollo de un sistema de gestión de información de esta forma se facilita a la empresa mejorar la eficiencia en el manejo de la información en tiempos de atención al cliente. Para esto hay tomar en cuenta las tecnologías que se aplicarán.

4.1.4 Relación entre la variable independiente y dependiente

X (i) Desarrollo de Sistema multiplataforma para el manejo de información en atención al cliente y gestión de reservas en línea.	**Y (i) Control de información en atención al cliente y gestión de reservas en línea.**
Analizar la situación actual de "MyBurguer".	Factibilidad de los procesos.
Determinar los requerimientos necesarios.	Mejor control en manejo de información y atención al cliente.
Elaborar el diseño lógico y físico del Sistema.	Solucionar necesidades del área de reservas.

4.2 Metodología para el desarrollo del sistema

4.2.1 Metodología RUP (Racional Unified Process) [15]

El Proceso Unificado Racional – RUP (Rational Unified Process) es un marco de referencia para el desarrollo del sistema y junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

Los **procesos de RUP** estiman tareas y horario del plan midiendo la velocidad de iteraciones concerniente a sus estimaciones originales. Las iteraciones tempranas de proyectos conducidos por RUP se enfocan fuertemente sobre arquitectura del sistema; la puesta en práctica rápida de características se retrasa hasta que se ha identificado y se ha probado una arquitectura firme.

Nos permite realizar un levantamiento exhaustivo de requerimientos.

Las actividades de RUP se centran en crear y mantener modelos, utilizando UML, en forma efectiva.

Busca detectar defectos en las fases iníciales.

Intenta reducir al número de cambios tanto como sea posible.

Realiza el Análisis y diseño, tan completo como sea posible.

Diseño genérico, intenta anticiparse a futuras necesidades.

Las necesidades de clientes no son fáciles de discernir.

Existe un contrato prefijado con los clientes.

El cliente interactúa con el equipo de desarrollo mediante reuniones.

Características esenciales:

Los autores de RUP destacan que el proceso de software propuesto por RUP tiene tres características esenciales: está dirigido por los Casos de Uso, está centrado en la arquitectura, y es iterativo e incremental.

Está dirigido por los Casos de Uso: Los Casos de Uso son una técnica de captura de requisitos que fuerza a pensar en términos de importancia para el usuario y no sólo en términos de funciones que sería bueno contemplar. Se define un Caso de Uso como un fragmento de funcionalidad del sistema que proporciona al usuario un valor añadido. Los Casos de Uso representan los requisitos funcionales del sistema.

Los Casos de Uso no sólo inician el proceso de desarrollo sino que proporcionan un hilo conductor, permitiendo establecer trazabilidad entre los artefactos que son generados en las diferentes actividades del proceso de desarrollo.

En RUP los Casos de Uso no son sólo una herramienta para especificar los requisitos del sistema. También guían su diseño, implementación y prueba.

Está Centrado en su Arquitectura: La arquitectura de un sistema es la organización o estructura de sus partes más relevantes, lo que permite tener una visión común entre todos los involucrados (desarrolladores y usuarios) y una perspectiva clara del sistema completo, necesaria para controlar el desarrollo.

La arquitectura involucra los aspectos estáticos y dinámicos más significativos del sistema, está relacionada con la toma de decisiones que indican cómo tiene que ser construido el sistema y ayuda a determinar en qué orden. Además la definición de la arquitectura debe tomar en consideración elementos de calidad del sistema, rendimiento, reutilización y capacidad de evolución por lo que debe ser flexible durante todo el proceso de desarrollo. La arquitectura se ve influenciada por la plataforma software, sistema operativo, gestor de bases de datos, protocolos, consideraciones de desarrollo como sistemas heredados. Muchas de estas restricciones constituyen requisitos no funcionales del sistema.

En el caso de RUP además de utilizar los Casos de Uso para guiar el proceso se presta especial atención al establecimiento temprano de una buena arquitectura

que no se vea fuertemente impactada ante cambios posteriores durante la construcción y el mantenimiento.

Cada producto tiene tanto una función como una forma. La función corresponde a la funcionalidad reflejada en los Casos de Uso y la forma la proporciona la arquitectura. Existe una interacción entre los Casos de Uso y la arquitectura, los Casos de Uso deben encajar en la arquitectura cuando se llevan a cabo y la arquitectura debe permitir el desarrollo de todos los Casos de Uso requeridos, actualmente y en el futuro. Esto provoca que tanto arquitectura como Casos de Uso deban evolucionar en paralelo durante todo el proceso de desarrollo de software.

Es Iterativo e Incremental: Según el equilibrio correcto entre los Casos de Uso y la arquitectura es algo muy parecido al equilibrio de la forma y la función en el desarrollo del producto, lo cual se consigue con el tiempo. Para esto, la estrategia que se propone en RUP es tener un proceso iterativo e incremental en donde el trabajo se divide en partes más pequeñas o mini proyectos. Permitiendo que el equilibrio entre Casos de Uso y arquitectura se vaya logrando durante cada mini proyecto, así durante todo el proceso de desarrollo. Cada mini proyecto se puede ver como una iteración (un recorrido más o menos completo a lo largo de todos los flujos de trabajo fundamentales) del cual se obtiene un incremento que produce un crecimiento en el producto.

Una iteración puede realizarse por medio de una cascada. Se pasa por los flujos fundamentales (Requisitos, Análisis, Diseño, Implementación y Pruebas), también existe una planificación de la iteración, un análisis de la iteración y algunas actividades específicas de la iteración. Al finalizar se realiza una integración de los resultados con lo obtenido de las iteraciones anteriores.

Fases en el ciclo de Desarrollo

Este proceso de desarrollo considera que cualquier desarrollo de un sistema software debe pasar por cuatro fases que se describirán a continuación, la figura

muestra las Fases de desarrollo y los diversos flujos de trabajo involucrados dentro de cada fase con una representación gráfica en cuál de los flujos se hace mayor énfasis según la fase, cabe destacar el flujo de trabajo concerniente al negocio.

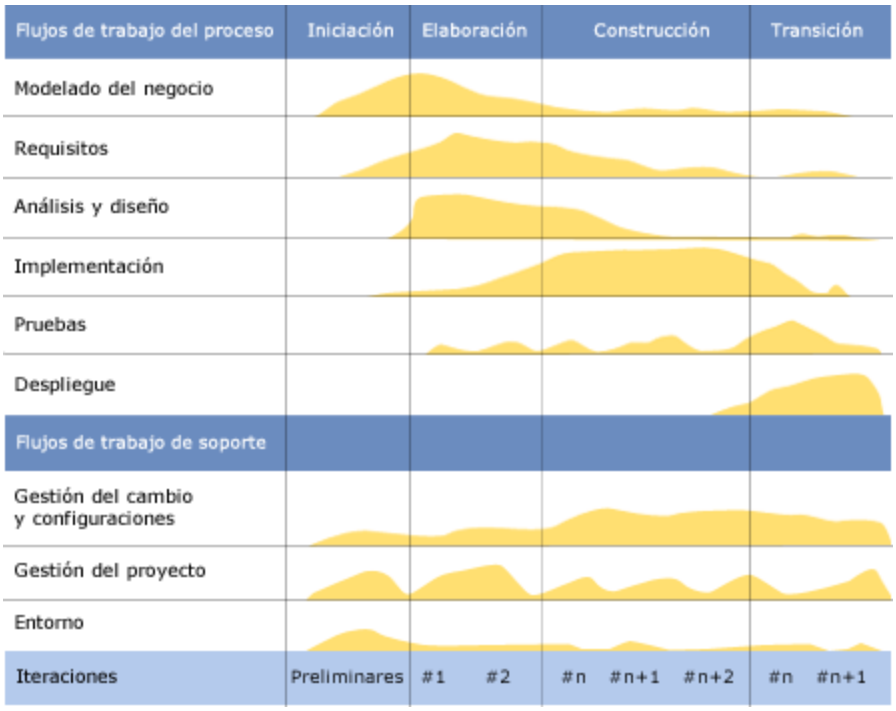


Figura 4. Fases de desarrollo de R.U.P.

Fase 1: Preparación Inicial ("Incepción")

Su objetivo principal es establecer los objetivos para el ciclo de vida del producto. En esta fase se establece el caso del negocio con el fin de delimitar el alcance del sistema, saber qué se cubrirá y delimitar el alcance del proyecto.

El caso de negocio incluye criterios de éxito, la evaluación de riesgos, y la estimación de los recursos necesarios, y un plan de la fase que muestre las fechas previstas e hitos importantes.

Fase 2: Preparación Detallada ("Elaboración")

Su objetivo principal es plantear la arquitectura para el ciclo de vida del producto. En esta fase se realiza la captura de la mayor parte de los

requerimientos funcionales, manejando los riesgos que interfieran con los objetivos del sistema, acumulando la información necesaria para el plan de construcción y obteniendo suficiente información para hacer realizable el caso del negocio.

El resultado de la fase de elaboración es:

Un modelo de caso de uso (por lo menos 80% completo) - todos los casos de uso y actores deben haber sido identificados-, y se han desarrollado la mayoría de las descripciones de casos de uso.

Requerimientos suplementarios que capturan los requerimientos no funcionales o cualquier requerimiento que no se asocie a un caso de uso específico.

Fase 3: **Construcción ("Construcción")**

Su objetivo principal es alcanzar la capacidad operacional del producto. En esta fase a través de sucesivas iteraciones e incrementos se desarrolla un producto software, listo para operar, éste es frecuentemente llamado versión beta.

Fase 4: Transición ("Transición")

Su objetivo principal es realizar la entrega del producto operando, una vez realizadas las pruebas de aceptación por un grupo especial de usuarios y habiendo efectuado los ajustes y correcciones que sean requeridos.

Éste incluye:

Operación en paralelo con un sistema anterior que el nuevo sistema esté sustituyendo.

La conversión de las bases de datos operacionales.

Entrenamientos y capacitación de los usuarios y la gente de mantenimiento.

4.2.2 Los diagramas del lenguaje unificado de modelado [17]

UML, por sus siglas en inglés, Unified Modeling Language, es el lenguaje de modelado de sistemas más conocido y utilizado en la actualidad. Es un lenguaje

gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio, funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y compuestos reciclados.

Es importante remarcar que UML es un "lenguaje de modelado" para especificar o para describir métodos o procesos. Se utiliza para definir un sistema, para detallar los artefactos en el sistema y para documentar y construir. En otras palabras, es el lenguaje en el que está descrito el modelo.

Se puede aplicar en el desarrollo de software gran variedad de formas para dar soporte a una metodología de desarrollo de software (tal como el Proceso Unificado Racional o RUP), pero no especifica en sí mismo qué metodología o proceso usar.

UML no puede compararse con la programación estructurada, pues UML significa Lenguaje Unificado de Modelado, no es programación, solo se diagrama la realidad de una utilización en un requerimiento. Mientras que, programación estructurada, es una forma de programar como lo es la orientación a objetos, sin embargo, la programación orientada a objetos viene siendo un complemento perfecto de UML, pero no por eso se toma UML sólo para lenguajes orientados a objetos.

Por otro lado, no debe olvidarse que realizar uno de estos diagramas insume tiempo. Más allá que el o los diagramas deben realizarse correctamente, el desarrollador debe considerar cuanto tiempo invierte en estas actividades.

Estos diagramas se pueden organizar en dos grupos:

Los que describen el comportamiento del negocio, del sistema, de un aspecto en particular.

- Diagrama de Actividad (Activity Diagram): Representa los procesos de negocio o la lógica de un sistema complejo. Incluye, opcionalmente, el

flujo de datos. el nivel de abstracción suele ser bastante alto, pero pueden realizarse diagramas de actividad exploratorios cuando la lógica que se trata es compleja.

- Diagrama de Casos de Uso (Use Case Diagram): Muestra casos de uso individuales, actores y las relaciones entre ellos. El Proceso Unificado dice está dirigido por los casos de uso, esto significa que este diagrama (en el nivel de abstracción que sea) es la base del lenguaje de modelado y representación.
- Diagrama de Secuencia (Sequence Diagram): Muestra la secuencia de la lógica, el orden en que se suceden los mensajes. Importante, especialmente cuando se trabaja en ambientes altamente compartidos.

Los que describen la estructura, la forma, la organización.

- Diagrama de Clases (Class Diagram): Muestra una colección de clases, sus tipos, sus contenidos y sus relaciones. Importantísimo representa el modelo de datos, y en consecuencia su persistencia en alguna forma de almacenamiento.
- Diagrama de Componentes (Component Diagram): Describe los elementos que componen un sistema. Debe detallar los elementos o componentes, las interacciones y relaciones así como las interfaces públicas.
- Diagrama de Despliegue (Deployment Diagram): Muestra la arquitectura de ejecución de un sistema. Incluye nodos, entornos de hardware y software.

UML se ha establecido como el estándar en la industria de desarrollo de software. Es cierto que puede utilizarse otro tipo de lenguaje, pero eso reduce la cantidad de personas que pueden leer (entender) el desarrollo.

4.2.3 Base de datos relacional

Una Base de Datos relacional es una base de datos en donde todos los datos visibles al usuario están organizados estrictamente como tablas de valores, y en donde todas las operaciones de la base de datos operan sobre estas tablas.

Estas bases de datos son percibidas por el usuario como una colección de relaciones normalizadas en diversos grados que varían como el tiempo.

El modelo relación representa un sistema de base de datos en un nivel de abstracción un tanto alejado de los detalles de la maquina subyacente, de la misma manera como por ejemplo, un lenguaje de tipo de PL/I representa un sistema de programación más bien abstracto, orientado de manera específica hacia las aplicaciones de Base de Datos.

4.3 Herramientas de construcción del sistema

4.3.1 NetBeans

Para el desarrollo del presente sistema se utilizo NetBeans porque es un entorno de desarrollo gratuito y de código abierto. Permite el uso de un amplio rango de tecnologías de desarrollo para aplicaciones Web. Además puede instalarse en varios sistemas operativos: Windows, Linux, Mac OS. Soporta el desarrollo de aplicación Java EJB y la integración con el servidor WildFly los cuales serán utilizados en el desarrollo del sistema.

4.3.2 Enterprise Architect

Enterprise Architect es una plataforma para el modelado, visualización y diseño, basada en el estándar UML 2.1, ésta herramienta facilita el diseño de los diagramas.

4.3.3 MySQL Workbench

Es una herramienta visual de diseño de bases de datos que integra desarrollo de software, administración de bases de datos, diseño de bases de datos, creación y mantenimiento para el sistema de base de datos MySQL.

En el desarrollo del sistema facilitará el análisis y diseño de la base de datos relacional. Ventajas:

- Libre, distribuida bajo la licencia GPL
- Multiplataforma. disponible para Windows, GNU/Linux. Mac
- Permite crear diagramas E-R
- Importar archivos SQL
- Permite generar los scripts SQL a partir del modelo creado.

4.3.4 WildFly

Es un servidor de aplicaciones, tiene la capacidad de escalar a más de un millón de conexiones. Cumple con toda las especificaciones que define JEE. Soporta los últimos estándares para el desarrollo web, posee una comunicación full-duplex. Esto es especialmente útil en la comunicación con los dispositivos móviles.

4.4 Tecnología utilizada

4.4.1 EJB

Los Enterprise Java Beans (EJB) son componentes del lado del servidor para la plataforma Java Enterprise Edition (Java EE), que apuntan a crear un desarrollo rápido y simple para aplicaciones distribuidas, transaccionales, seguras y portables.

Proporciona un modelo de componentes distribuido estándar del lado del servidor. EJB permite abstraerse de los problemas generales como ser: concurrencia, transacciones, persistencia, seguridad, etc. para centrarse en el desarrollo de la lógica de negocio en sí.

Se utilizará EJB para la implementación del sistema ya que éste requiere mayores necesidades de concurrencia o distribución.

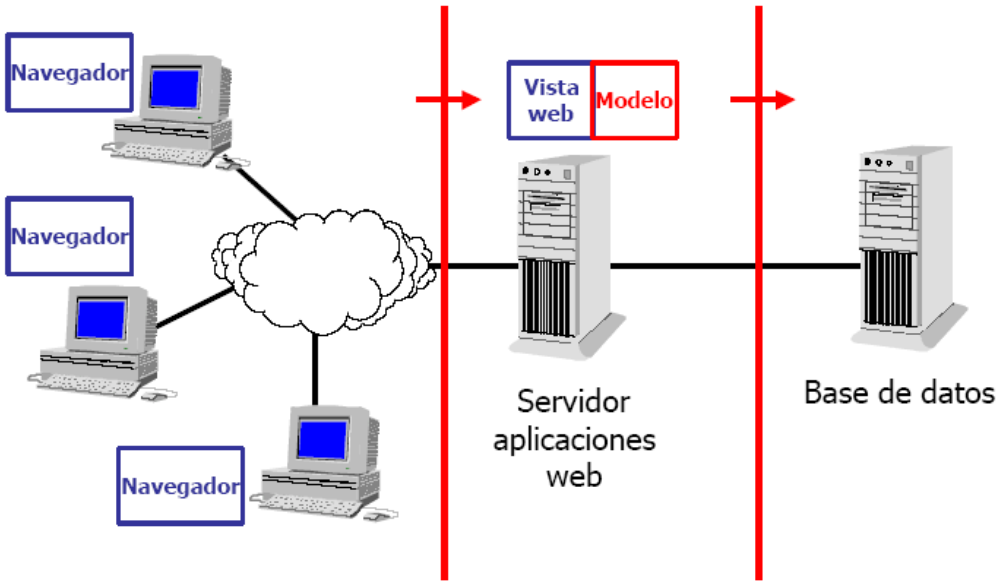
4.4.2 MyBatis

MyBatis es una herramienta de software libre; es un framework de persistencia disponible para Java que se encarga de mapear sentencias SQL y procedimientos almacenados con objetos a partir de ficheros XML o anotaciones. No se trata de un ORM. Simplifica la programación frente al uso directo de JDBC. Las líneas de código necesarias para ejecutar una sentencia se reducen casi siempre a una. Esta simplificación ahorra tiempo y evita errores. Además previene el temido SQL Inyection.

4.4.3 Java

Es un lenguaje de programación orientado a objetos. Puede ser ejecutado en cualquier dispoisitivo.

4.4.4 Patrón de diseño "Modelo Vista Controlador (MVC)" [12]



Es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos.

El patrón MVC se ve frecuentemente en *aplicaciones web*, su concepto se basa en separar el modelo de datos de la aplicación de su representación de cara al

usuario y de la interacción de éste con la aplicación, mediante la división de la aplicación en tres partes fundamentales:

- El modelo, que contiene la lógica de negocio de la aplicación.
- La vista, que muestra al usuario la información que éste necesita.
- El controlador, que recibe e interpreta la interacción del usuario, actuando sobre modelo y vista de manera adecuada para provocar cambios de estado en la representación interna de los datos, así como en su visualización.

Esta arquitectura ha demostrado ser muy apropiada para las aplicaciones web y especialmente adaptarse bien a las tecnologías proporcionadas por la plataforma J2EE, de manera que:

- El modelo, conteniendo lógica de negocio, sería modelado por un conjunto de clases Java, existiendo dos claras alternativas de implementación, utilizando objetos java tradicionales llamados POJOs (Plain Old Java Objects) o bien utilizando EJB (Enterprise JavaBeans) en sistemas con unas mayores necesidades de concurrencia o distribución.
- La vista proporcionará una serie de páginas web dinámicamente al cliente, siendo para él simples páginas HTML. El framework que genera estas páginas web a partir de distintos formatos es JSP (JavaServer Pages), que mediante un conjunto de tags XML proporciona un interfaz sencillo y adecuado a clases Java y objetos proporcionados por el servidor de aplicaciones. Esto permite que sean sencillas de desarrollar con la tecnología HTML. Entre estos tags tienen mención especial la librería estándar JSTL (JavaServer Pages Standard Tag Library) que proporciona una gran funcionalidad y versatilidad.
- El controlador en la plataforma J2EE se desarrolla mediante servlets, que hacen de intermediarios entre la vista y el modelo, más versátiles que los JSP para esta función al estar escritos como clases Java

normales, evitando mezclar código visual (HTML, XML...) con código Java.

Con todo lo anterior, el funcionamiento de una aplicación web J2EE que utilice el patrón arquitectural MVC se puede descomponer en una serie de pasos:

- El usuario realiza una acción en su navegador, que llega al servidor mediante una petición HTTP y es recibida por un servlet (controlador). Esa petición es interpretada y se transforma en la ejecución de código java que delegará al modelo la ejecución de una acción de éste.
- El modelo recibe las peticiones del controlador, a través de un interfaz o fachada que encapsulará y ocultará la complejidad del modelo al controlador. El resultado de esa petición será devuelto al controlador.
- El controlador recibe del modelo el resultado, y en función de éste, selecciona la vista que será mostrada al usuario, y le proporcionará los datos recibidos del modelo y otros datos necesarios para su transformación a HTML. Una vez hecho esto el control pasa a la vista para la realización de esa transformación.
- En la vista se realiza la transformación tras recibir los datos del controlador, elaborando la respuesta HTML adecuada para que el usuario la visualice.

Esta arquitectura de aplicaciones otorga varias ventajas clave al desarrollo del sistema, destacando:

- Al separar de manera clara la lógica de negocio (modelo) de la vista permite la reusabilidad del modelo.
- Permite una sencilla división de tareas, diseñar las vistas sin necesidad de mezclar código Java entre el código visual que se está desarrollando (tan sólo utilizando algunos tags, no muy diferentes de los usados en el código HTML).

4.4.5 Boostrap

Ya que es un requisito que el sistema funcione en equipos de escritorio y portátiles a través de navegadores web, se utilizó Boostrap que es un framework desarrollado por Twitter para crear interfaces y diseños web responsivos basados en HTML5, JavaScript, JQuery y CSS3, que presentará vistas llamativas y amigables.

Para el desarrollo de las vistas del sistema se adaptó una plantilla, con tecnología Boostrap, llamada DevOOPS.

4.5 Base de datos

4.5.1 MySQL

Es un sistema de administración de bases de datos (Database Management System, DBMS) para bases de datos relacionales.

Para el desarrollo del presente sistema se utilizo este gestor por ser una de base de datos relacional, multihilo y multiusuario, es muy rápido en la lectura de datos, ideal para aplicaciones web y además se integra en distintos sistemas operativos el cual es un requisito importante.

4.6 Glosario

Del Negocio.

Restaurante: establecimiento comercial público donde se paga por la comida y bebida, para ser consumidas en el mismo local.

Mesero: es la persona que tiene como oficio atender a los clientes del restaurante, proporcionándoles alimentos, bebidas, y asistencia durante la estancia, suele controlar un rango de varias mesas.

Chef: persona que cocina por oficio y profesión. Las funciones en la cocina están categorizadas, en función de los conocimientos y las especialidades de cada uno de los tipos de chef.

Recepcionista: Persona que actúa como un intermediario entre el cliente y otros miembros del personal, programa las reservas de pedidos.

Reserva: Productos solicitados por el cliente.

Orden: Detalle de la socilitud de productos, en la cual el chef se basa para prepararlos en cocina.

Del Sistema.

Administrar: Acción de agregar, modificar, eliminar y consultar la información de un determinado objeto o persona.

Administrador del Sistema: Persona encargada de ofrecer el soporte técnico y operativo.

Backup (copia de respaldo, copia de seguridad): Copia de ficheros o datos de forma que estén disponibles en caso de que un fallo produzca la pérdida de los originales. Esta sencilla acción evita numerosos, y a veces irremediables, problemas si se realiza de forma habitual y periódica.

Solicitud: Transacción o una diligencia cuidadosa o un pedido de un usuario del sistema hacia las funcionalidades y recursos del sistema.

Login: nombre de usuario. Es el nombre que adquiere el usuario para acceder a un determinado servicio del sistema.

Password (palabra de paso, contraseña): Conjunto de caracteres alfanuméricos que permite a un usuario el acceso a un determinado recurso o la utilización de un servicio dado.

Sesión: una sesión es la duración de una conexión empleando una capa de sesión de un protocolo de red, o la duración de una conexión entre un usuario (el agente) y un servidor, generalmente involucrando el intercambio de múltiples paquetes de datos entre la computadora del usuario y el servidor.

Racional Unified Process: Es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar

más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

Sistema Informático: Conjunto de partes (hardware y software) que funcionan relacionándose entre sí con un objetivo preciso. Los usuarios son parte del sistema informático.

Sistema Operativo: Un sistema operativo (SO) es un conjunto de programas o software destinado a permitir la comunicación del usuario con un ordenador y gestionar sus recursos de manera cómoda y eficiente. Comienza a trabajar cuando se enciende el ordenador, y gestiona el hardware de la máquina desde los niveles más básicos .Ejemplos Windows, Linux, MacOS, Solaris.

Usuarios: El usuario de un producto informático (bien sea hardware o software), es la persona a la que va destinada dicho producto una vez que ha superado las fases de desarrollo correspondientes. Normalmente, el sistema se desarrolla pensando en la comodidad del usuario final, y por esto se presta especial interés y esfuerzo en conseguir una interfaz de usuario lo más clara y sencilla posible.

De tecnología.

Computadora (computador, ordenador): Máquina electrónica capaz de procesar información siguiendo instrucciones almacenadas en programas. Antes que electrónicas estas máquinas fueron mecánicas o electromecánicas.

Dominio: Nombre base que agrupa a un conjunto de equipos o dispositivos y que permite proporcionar nombres de equipo más fácilmente recordables en lugar de una dirección IP numérica, Ej: sislabc.com.

Servidor WEB: Es un programa que implementa el protocolo HTTP (hypertext transfer protocol). Este protocolo está diseñado para transferir lo que llamamos hipertextos, páginas web o páginas HTML (hypertext markup language): textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de música.

URL: Significa Uniform Resource Locator, es decir, localizador uniforme de recurso. Es una secuencia de caracteres, de acuerdo a un formato estándar, que se usa para nombrar recursos, como documentos e imágenes en Internet, por su localización.

HTTP: El protocolo de transferencia de hipertexto (HTTP, HyperText Transfer Protocol) es el protocolo usado en cada transacción de la Web (WWW).

Enlaces: una referencia en un documento de hipertexto a otro documento o recurso.

Sistema de gestión de Base de Datos: Son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan.

Base de Datos: Es un conjunto de datos que pertenecen al mismo contexto almacenados sistemáticamente para su posterior uso.

Aplicación: Es un programa informático diseñado para facilitar al usuario la realización de un determinado tipo de trabajo.

MySQL: Sistema de gestión de base de datos relacional. Comúnmente utilizados en aplicaciones Web.

WildFlly: Servidor de WEB de código abierto.

Dirección IP (Internet Protocol): Número que identifica de manera lógica y jerárquica a una interfaz de un dispositivo (habitualmente una computadora) dentro de una red que utilice el protocolo IP (Internet Protocol).

Protocolo IP: El Protocolo de Internet (IP, de sus siglas en inglés Internet Protocol) es un protocolo no orientado a la conexión, usado tanto por el origen como por el destino para la comunicación de estos a través de una red (Internet).

Internet: Es un método de interconexión descentralizada de redes de computadoras implementado en un conjunto de protocolos denominado TCP/IP.

Browser o Motor de Navegación: Es una aplicación que permite al usuario recuperar y visualizar documentos de hipertexto, comúnmente descritos en HTML, desde servidores web de todo el mundo a través de Internet.

HTML (HyperText Markup Language): Es un lenguaje de marcación diseñado para estructurar textos y presentarlos en forma de hipertexto, que es el formato estándar de las páginas web.

Ajax (Asynchronous JavaScript And XML): es una técnica de desarrollo web para crear aplicaciones interactivas.

Javascript: Es un lenguaje interpretado, es decir, que no requiere compilación, utilizado principalmente en páginas web.

Java: es un lenguaje de programación orientado a objetos que fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible.

Acrónimos.

BBDD, BD: Bases de Datos, Base de Datos.

Abreviaturas.

ERS: Especificación de Requerimientos Software.

RUP: Racional Unified Process.

TIC: Tecnologías de la Información y Comunicación.

UML: Lenguaje Unificado de Modelado.

4.7 Plan de desarrollo del sistema

4.7.1 Introducción

Este plan de desarrollo del sistema es una respuesta al proyecto Mejoramiento en el manejo de la información y atención al cliente mediante la automatización de procesos en línea de “My Burguer”, de la asignatura de Taller III de la carrera de Ingeniería informática de la Facultad de Ciencias y Tecnología de la Universidad Autónoma Juan Misael Saracho. Este documento provee una visión global del enfoque de desarrollo propuesto.

El presente proyecto es propuesto por la Univ. Marcia Milenka Andrade Llanos basado en una metodología de Racional Unified Process en la que únicamente se procederá a cumplir con las tres primeras fases que marca la metodología, constando únicamente en la tercera fase de dos iteraciones. Es importante destacar esto puesto que utilizaremos la terminología RUP en este documento. Se incluirá el detalle para las fases de Inicio y Elaboración y adicionalmente se esbozaran las fases posteriores de Construcción y Transición para dar una visión global de todo proceso.

El enfoque desarrollo propuesto constituye una configuración del proceso RUP de acuerdo a las características del proyecto seleccionando los roles de los participantes, las actividades a realizar y los artefactos (entregables) que serán generados, éste documento es a su vez uno de los artefactos de RUP.

4.7.2 Propósito

El propósito del plan de desarrollo del sistema es proporcionar la información necesaria para controlar el proyecto. En él se describe el enfoque de desarrollo del sistema.

Los usuarios del Plan de Desarrollo del Sistema son:

- ✓ El director del proyecto lo utiliza para organizar la agenda y necesidades de recursos y para realizar su seguimiento y poder

cumplir con todas las fechas trazadas para su posterior culminación en el tiempo programado.

- ✓ Los miembros del equipo de desarrollo como programadores, diseñadores gráficos, lo usan para entender lo que deben hacer, cuando deben hacerlo y que otras actividades dependen de ello.

4.7.3 Alcance

Aplicando el plan de desarrollo del sistema se obtiene una herramienta importante para realizar el plan de trabajo el cual coadyuvará al cumplimiento de los objetivos en el tiempo propuesto gracias al cronograma de actividades establecido.

4.7.4 Resumen

El presente plan de desarrollo contiene la siguiente información:

Descripción del Proyecto: ofrece una descripción del propósito del proyecto, el alcance y los objetivos. También define los entregables que se esperan del producto.

Organización del Proyecto: describe la estructura de organización del equipo del proyecto.

Proceso de Gestión: explica el calendario, se determinan las fases y entregas incrementales del proyecto, y se describe cómo será llevado a cabo el seguimiento del proyecto.

Planes y Directrices aplicables: ofrece una visión general del proceso de desarrollo del sistema, incluidos los métodos, herramientas y técnicas a seguir.

4.8 Vista General del proyecto

Proporciona una descripción del propósito, alcance y objetivos del proyecto, estableciendo los artefactos que serán producidos y utilizados durante el proyecto.

4.8.1 Propósito

Proporcionar una herramienta software que permita automatizar el manejo de la información como ser las reservas en línea de los clientes, registrar y organizar productos y combos, generación de reportes cuando se lo requiera, además de otorgar a los usuarios un registro adecuado de la información: teniendo toda información almacenada en una Base de Datos a la que solo podrá acceder personal autorizado, brindando de esta manera total seguridad a todos los datos almacenados.

4.8.2 Alcances

Analizando factores predominantes dentro de la empresa, se llega a la conclusión de que la implementación del sistema automatizado dará beneficios claramente perceptibles, dando soluciones a problemas que arrastra; este sistema proyectará una solución a mediano plazo que beneficiará a dicha empresa.

Esta propuesta del sistema contiene una serie de alternativas de mejoramiento para las expectativas futuras de la empresa, las cuales se detallan a continuación:

Brindar seguridad al sistema mediante una clave de ingreso, permitiendo el acceso al mismo sólo al personal autorizado.

Opciones que permitan el registro automatizado consistente de usuarios y clientes, consultas, seguimiento de las reservas, además de realizar reportes.

Desarrollar un manual de usuario y de instalación del sistema.

4.8.3 Objetivos

Según el estudio previamente realizado para esta empresa se plantean los siguientes objetivos:

- Obtener un sistema que permita obtener reportes de forma rápida y sencilla.

- Otorgar a los usuarios total seguridad en el resguardo de su información, permitiendo el acceso sólo a usuarios autorizados.
- Obtener un sistema que permita generar reportes de manera rápida y sencilla.
- Crear una base de datos de acuerdo a las necesidades de la empresa.
- Diseñar una interfaz de usuario de fácil manejo.
- Desarrollar una aplicación web escalable.
- Desarrollar casos de prueba que garanticen la calidad del sistema.

4.8.4 Suposiciones y Restricciones

Suposiciones

Se supone que la empresa "My Burguer" cuenta con:

- Equipo de computación. En el que incluye un computador, impresora.
- La disponibilidad de otorgar toda la información necesaria para la creación del sistema.
- Personal con conocimientos básicos de computación.
- Cuenta con equipos para implementar el sistema

Restricciones

Se establece las siguientes restricciones para el sistema:

- El sistema será restringido, sólo usuarios privilegiados podrán acceder al sistema.
- Para la manipulación de la base de datos, sólo podrán acceder el personal autorizado.
- Deberá contar como mínimo con un sistema operativo.
- El administrador deberá contar con el conocimiento necesario para el manejo del sistema
- El sistema no soporta la emisión de facturas.
- El sistema no soporta el funcionamiento contable.

- El sistema no gestiona la planilla de sueldos del personal, tampoco realiza el control de horarios de ingresos y salida del personal.

4.9 Entregables del proyecto

A continuación se indican y describen cada uno de los artefactos que serán generados y utilizados por el proyecto y que constituyen los entregables. Esta lista constituye la configuración de RUP desde la perspectiva de artefactos, y que proponemos para este proyecto.

Es preciso destacar que de acuerdo a la filosofía de RUP (y de todo proceso iterativo e incremental), todos los artefactos son objeto de modificaciones a lo largo del proceso de desarrollo, con lo cual, sólo al término del proceso podríamos tener una versión definitiva y completa de cada uno de ellos. Sin embargo, el resultado de cada iteración y los hitos del proyecto están enfocados a conseguir un cierto grado de completitud y estabilidad de los artefactos. Esto será indicado más adelante cuando se presenten los objetivos de cada iteración.

4.9.1 Plan de desarrollo del sistema

El plan de desarrollo del sistema se describirá paso a paso en los siguientes puntos del proyecto en base a la metodología RUP.

4.9.2 Modelo de Objetos del Negocio

Es un modelo que describe la realización de cada caso de uso del negocio, estableciendo los actores internos, la información que en términos generales manipulan y los flujos de trabajo (workflows) asociados al caso de uso del negocio.

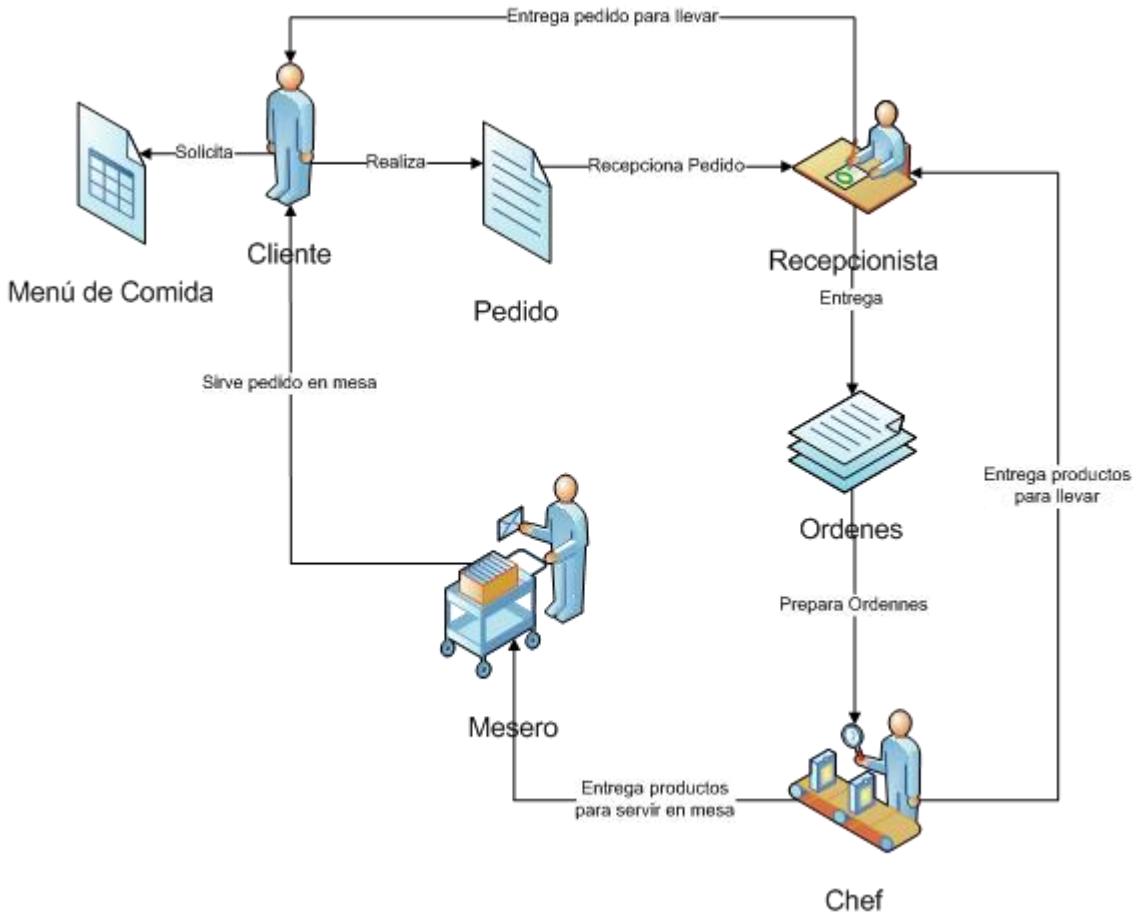


Figura 5. Modelo de Objetos del Negocio

4.9.3 Casos de uso del negocio

4.9.3.1 Introducción:

El modelo de casos de uso del negocio es un artefacto de la disciplina de Requisitos en la metodología RUP la cual estamos implementando.

Es el modelo de las funciones de negocio vistas desde la perspectivas de los actores externos permite situar al sistema en el contexto organizacional haciendo énfasis en los objetivos en este ámbito. Este modelo se representa con un Diagrama de Casos de Uso a través de estereotipos específicos para este modelo.

4.9.3.2 Propósito:

El propósito del modelo de casos de uso del negocio es mostrar con claridad los procesos que realizan cada actor en sus actividades diarias, en detalle se muestra la interacción realizada por el actor con los diferentes procesos que constituirá el proyecto.

4.9.3.3 Alcances:

- ✓ Describir el comportamiento de los procesos de negocio.
- ✓ Identificar y definir los objetos del negocio.

Caso de uso de negocio general

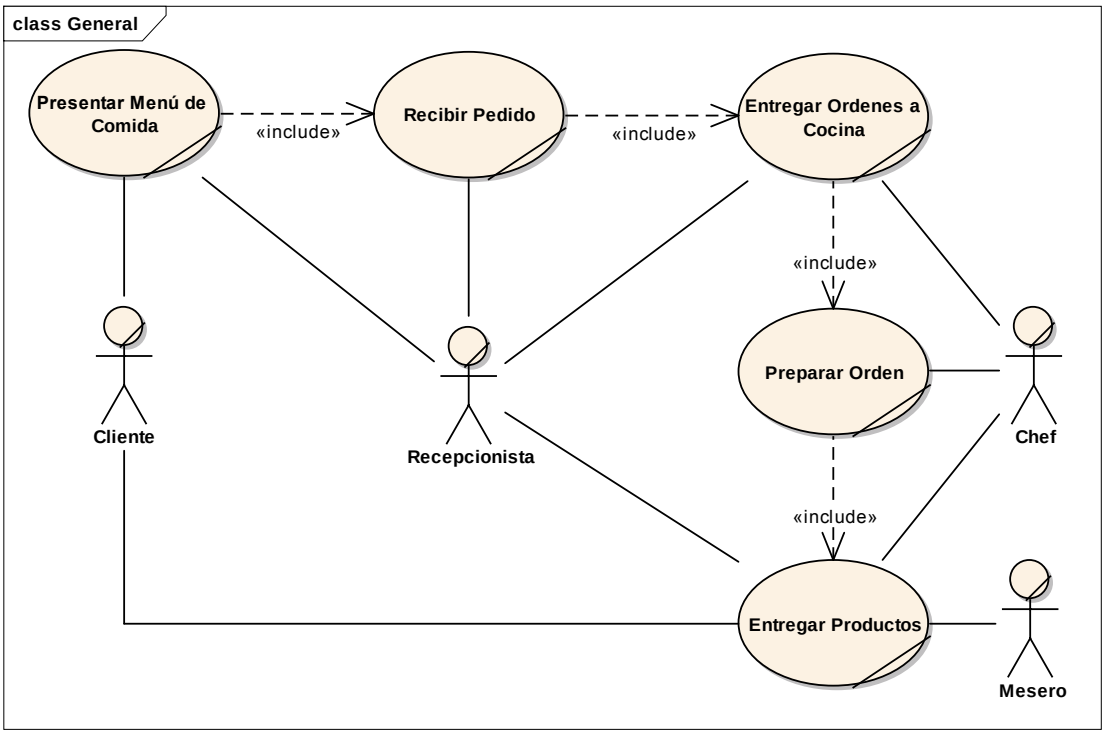


Figura 6. Caso de uso de negocio general

Presentar menú de comida

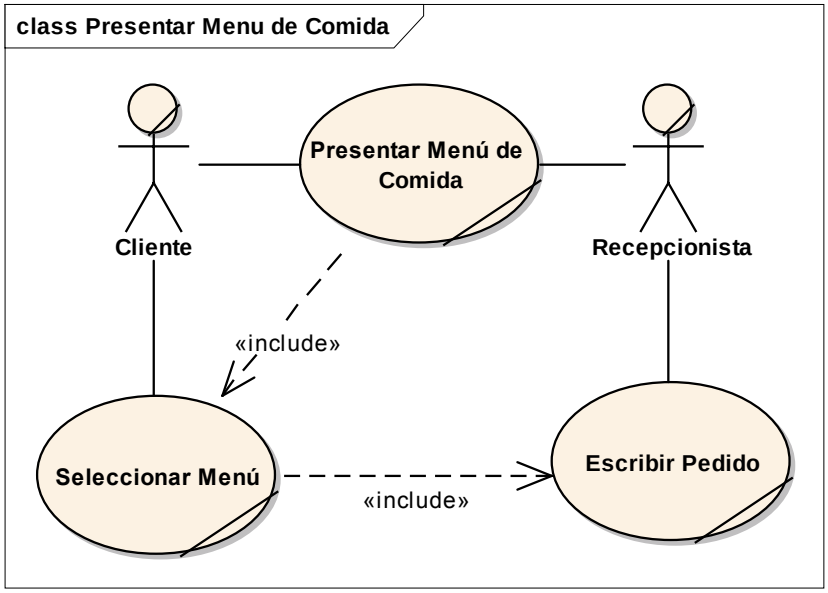


Figura 7. Caso de uso de negocio presentar menú de comida

Entregar órdenes a cocina

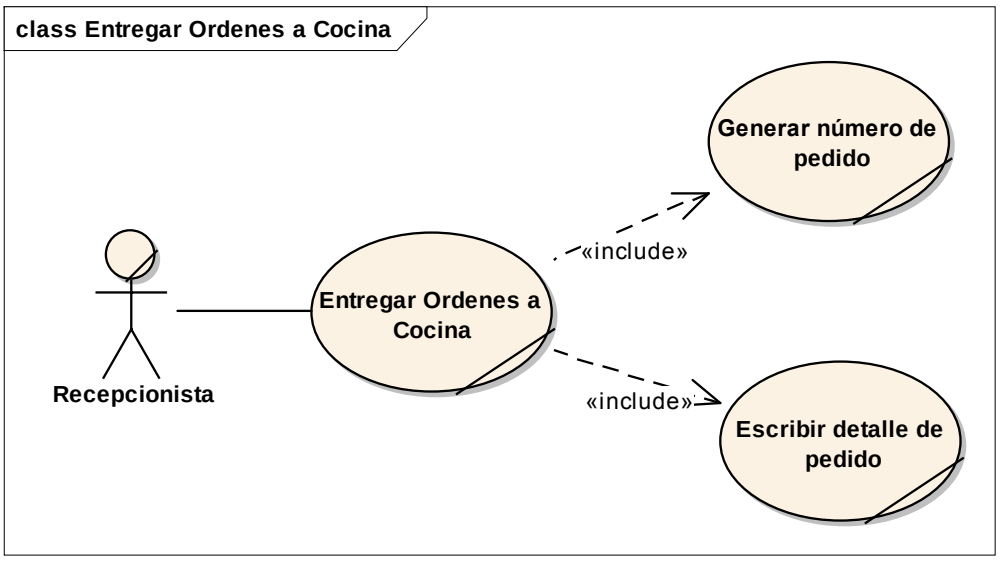


Figura 8. Caso de uso de negocio entregar órdenes a cocina

Preparar orden

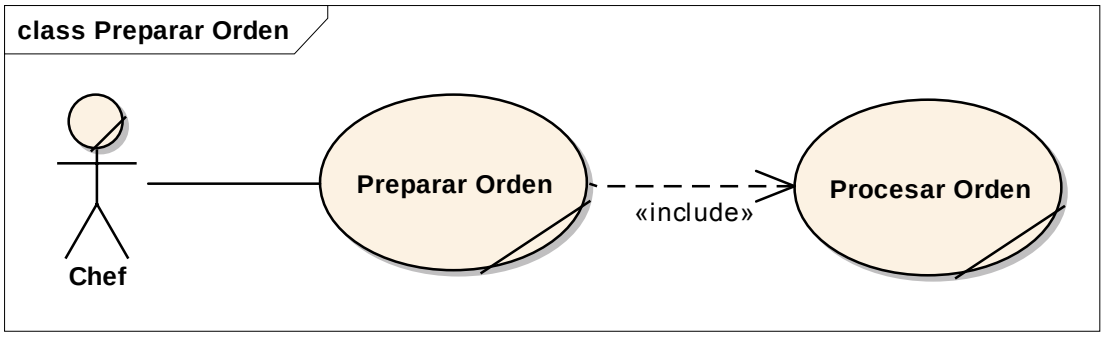


Figura 9. Caso de uso de negocio preparar orden

En tregar Productos

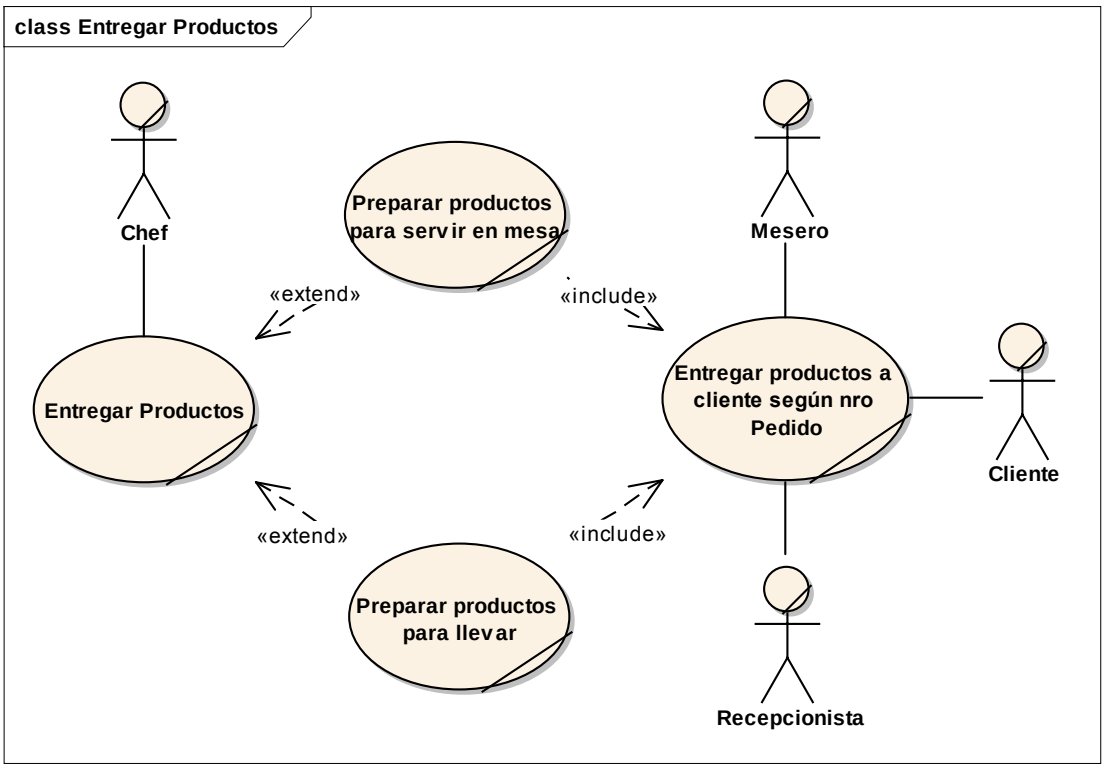


Figura 10. Modelo de Negocio Entregar Pedido

4.9.4 Modelo de casos de uso [32]

4.9.4.1 Introducción:

El modelo de casos de uso es un modelo del sistema que contiene actores, casos de uso y sus relaciones, describe lo que hace el sistema para cada tipo de usuario.

Es decir cada forma en que los actores usan el sistema se representa con uno caso de uso, los mismos son fragmentos de funcionalidad, especifican una secuencia de acciones que el sistema puede llevar a cabo interactuando con sus actores.

4.9.4.2 Propósito:

- ✓ Comprender la estructura y la dinámica del sistema deseado para la organización.
- ✓ Identificar posibles mejoras.

4.9.4.3 Alcance:

- ✓ Describe los procesos de sistema
- ✓ Identificar y definir los procesos del sistema según los objetivos de la organización
- ✓ Definir un caso de uso para cada procesos del sistema (el diagrama de casos de uso puede mostrar el contexto y los límites de la organización)

Caso de Uso General

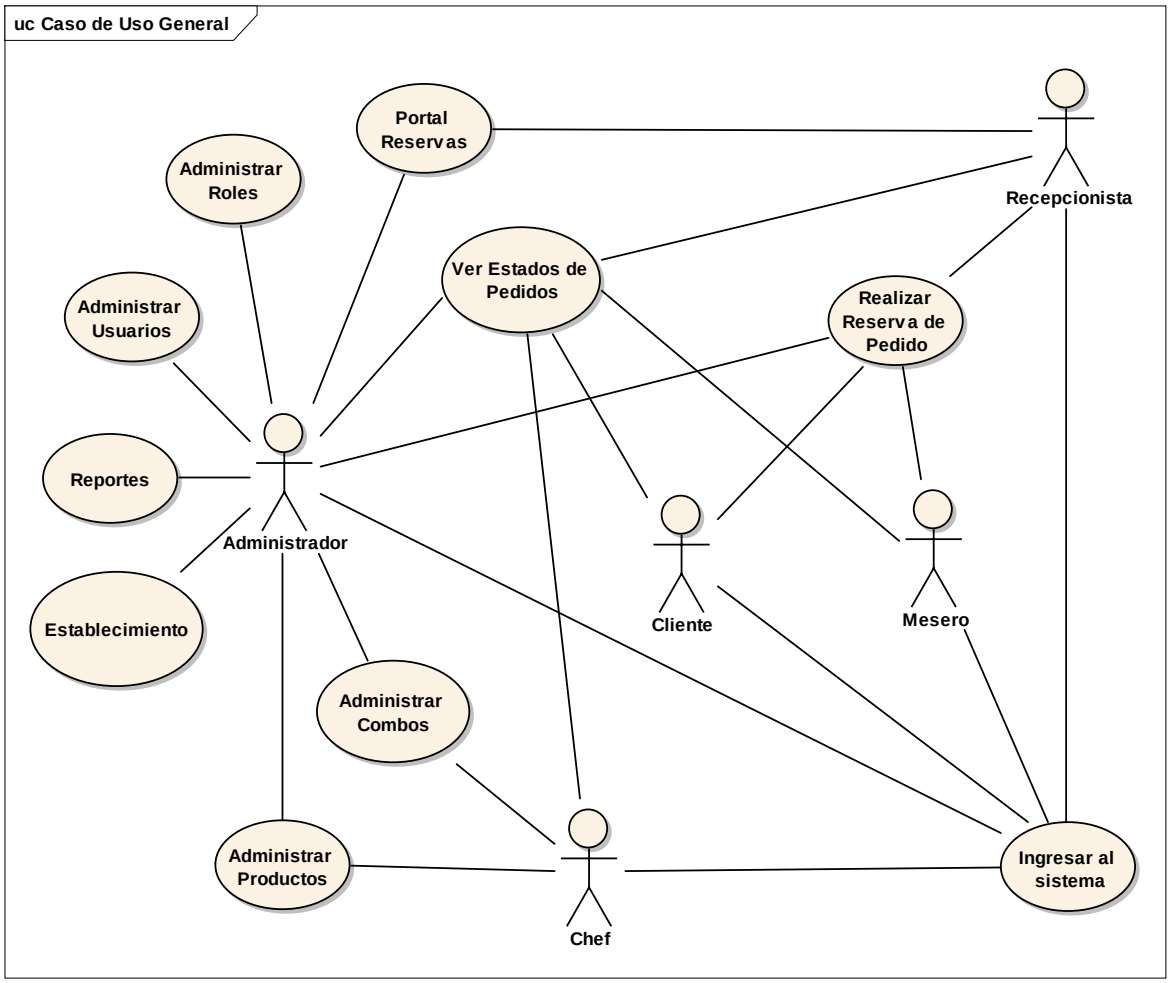


Figura 11. Caso de Uso General

Ingresar al Sistema

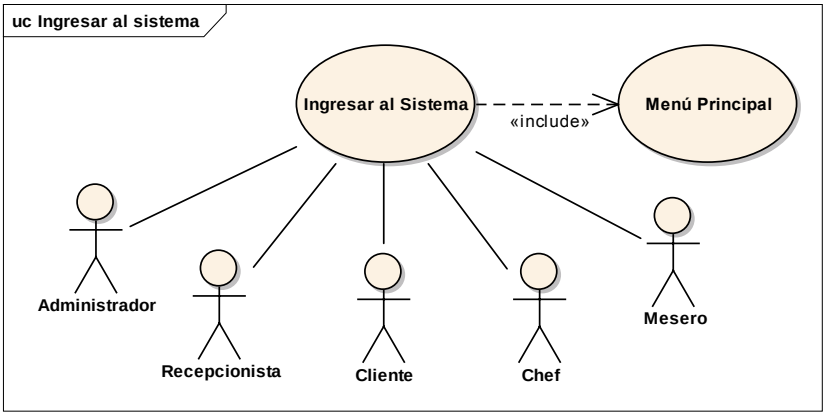


Figura 12. Caso de uso Ingresar al Sistema

Administrar Usuarios

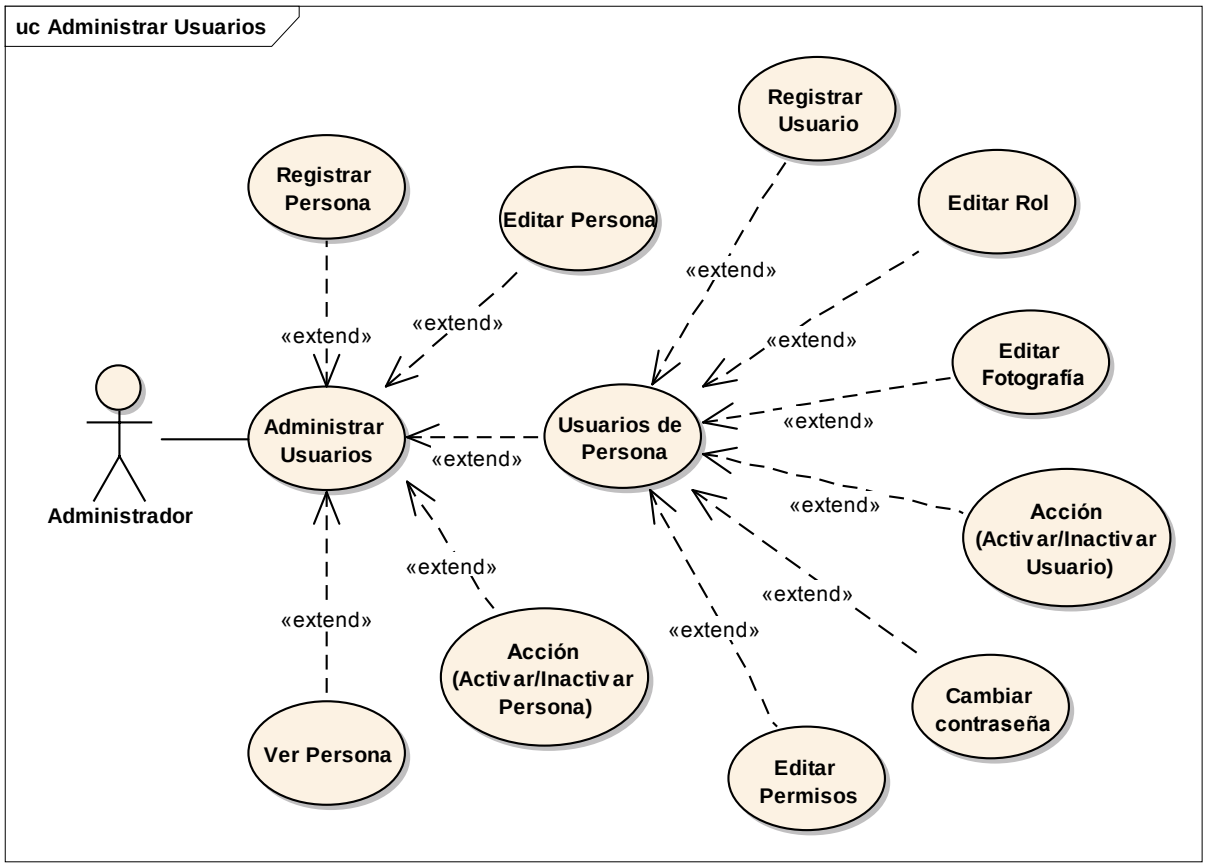


Figura 13. Caso de uso Administrar Usuario

Administrar Roles

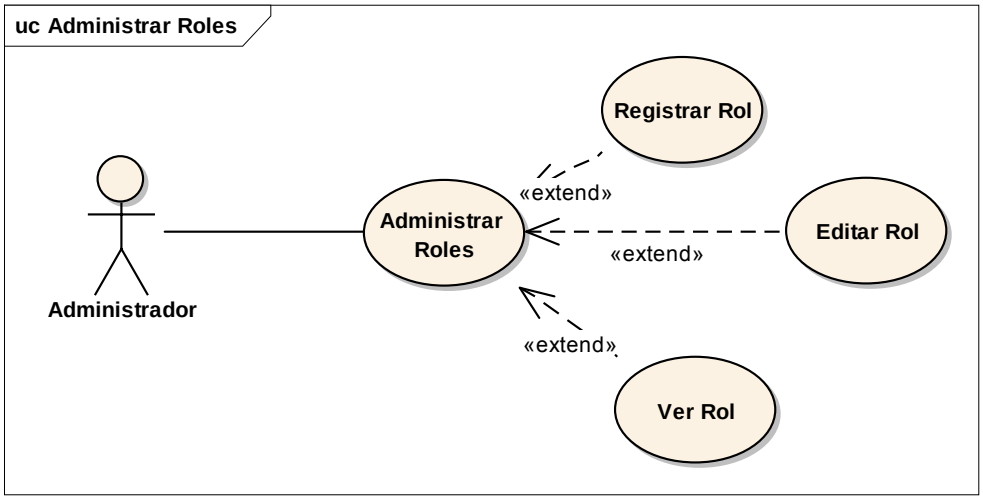


Figura 14. Caso de uso Administrar Roles

Administrar Productos

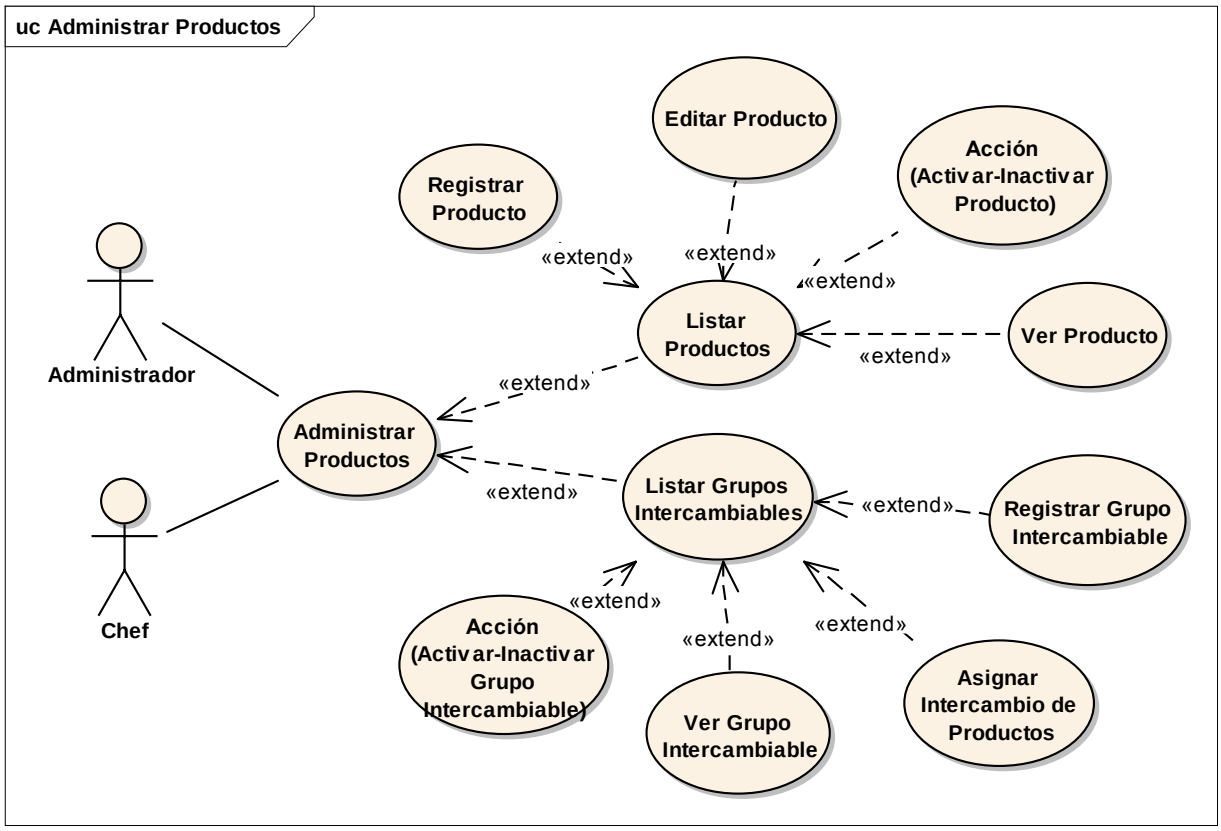


Figura 15. Caso de uso Administrar Productos

Administrar Combos

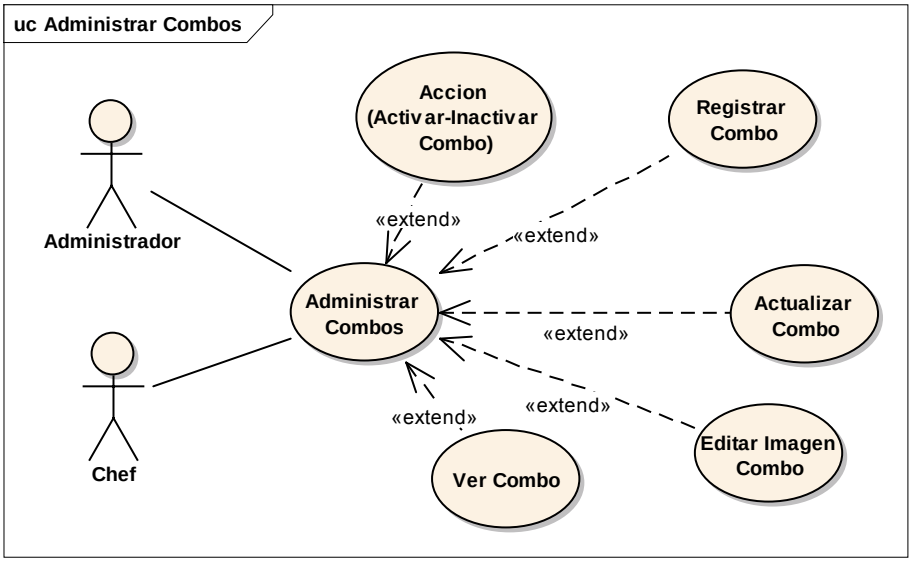


Figura 16. Caso de uso Administrar Combos

Realizar Reserva de Pedido

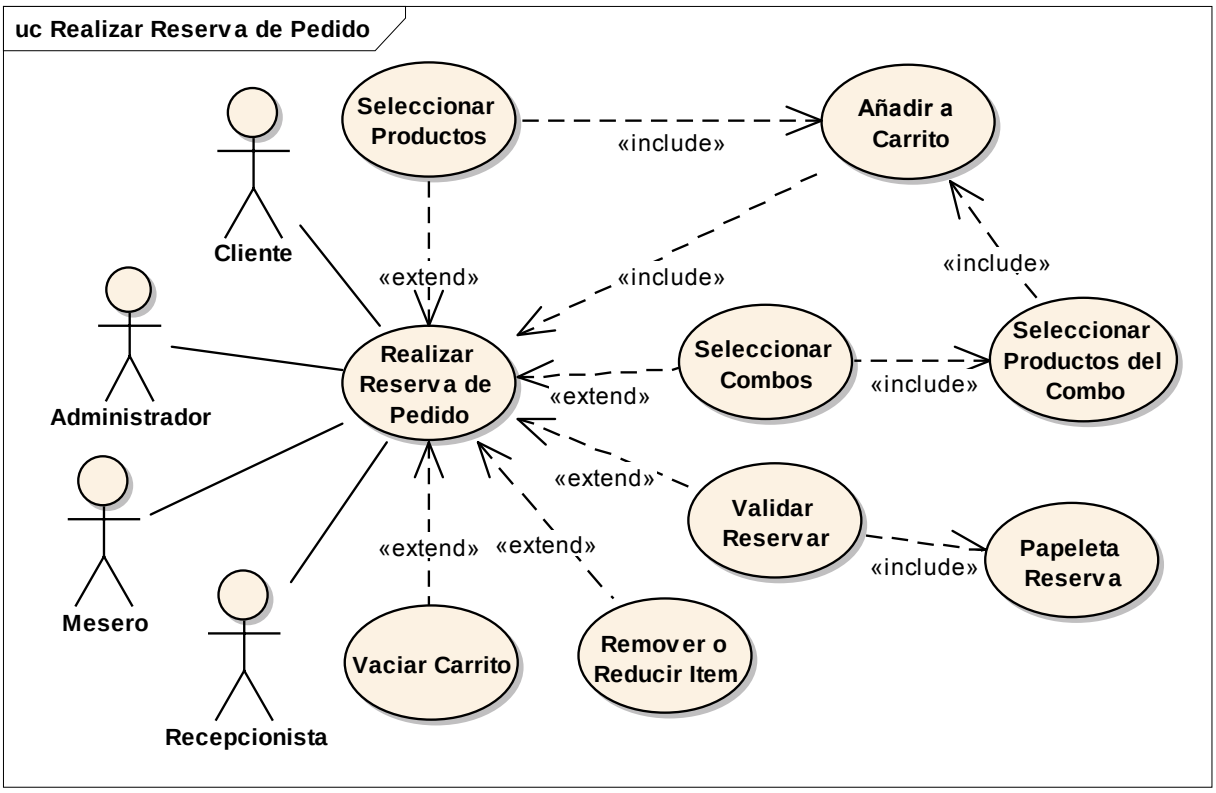


Figura 17. Caso de uso Realizar Reserva de Pedido

Ver Estados de Pedidos

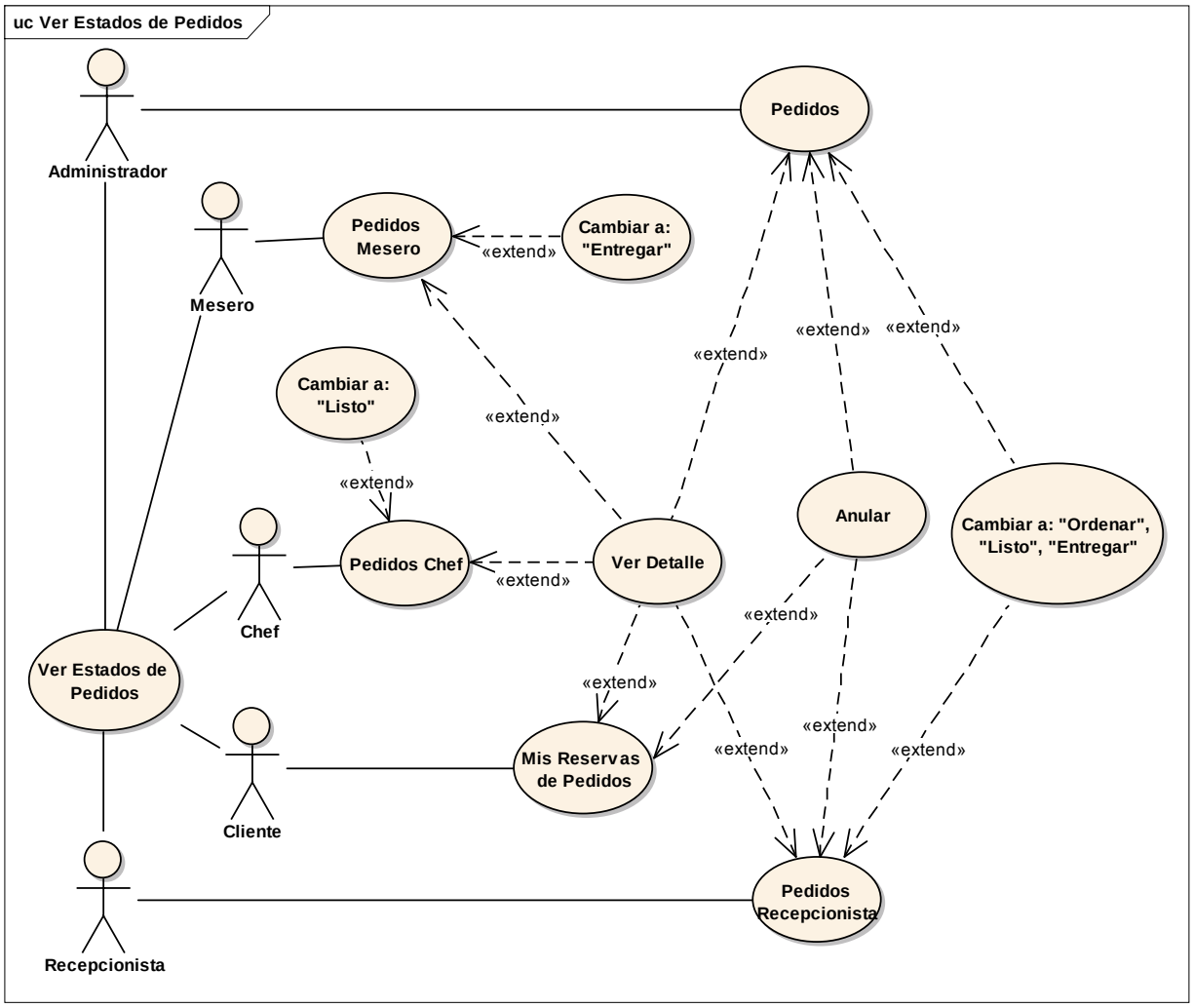


Figura 18. Caso de uso Ver Estados de Pedidos

Portal Reservas

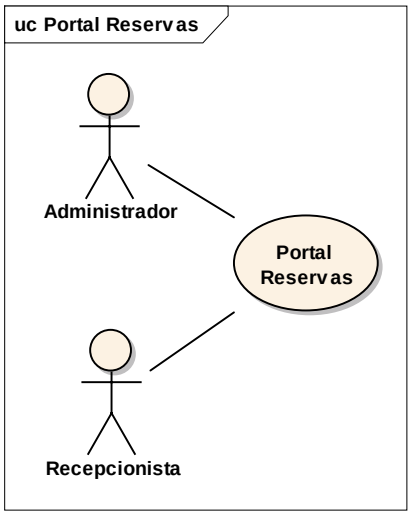


Figura 19. Caso de uso Portal Reservas

Establecimiento

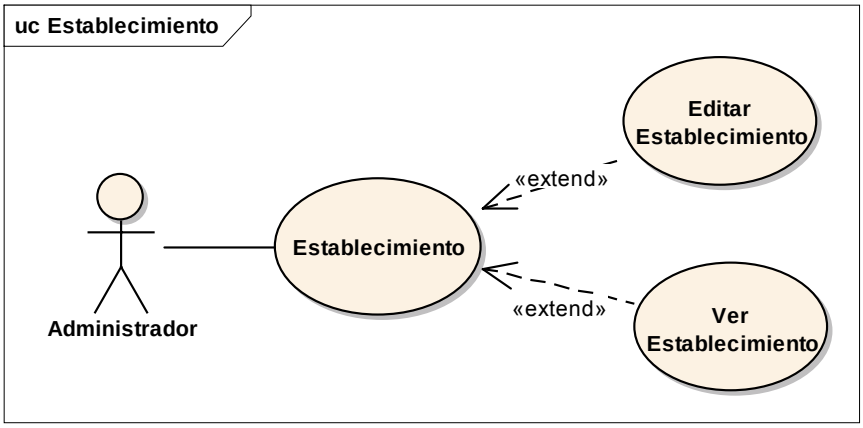


Figura 20. Caso de uso Establecimiento

Reportes

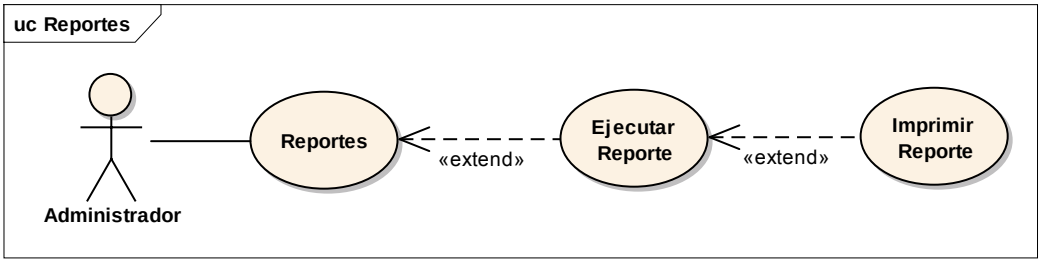


Figura 21. Caso de uso Reportes

4.9.5 Especificación de Casos de uso.

4.9.5.1 Introducción.

Las especificaciones de los casos de uso es una descripción detallada de los casos de uso del sistema.

4.9.5.2 Propósito.

- ✓ Comprender los casos de uso del sistema
- ✓ Describir específicamente cada caso de uso

4.9.5.3 Alcance.

- ✓ Describe los procesos internos de los casos de uso
- ✓ Detalla los flujos de cada caso de uso según lo establecido por la organización.

Caso de Uso: Ingresar al Sistema	
Descripción: Permite ingresar al sistema, este caso de uso tiene como función controlar el acceso y al mismo tiempo recuperar los permisos correspondientes al momento que la persona introduzca su usuario y clave en el sistema.	
Actores: Administrador, Recepcionista, Chef, Mesero, Cliente	
Precondiciones: El actor posee un login y clave	
Flujo Normal: 1. El actor introduce datos en el sistema. 2. El actor presiona botón ingresar. 3. El sistema valida sus datos. 4. Si los datos son correctos muestra el Menú Principal.	**Flujo Alternativo:** 5. Si el usuario y/o contraseña son incorrectos vuelve a mostrar la pantalla Ingresar al Sistema.
Post Condiciones: Ninguno	

Tabla 6. Especificación del CU Ingresar al Sistema

Caso de Uso: Menú Principal
Descripción: Este caso de uso tiene como función presentar el menú del sistema según sea el rol del usuario logueado.
Actores: Administrador, Recepcionista, Chef, Mesero, Cliente
Precondiciones: El actor debe estar logueado en el sistema.
Post Condiciones: Ninguno

Tabla 7. Especificación del CU Menú Principal

Caso de Uso: Administrar Usuarios	
Descripción: Este caso de uso tiene como objetivo administrar los usuarios del sistema, el sistema presenta una operación de búsqueda por CI, Nombres y Apellidos, un botón de "Registrar Persona", una lista de personas registradas en el sistema, botones "Editar", "Usuarios", "Ver" y "Acción" que direccionan a sus propias pantallas.	
Actores: Administrador	
Precondiciones: El actor debe estar logueado en el sistema	
Flujo Normal: 1. El actor selecciona la opción Administrar Usuarios. 2. El sistema muestra la lista de personas registradas en el sistema y las operaciones que puede realizar.	**Flujo Alternativo:** Ninguno
Post Condiciones: Ninguno	

Tabla 8. Especificación del CU Administrar Usuarios

Caso de Uso: Registrar Persona	
Descripción: Este caso de uso tiene como objetivo adicionar una nueva persona al sistema.	
Actores: Administrador	
Precondiciones: El actor debe haber presionado el botón "Registrar Persona" en la pantalla Administrar Usuarios / Listar Personas.	
Flujo Normal: 1. El sistema muestra el formulario. 2. El actor introduce datos al formulario. 3. Presiona Guardar, se valida datos del formulario. 4. El sistema valida los campos obligatorios vacíos. 5. El sistema guarda datos del formulario	**Flujo Alternativo:** 3. Por Cancelar, se regresa a la pantalla de Listar Personas. 4. Si existen campos obligatorios vacíos se regresa a la pantalla Registrar Persona remarcando los campos que faltan completar.
Post Condiciones: Ninguno	

Tabla 9. Especificación del CU Registrar Persona

Caso de Uso: Editar Persona	
Descripción: Este caso de uso tiene como objetivo editar los datos de una persona con excepción de la Cédula de Identidad que se encuentra deshabilitada.	
Actores: Administrador	
Precondiciones: El actor debe haber seleccionado el botón "Editar Persona" en la pantalla Administrar Usuarios / Listar Personas.	
Flujo Normal: 1. El sistema muestra el formulario con los campos permitidos para modificar. 2. El actor introduce datos al formulario 3. Presiona Guardar, se valida datos del formulario. 4. El sistema valida los campos obligatorios vacíos.	**Flujo Alternativo:** 3. Por Cancelar, se regresa a la pantalla de Listar Personas. 4. Si existen campos obligatorios vacíos se regresa a la pantalla Editar Persona

5. El sistema guarda datos del formulario	remarcando los campos que faltan completar.
Post Condiciones: Ninguno	

Tabla 10. Especificación del CU Editar Persona

Caso de Uso: Acción (Activar-Inactivar Persona)	
Descripción: Este caso de uso tiene como objetivo inactivar a una persona, es decir, se cambia el estado de la persona a inactivo.	
Actores: Administrador	
Precondiciones: El actor debe haber presionado el botón "Acción" en la fila correspondiente a la persona.	
Flujo Normal: 1. Se presenta al usuario un mensaje con la opción Aceptar o Cancelar. 2. Presiona Aceptar. 3. El sistema cambia el estado de la persona seleccionada a inactivo y regresa a la pantalla Listar Personas.	**Flujo Alternativo:** 2. Por Cancelar, regresa a la pantalla Listar Personas sin registrar cambio alguno.
Post Condiciones: Ninguno	

Tabla 11. Especificación del CU Acción (Activar-Inactivar Persona)

Caso de Uso: Ver Persona	
Descripción: Este caso de uso tiene como objetivo mostrar los datos completos de la persona seleccionada, además de los usuarios correspondientes a esa persona.	
Actores: Administrador	
Precondiciones: El actor debe haber presionado el botón "Ver" en la fila correspondiente a la persona.	
Flujo Normal: 1. Se presenta la pantalla Ver Persona	**Flujo Alternativo:** Ninguno
Post Condiciones: Ninguno	

Tabla 12. Especificación del CU Ver Persona

Caso de Uso: Usuarios de Persona	
Descripción: Este caso de uso tiene como objetivo mostrar la lista de usuarios pertenecientes a una persona.	
Actores: Administrador	
Precondiciones: El actor debe haber presionado el botón "Usuarios" en la fila correspondiente a la persona.	
Flujo Normal: 1. El actor selecciona la opción Usuarios de Persona. 2. El sistema muestra la lista de usuarios pertenecientes a una persona registrada en el sistema y las opciones que puede realizar.	**Flujo Alternativo:** Ninguno
Post Condiciones: Ninguno	

Tabla 13. Especificación del CU Usuarios de Persona

Caso de Uso: Registrar Usuario	
Descripción: Este caso de uso tiene como objetivo adicionar un nuevo usuario con su login y clave para que pasen a ser reconocidos como usuarios del sistema, además de asignar un rol al usuario.	
Actores: Administrador	
Precondiciones: El actor debe haber presionado el botón "Registrar Usuario".	
Flujo Normal: 1. El sistema muestra el formulario 2. El actor selecciona un rol para el usuario asignando los permisos por defecto de dicho rol. 3. El actor introduce datos al formulario. 4. Presiona Guardar, se valida datos del formulario.	**Flujo Alternativo:** 4. Por Cancelar, se regresa a la pantalla de Usuarios de Persona.

5. El sistema valida los campos obligatorios vacíos. 6. El sistema guarda datos del formulario	5. Si existen campos obligatorios vacíos se regresa a la pantalla Registrar Usuario remarcando los campos que faltan completar.
Post Condiciones: Ninguno	

Tabla 14. Especificación del CU Registrar Usuario

Caso de Uso: Editar Usuario	
Descripción: Este caso de uso tiene como objetivo editar los datos de un usuario	
Actores: Administrador	
Precondiciones: El actor debe haber seleccionado la opción Editar Usuario	
Flujo Normal: 1. El sistema muestra los campos permitidos para modificar. 2. El actor puede cambiar o mantener el rol del usuario. 3. El actor realiza los cambios correspondientes. 4. Presiona Guardar. 5. El sistema valida los campos requeridos. 6. El sistema guarda los cambios.	**Flujo Alternativo:** 4. Por Cancelar, se regresa a la pantalla de Usuarios de Persona. 5. Si existen campos requeridos vacíos se regresa a la pantalla Editar Usuario remarcando los campos que faltan completar.
Post Condiciones: Ninguno	

Tabla 15. Especificación del CU Editar Usuario

Caso de Uso: Cambiar Contraseña	
Descripción: Este caso de uso tiene como objetivo permitir el cambio de contraseña de un usuario.	
Actores: Administrador	
Precondiciones: El actor debe haber seleccionado la opción Cambiar Contraseña	
Flujo Normal: 1. El sistema muestra el login del usuario y solicita la contraseña anterior para poder realizar cambios. 2. El actor realiza los cambios correspondientes. 3. Presiona Guardar 4. El sistema valida los campos requeridos confirmando la contraseña anterior como la nueva. 5. El sistema guarda los cambios	**Flujo Alternativo:** 3. Por Cancelar, se regresa a la pantalla de Usuarios de Persona. 4. Si la contraseña anterior es incorrecta o si existen campos requeridos vacíos se regresa a la pantalla Cambiar Contraseña.
Post Condiciones: Ninguno	

Tabla 16. Especificación del CU Cambiar Contraseña

Caso de Uso: Permisos de Usuario	
Descripción: Este caso de uso tiene como objetivo mostrar el rol al cual corresponde el usuario más una lista de permisos sus asignables.	
Actores: Administrador	
Precondiciones: El actor debe haber seleccionado la opción Permisos de Usuario en la casilla correspondiente al usuario.	
Flujo Normal: 1. El actor selecciona la opción Permisos de Usuario. 2. El sistema muestra el rol y la lista de permisos modificables.	**Flujo Alternativo:**

3. Presiona Guardar. 4. El sistema guarda los cambios.	3. Por Cancelar, se regresa a la pantalla de Usuarios de Persona.
Post Condiciones: Ninguno	

Tabla 17. Especificación del CU Permisos de Usuario

Caso de Uso: Inactivar Usuario	
Descripción: Este caso de uso tiene como objetivo inactivar a un usuario, es decir, se cambia el estado del usuario a inactivo.	
Actores: Administrador	
Precondiciones: El actor debe haber seleccionado la opción Inactivar Usuario en la casilla correspondiente al usuario	
Flujo Normal: 1. Se presenta al usuario un mensaje con la opción Aceptar o Cancelar. 2. Presiona Aceptar. 3. El sistema inactiva al usuario seleccionado y regresa a la pantalla Usuarios de Persona.	**Flujo Alternativo:** 2. Presiona Cancelar. 3. Regresa a la pantalla Usuarios de Persona.
Post Condiciones: Ninguno	

Tabla 18. Especificación del CU Acción (Activar-Inactivar) Usuario

Caso de Uso: Administrar Roles	
Descripción: Este caso de uso tiene como objetivo administrar los roles de usuarios del sistema.	
Actores: Administrador	
Precondiciones: El actor debe estar logueado en el sistema.	
Flujo Normal: 1. El actor selecciona la opción Administrar Roles. 2. El sistema muestra la lista de roles del sistema y las operaciones que puede realizar.	**Flujo Alternativo:** Ninguno
Post Condiciones: Ninguno	

Tabla 19. Especificación del CU Administrar Roles

Caso de Uso: Registrar Rol	
Descripción: Este caso de uso tiene como objetivo adicionar un nuevo rol de usuario en el sistema.	
Actores: Administrador	
Precondiciones: El actor debe haber seleccionado la opción Registrar Rol	
Flujo Normal: 1. El sistema muestra el formulario. 2. El actor introduce datos al formulario. 3. El actor selecciona los permisos a los cuales el rol tendrá acceso. 4. Presiona Guardar, se valida datos del formulario. 5. El sistema valida los campos obligatorios vacíos. 6. El sistema guarda datos del formulario	**Flujo Alternativo:** 4. Por Cancelar, se regresa a la pantalla de Administrar Roles. 5. Si existen campos obligatorios vacíos se regresa a la pantalla Registrar Rol remarcando los campos que faltan completar.
Post Condiciones: Ninguno	

Tabla 20. Especificación del CU Registrar Rol

Caso de Uso: Editar Rol	
Descripción: Este caso de uso tiene como objetivo editar los datos de un rol de usuario del sistema.	
Actores: Administrador	
Precondiciones: El actor debe haber seleccionado la opción Editar Rol	
Flujo Normal: 1. El sistema muestra el nombre del rol con sus respectivos permisos y la lista de permisos modificables. 2. El actor modifica los permisos del rol. 3. Presiona Guardar.	**Flujo Alternativo:** 3. Por Cancelar, se regresa a la pantalla de Administrar Roles.

4. El sistema guarda datos.	
Post Condiciones: Ninguno	

Tabla 21. Especificación del CU Editar Rol

Caso de Uso: Ver Rol	
Descripción: Este caso de uso tiene como objetivo mostrar los datos y permisos del rol seleccionado.	
Actores: Administrador	
Precondiciones: El actor debe haber seleccionado la opción Ver Rol	
Flujo Normal: 1. Se presenta la pantalla Ver Rol	**Flujo Alternativo:** Ninguno
Post Condiciones: Ninguno	

Tabla 22. Especificación del CU Ver Rol

Caso de Uso: Administrar Productos	
Descripción: Este caso de uso tiene como objetivo administrar productos.	
Actores: Administrador, Chef	
Precondiciones: El actor debe estar logueado en el sistema	
Flujo Normal: 1. El actor selecciona la opción Administrar Productos 2. El sistema muestra despliega un submenú con las operaciones Listar Productos y Listar Grupos Intercambiables.	**Flujo Alternativo:** Ninguno
Pos Condiciones: Ninguno	

Tabla 23. Especificación del CU Administrar Productos

Caso de Uso: Listar Productos	
Descripción: Este caso de uso tiene como objetivo listar productos registrados en el sistema.	
Actores: Administrador, Chef	
Precondiciones: El actor debe haber seleccionado la opción Listar Productos	
Flujo Normal: 1. El actor selecciona la opción Listar Productos 2. El sistema muestra la lista de productos registrados en el sistema y las operaciones que puede realizar.	**Flujo Alternativo:** Ninguno
Pos Condiciones: Ninguno	

Tabla 24. Especificación del CU Listar Productos

Caso de Uso: Registrar Producto	
Descripción: Este caso de uso tiene como objetivo registrar un producto en el sistema	
Actores: Administrador, Chef	
Precondiciones: El actor debe haber seleccionado la opción Registrar Producto	
Flujo Normal: 1. El sistema muestra el formulario 2. El actor introduce datos al formulario 3. Presiona Guardar, se valida datos del formulario 4. El sistema valida los campos obligatorios vacios 5. El sistema guarda datos del formulario	**Flujo Alternativo:** 3. Por Cancelar, se regresa a la pantalla de Listar Productos 4. Si existen campos obligatorios vacios se regresa a la pantalla Administrar Productos remarcando los campos que faltan completar.
Pos Condiciones: Ninguno	

Tabla 25. Especificación del CU Registrar Producto

Caso de Uso: Modificar Producto	
Descripción: Este caso de uso tiene como objetivo modificar los datos de un producto registrado en el sistema.	
Actores: Administrador, Empleado	
Precondiciones: El actor debe haber seleccionado la opción Modificar Producto	
Flujo Normal: 1. El sistema muestra el formulario con los campos permitidos para modificar. 2. El actor introduce datos al formulario. 3. Presiona Guardar. 4. El sistema valida los campos obligatorios vacios 5. El sistema guarda datos del formulario	**Flujo Alternativo:** 3. Por Cancelar, se regresa a la pantalla de Listar Productos. 4. Si existen campos obligatorios vacios se regresa a la pantalla Listar Productos remarcando los campos que faltan completar.
Pos Condiciones: Ninguno	

Tabla 26. Especificación del CU Modificar Producto

Caso de Uso: Inactivar Producto	
Descripción: Este caso de uso tiene como objetivo inactivar un producto, es decir, se cambia el estado del producto a inactivo.	
Actores: Administrador, Chef	
Precondiciones: El actor debe haber seleccionado la opción Inactivar Producto en la casilla correspondiente al producto.	
Flujo Normal: 1. Se presenta al usuario un mensaje con la opción “Aceptar o Cancelar”. 2. Presiona Aceptar 3. El sistema inactiva el producto seleccionado.	**Flujo Alternativo:** 3. Presiona Cancelar 4. Regresa a la pantalla Listar Productos.
Pos Condiciones: Ninguno	

Tabla 27. Especificación del CU Acción (Activar – Inactivar) Producto

Caso de Uso: Ver Producto	
Descripción: Este caso de uso tiene como objetivo mostrar los datos del producto registrado en el sistema	
Actores: Administrador, Chef	
Precondiciones: El actor debe haber seleccionado la opción Ver Producto	
Flujo Normal: 1. Se presenta la pantalla Ver Producto	**Flujo Alternativo:** Ninguno
Pos Condiciones: Ninguno	

Tabla 28. Especificación del CU Ver Producto

Caso de Uso: Listar Grupos Intercambiables	
Descripción: Este caso de uso tiene como objetivo listar los grupos intercambiables registrados en el sistema.	
Actores: Administrador, Chef	
Precondiciones: El actor debe haber seleccionado la opción Listar Grupos Intercambiables	
Flujo Normal: 1. El actor selecciona la opción Listar Grupos Intercambiables. 2. El sistema muestra la lista de grupos registrados en el sistema y las operaciones que puede realizar.	**Flujo Alternativo:** Ninguno
Pos Condiciones: Ninguno	

Tabla 29. Especificación del CU Listar Grupos Intercambiables

Caso de Uso: Registrar Grupo Intercambiable	
Descripción: Este caso de uso tiene como objetivo registrar un grupo en el sistema	
Actores: Administrador, Chef	
Precondiciones: El actor debe haber seleccionado la opción Registrar Grupo Intercambiable	
Flujo Normal: 1. El sistema muestra el formulario 2. El actor introduce datos al formulario 3. Presiona Guardar, se valida datos del formulario 4. El sistema valida los campos obligatorios vacios 5. El sistema guarda datos del formulario	**Flujo Alternativo:** 3. Por Cancelar, se regresa a la pantalla de Listar Grupos Intercambiables. 4. Si existen campos obligatorios vacios se regresa a la pantalla Listar Grupos Intercambiables remarcando los campos que faltan completar.
Pos Condiciones: Ninguno	

Tabla 30. Especificación del CU Registrar Grupo Intercambiable

Caso de Uso: Asignar Intercambio de Productos	
Descripción: Este caso de uso tiene como objetivo asignar que productos podrán ser intercambiados entre sí dentro del grupo.	
Actores: Administrador, Chef	
Precondiciones: El actor debe haber seleccionado la opción Asginar Intercambiables	
Flujo Normal: 1. El sistema muestra los productos pertenecientes al grupo y las casillas de check para asignar. 2. El actor asigna los productos. 3. Presiona Asignar. 5. El sistema guarda la asignación.	**Flujo Alternativo:** 3. Por Cancelar, se regresa a la pantalla de Listar Grupos Intercambiables.
Pos Condiciones: Ninguno	

Tabla 31. Especificación del CU Asignar Intercambio de Productos

Caso de Uso: Ver Grupo	
Descripción: Este caso de uso tiene como objetivo mostrar los productos del grupo indicando cuales son intercambiables entre sí.	
Actores: Administrador, Chef	
Precondiciones: El actor debe haber seleccionado la opción Ver Grupo Intercambiable	
Flujo Normal: 1. Se presenta la pantalla Ver Grupo Intercambiable	**Flujo Alternativo:** Ninguno
Pos Condiciones: Ninguno	

Tabla 32. Especificación del CU Ver Grupo Intercambiable

Caso de Uso: Administrar Combos	
Descripción: Este caso de uso tiene como objetivo armar los combos con los productos registrados en el sistema.	
Actores: Administrador, Chef	
Precondiciones: El actor debe estar logueado en el sistema	
Flujo Normal: 1. El actor selecciona la opción Administrar Combos. 2. El sistema muestra la lista de combos y las operaciones que puede realizar.	**Flujo Alternativo:** Ninguno
Pos Condiciones: Ninguno	

Tabla 33. Especificación del CU Administrar Combos

Caso de Uso: Registrar Combo	
Descripción: Este caso de uso tiene como objetivo registrar un nuevo combo en el sistema.	
Actores: Administrador, Chef	
Precondiciones: El actor debe haber seleccionado la opción Registrar Combo	
Flujo Normal: 1. El sistema muestra el formulario 2. El actor introduce datos al formulario. 3. Presiona Guardar. 4. El sistema valida los campos obligatorios vacios 5. El sistema guarda datos del formulario	**Flujo Alternativo:** 3. Por Cancelar, se regresa a la pantalla de Administrar Combos. 4. Si existen campos obligatorios vacios se regresa a la pantalla Administrar Combos remarcando los campos que faltan completar.
Pos Condiciones: Ninguno	

Tabla 34. Especificación del CU Registrar Combo

Caso de Uso: Actualizar Combo	
Descripción: Este caso de uso tiene como objetivo actualizar los datos de un combo registrado en el sistema.	
Actores: Administrador, Chef	
Precondiciones: El actor debe haber seleccionado la opción Actualizar Combo	
Flujo Normal: 1. El sistema muestra el formulario con los campos permitidos para modificar junto con sus correspondientes productos. 2. El actor introduce datos al formulario y/o modifica los productos del combo a actualizar. 3. Presiona Guardar. 4. El sistema valida los campos obligatorios vacios	**Flujo Alternativo:** 3. Por Cancelar, se regresa a la pantalla de Administrar Combos. 4. Si existen campos obligatorios vacios se regresa a la pantalla Administrar Combo

5. El sistema guarda datos del formulario	remarcando los campos que faltan completar.
Pos Condiciones: Ninguno	

Tabla 35. Especificación del CU Actualizar Combo

Caso de Uso: Inactivar Combo	
Descripción: Este caso de uso tiene como objetivo eliminar un menú, es decir, se cambia el estado del combo a inactivo.	
Actores: Administrador	
Precondiciones: El actor debe haber seleccionado la opción Eliminar Combo en la casilla correspondiente al producto.	
Flujo Normal: 1. Se presenta al usuario un mensaje con la opción 2. Aceptar o Cancelar 3. Presiona Aceptar 4. El sistema elimina el menú seleccionado	**Flujo Alternativo:** 3. Presiona Cancelar 4. Regresa a la pantalla Administrar Combo
Pos Condiciones: Ninguno	

Tabla 36. Especificación del CU Inactivar Combo

Caso de Uso: Ver Combo	
Descripción: Este caso de uso tiene como objetivo mostrar los datos del combo registrados en el sistema.	
Actores: Administrador	
Precondiciones: El actor debe haber seleccionado la opción Ver Combo	
Flujo Normal: 1. Se presenta la pantalla Ver Combo	**Flujo Alternativo:** Ninguno
Pos Condiciones: Ninguno	

Tabla 37. Especificación del CU Ver Combo

Caso de Uso: Realizar Reserva de Pedido	
Descripción: Este caso de uso permite visualizar el carrito del usuario y las opciones de combos y productos disponibles del restaurante.	
Actores: Administrador, Chef, Recepcionista, Mesero y Cliente.	
Precondiciones: El actor debe estar logueado en el sistema	
Flujo Normal: 1. El actor elige la opción Realizar Reserva de Pedido. 2. El sistema muestra el monto total de la reserva en Bs., el carrito, el menú de comida y las operaciones que puede realizar.	**Flujo Alternativo:** Ninguno
Pos Condiciones: Ninguno	

Tabla 38. Realizar Especificación del CU Realizar Reserva de Pedido

Caso de Uso: Seleccionar Combos	
Descripción: Este caso de uso tiene como objetivo listar los combos disponibles para ser añadidos al carrito.	
Actores: Administrador, Chef, Recepcionista, Mesero y Cliente.	
Precondiciones: El actor debe haber seleccionado la opción Combos.	
Flujo Normal: 1. El sistema muestra el precio total del carrito y la lista de combos disponibles. 2. El actor selecciona un combo.	**Flujo Alternativo:** 2. El actor selecciona la opción “Realizar Reserva de Pedido” para regresar a la pantalla anterior.
Pos Condiciones: Ninguno	

Tabla 39. Especificación del CU Seleccionar Combos

Caso de Uso: Seleccionar Productos de Combo	
Descripción: Este caso de uso tiene como objetivo listar los productos del combo seleccionado con las opciones de intercambio entre los productos del mismo grupo, para luego ser añadidos al carrito.	
Actores: Administrador, Chef, Recepcionista, Mesero y Cliente.	
Precondiciones: El actor debe haber seleccionado un Combo de la pantalla "Seleccionar Combos".	
Flujo Normal: 1. El sistema muestra el descuento del combo y la lista de productos intercambiables en sus diferentes grupos. 2. El actor selecciona la cantidad de combos que desea añadir y lo productos que son intercambiables entre sí pertenecientes al mismo grupo. 3. Presiona "Añadir". 4. Añade el combo al carrito con sus respectivos productos seleccionados y regresa a la pantalla "Realizar Reserva de Pedido".	**Flujo Alternativo:** 2. El actor selecciona la opción "Realizar Reserva de Pedido" para ver el carrito. 3. Presiona "Cancelar". 4. Regresa a la pantalla "Combos".
Pos Condiciones: Ninguno	

Tabla 40. Especificación del CU Seleccionar Productos de Combo

Caso de Uso: Seleccionar Productos	
Descripción: Este caso de uso tiene como objetivo seleccionar un producto para ser añadido al carrito.	
Actores: Administrador, Chef, Recepcionista, Mesero y Cliente.	
Precondiciones: El actor debe haber seleccionado la opción Seleccionar Producto.	
Flujo Normal: 1. El sistema muestra el precio total del carrito y la lista de productos disponibles. 2. El actor selecciona un producto.	**Flujo Alternativo:** 2. El actor selecciona la opción "Realizar

3. Añade el producto al carrito y regresa a la pantalla “Realizar Reserva de Pedido”	Reserva de Pedido” para regresar a la pantalla ver el carrito.
Pos Condiciones: Ninguno	

Tabla 41. Especificación del CU Seleccionar Productos

Caso de Uso: Reservar	
Descripción: Este caso de uso tiene como objetivo mostrar el detalle del carrito y permitir la selección de las características del pedido como ser hora de entrega y si es para llevar o servirse en el local.	
Actores: Administrador, Chef, Recepcionista, Mesero y Cliente.	
Precondiciones: El actor debe haber seleccionado la opción “Reservar”.	
Flujo Normal: 1. El sistema muestra el detalle del carrito y las casillas de selección como parte de las características del pedido. 2. El actor define la información del pedido. 3. Presiona ”Realizar Reserva” 4. El sistema registra la reserva y genera una papeleta de reserva con los datos respectivos.	**Flujo Alternativo:**
Pos Condiciones: Ninguno	

Tabla 42. Especificación del CU Reservar

Caso de Uso: Vaciar Carrito	
Descripción: Este caso de uso tiene como objetivo vaciar el carrito.	
Actores: Administrador, Chef, Recepcionista, Mesero y Cliente.	
Precondiciones: El actor debe haber seleccionado la opción “Vaciar Carrito”	
Flujo Normal: 1. El sistema vacía el carrito.	**Flujo Alternativo:**
Pos Condiciones: Ninguno	

Tabla 43. Especificación del CU Vaciar Carrito

Caso de Uso: Ver Estados de Pedidos	
Descripción: Este caso de uso tiene como objetivo administrar los pedidos mostrando los estados en el que se encuentra cada uno.	
Actores: Administrador, Chef, Recepcionista, Mesero y Cliente.	
Precondiciones: El actor debe estar logueado en el sistema	
Flujo Normal: 1. El actor selecciona la opción Ver Estados de Pedidos. 2. El sistema muestra el listado de pedidos realizados durante todo el día y las operaciones que puede realizar.	**Flujo Alternativo:** Ninguno
Pos Condiciones: Ninguno	

Tabla 44. Especificación del CU Ver Estados de Pedidos

Caso de Uso: Ver Detalle Pedido	
Descripción: Este caso de uso tiene como objetivo mostrar el detalle del pedido seleccionado.	
Actores: Administrador, Chef, Recepcionista, Mesero y Cliente.	
Precondiciones: El actor debe haber seleccionado la opción “Ver Detalle Pedido”	
Flujo Normal: 1. El sistema muestra el detalle del pedido con todos sus datos más los combos y/o productos seleccionados.	**Flujo Alternativo:** Ninguno
Pos Condiciones: Ninguno	

Tabla 45. Especificación del CU Ver Detalle Pedido

Caso de Uso: Anular Pedido	
Descripción: Este caso de uso tiene como objetivo anular un pedido seleccionado.	
Actores: Administrador, Chef, Recepcionista, Mesero y Cliente.	
Precondiciones: El actor debe haber seleccionado la opción "Anular Pedido", es necesario que el estado del pedido se encuentre en "Reservado" caso contrario no podrá anularse.	
Flujo Normal: 1. Se presenta al usuario un mensaje con la opción "¿Anular Pedido?". 2. Presiona "Aceptar" 3. El sistema anula el pedido cambiando su estado a "Anulado" y regresa a la pantalla "Ver Estado Pedido".	**Flujo Alternativo:** Ninguno 2. Presiona "Cancelar" 3. No se realiza ninguna acción y regresa a la pantalla "Ver Estado Pedido".
Pos Condiciones: Ninguno	

Tabla 46. Especificación del CU Anular Pedido

Caso de Uso: Ordenar Pedido	
Descripción: Este caso de uso tiene como objetivo ordenar un pedido seleccionado, puede ser accionado manualmente o automáticamente por el sistema con una anticipación de 30 min antes de la hora de entrega especificada en el pedido.	
Actores: Sistema, Administrador, Chef, Recepcionista, Mesero y Cliente.	
Precondiciones: El actor debe haber seleccionado la opción "Ordenar Pedido" y el estado del pedido debe estar en "Reservado".	
Flujo Normal: 1. El sistema o el actor cambia el estado del pedido a "Ordenado" y regresa a la pantalla "Ver Estados de Pedidos".	**Flujo Alternativo:** Ninguno
Pos Condiciones: Ninguno	

Tabla 47. Especificación del CU Ordenar Pedido

Caso de Uso: Pedido Listo	
Descripción: Este caso de uso tiene como objetivo indicar que el pedido ya está listo para ser entregado.	
Actores: Administrador, Chef, Recepcionista y Mesero.	
Precondiciones: El actor debe haber seleccionado la opción "Listo" y el estado del pedido debe estar en "Ordenado".	
Flujo Normal: 1. El sistema cambia el estado del pedido a "Listo" y regresa a la pantalla "Ver Estados de Pedidos".	**Flujo Alternativo:** Ninguno
Pos Condiciones: Ninguno	

Tabla 48. Especificación del CU Pedido Listo

Caso de Uso: Entregar Pedido	
Descripción: Este caso de uso tiene como objetivo indicar que el pedido ya se entregó al cliente.	
Actores: Administrador, Chef, Recepcionista y Mesero.	
Precondiciones: El actor debe haber seleccionado la opción "Entregar" y el estado del pedido debe estar en "Listo".	
Flujo Normal: 1. El sistema cambia el estado del pedido a "Entregado" y regresa a la pantalla "Ver Estados de Pedidos". 2. El sistema genera una papeleta de entrega.	**Flujo Alternativo:** Ninguno
Pos Condiciones: Ninguno	

Tabla 49. Especificación del CU Entregar Pedido

Caso de Uso: Portal Reservas	
Descripción: Este caso de uso tiene como objetivo actualizar el horario de atención del restaurante, el estado del portal y un mensaje de efectivización de reserva.	
Actores: Administrador, Chef, Recepcionista, Mesero y Cliente.	
Precondiciones: El actor debe haber seleccionado la opción "Vaciar Carrito"	
Flujo Normal: 1. El sistema vacía el carrito.	**Flujo Alternativo:**
Pos Condiciones: Ninguno	

Tabla 50. Especificación del CU Portal Reservas

Caso de Uso: Establecimiento	
Descripción: Este caso de uso tiene como objetivo administrar el establecimiento.	
Actores: Administrador	
Precondiciones: El actor debe estar logueado en el sistema	
Flujo Normal: 1. El actor selecciona la opción Establecimiento. 2. El sistema muestra el establecimiento registrado en el sistema y las operaciones que puede realizar.	**Flujo Alternativo:** Ninguno
Pos Condiciones: Ninguno	

Tabla 51. Especificación del CU Establecimiento

Caso de Uso: Modificar Establecimiento	
Descripción: Este caso de uso tiene como objetivo modificar los datos del establecimiento registrado en el sistema.	
Actores: Administrador	
Precondiciones: El actor debe haber seleccionado la opción Modificar Establecimiento	
Flujo Normal: 1. El sistema muestra el formulario con los campos para modificar. 2. El actor introduce datos al formulario. 3. Presiona Guardar.	**Flujo Alternativo:** 3. Por Cancelar, se regresa a la pantalla de

4. El sistema valida los campos obligatorios vacios 5. El sistema guarda datos del formulario	Establecimiento. 4. Si existen campos obligatorios vacios se regresa a la pantalla Establecimiento remarcando los campos que faltan completar.
Pos Condiciones: Ninguno	

Tabla 52. Especificación del CU Modificar Establecimiento

Caso de Uso: Ver Establecimiento	
Descripción: Este caso de uso tiene como objetivo mostrar los datos del establecimiento registrado en el sistema	
Actores: Administrador	
Precondiciones: El actor debe haber seleccionado la opción Ver Establecimiento	
Flujo Normal: 1. Se presenta la pantalla Ver Producto	**Flujo Alternativo:** Ninguno
Pos Condiciones: Ninguno	

Tabla 53. Especificación del CU Ver Establecimiento

Caso de Uso: Reportes	
Descripción: Este caso de uso tiene como objetivo mostrar la información solicitada por el usuario.	
Actores: Administrador	
Precondiciones: El actor debe haber seleccionado la opción Reportes	
Flujo Normal: 1. Se presenta la pantalla Reportes 2. El sistema muestra las opciones de parametrización del reporte.	**Flujo Alternativo:** Ninguno
Pos Condiciones: Ninguno	

Tabla 54. Especificación del CU Reportes

4.10 Modelo de análisis y diseño

4.10.1 Diagrama de secuencia

4.10.1.1 Introducción

El diagrama de Secuencia es uno de los diagramas más efectivos para modelar interacción entre objetos de un sistema. Un diagrama de secuencia se modela para cada caso de uso. Mientras que el diagrama de Caso de uso permite el modelado de una vista del negocio del escenario, el diagrama de secuencia contiene detalles de implementación del escenario, incluyendo los objetos y clases que se usan para implementar el escenario y mensajes pasados entre los objetos.

Un diagrama de secuencia muestra los objetos que intervienen en el escenario con líneas discontinuas verticales, y los mensajes pasados entre los objetos como vectores horizontales. Los mensajes se dibujan cronológicamente desde la parte superior del diagrama a la parte superior del diagrama a la parte inferior; la distribución horizontal de los objetos es arbitraria.

4.10.1.2 Propósito

- ✓ Comprender la dinámica del sistema deseado para la organización.
- ✓ Identificar clases de análisis y diseño.

4.10.1.3 Alcances

- ✓ Describe la dinámica de sistema en el tiempo de vida de las clases u objetos
- ✓ Definir un diagrama de secuencia para cada caso de uso del sistema

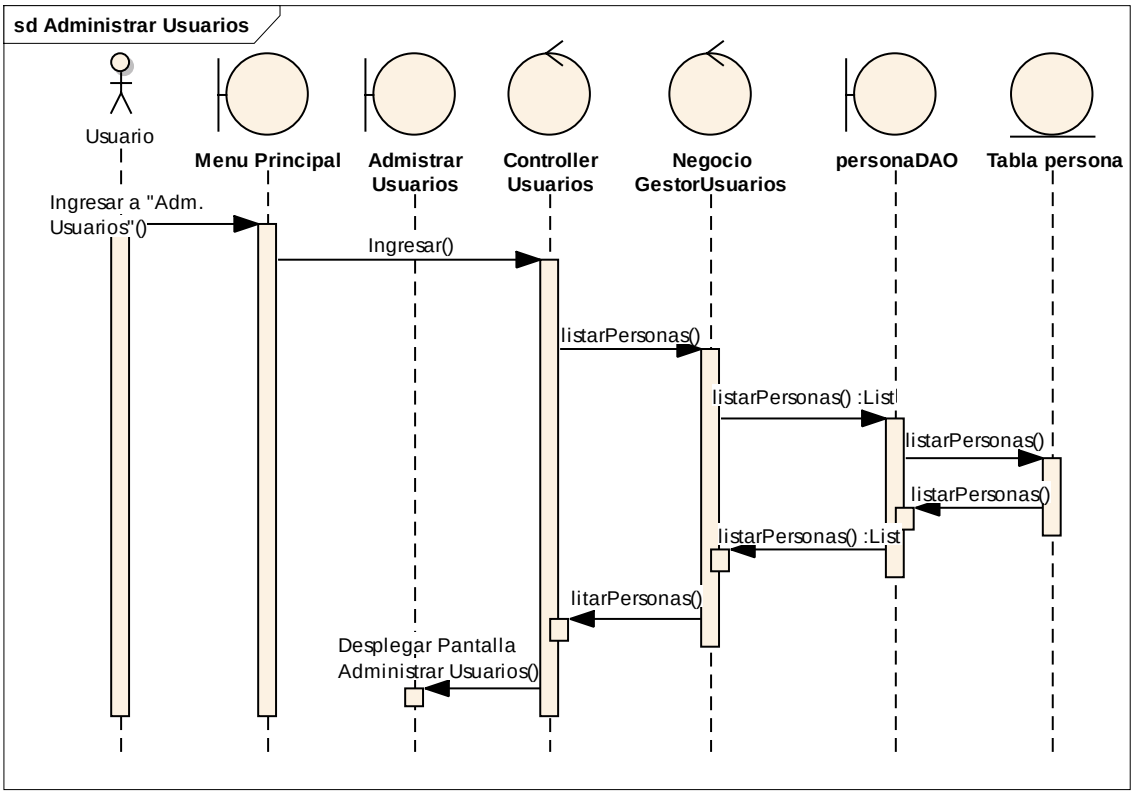


Figura 22. Diagrama de Secuencia Administrar Usuarios

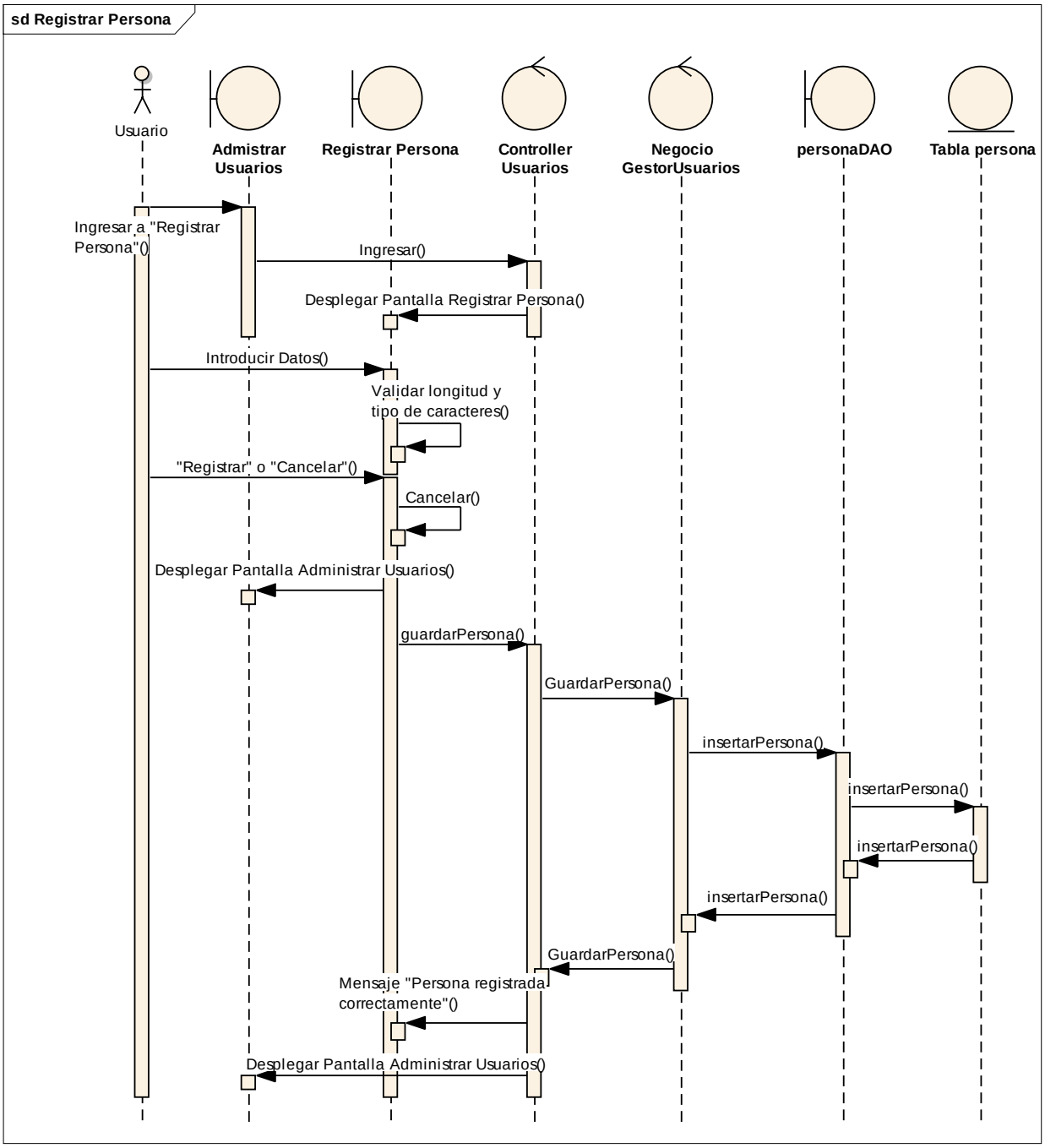


Figura 23. Diagrama de Secuencia Registrar Persona

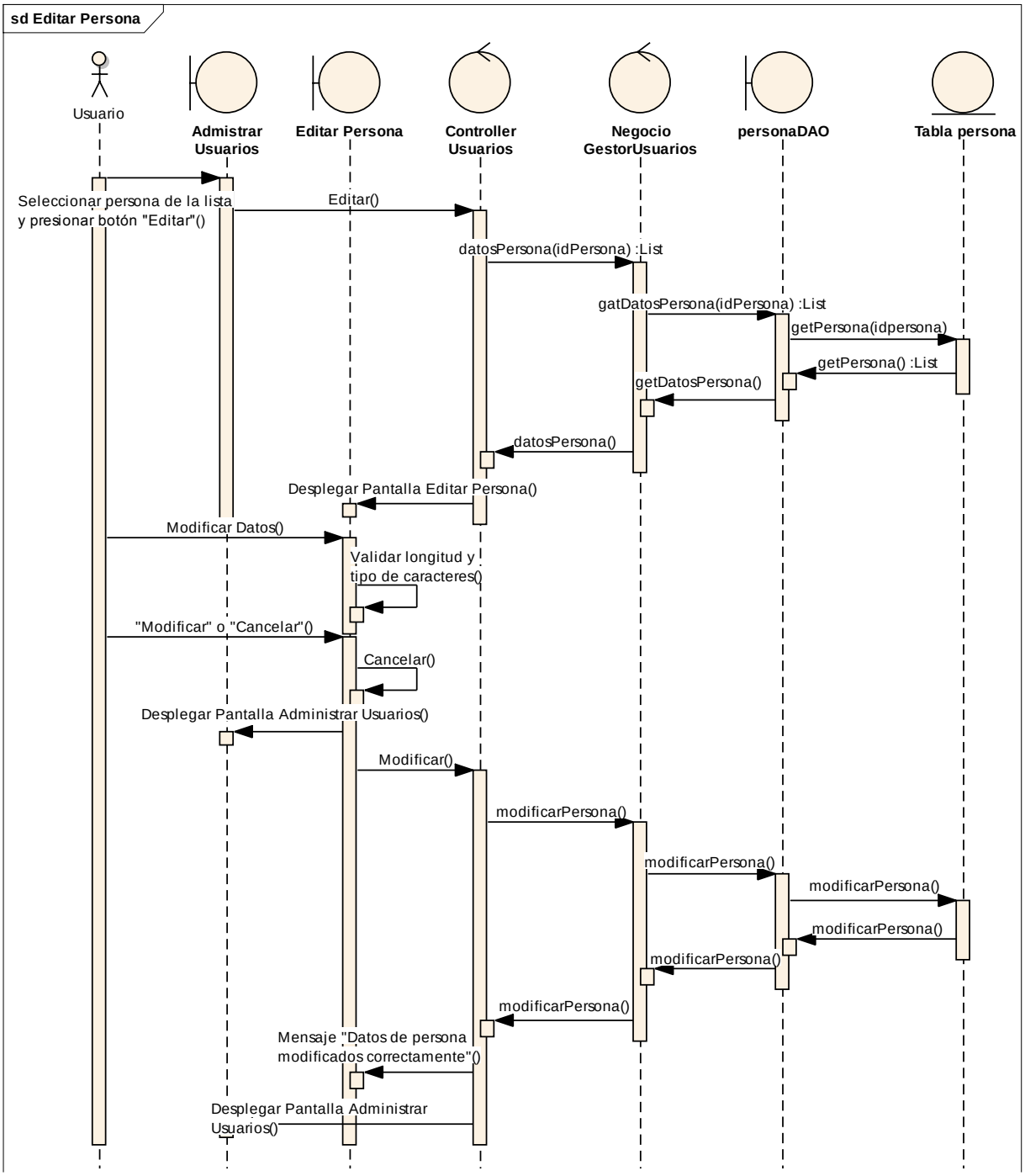


Figura 24. Diagrama de Secuencia Editar Persona

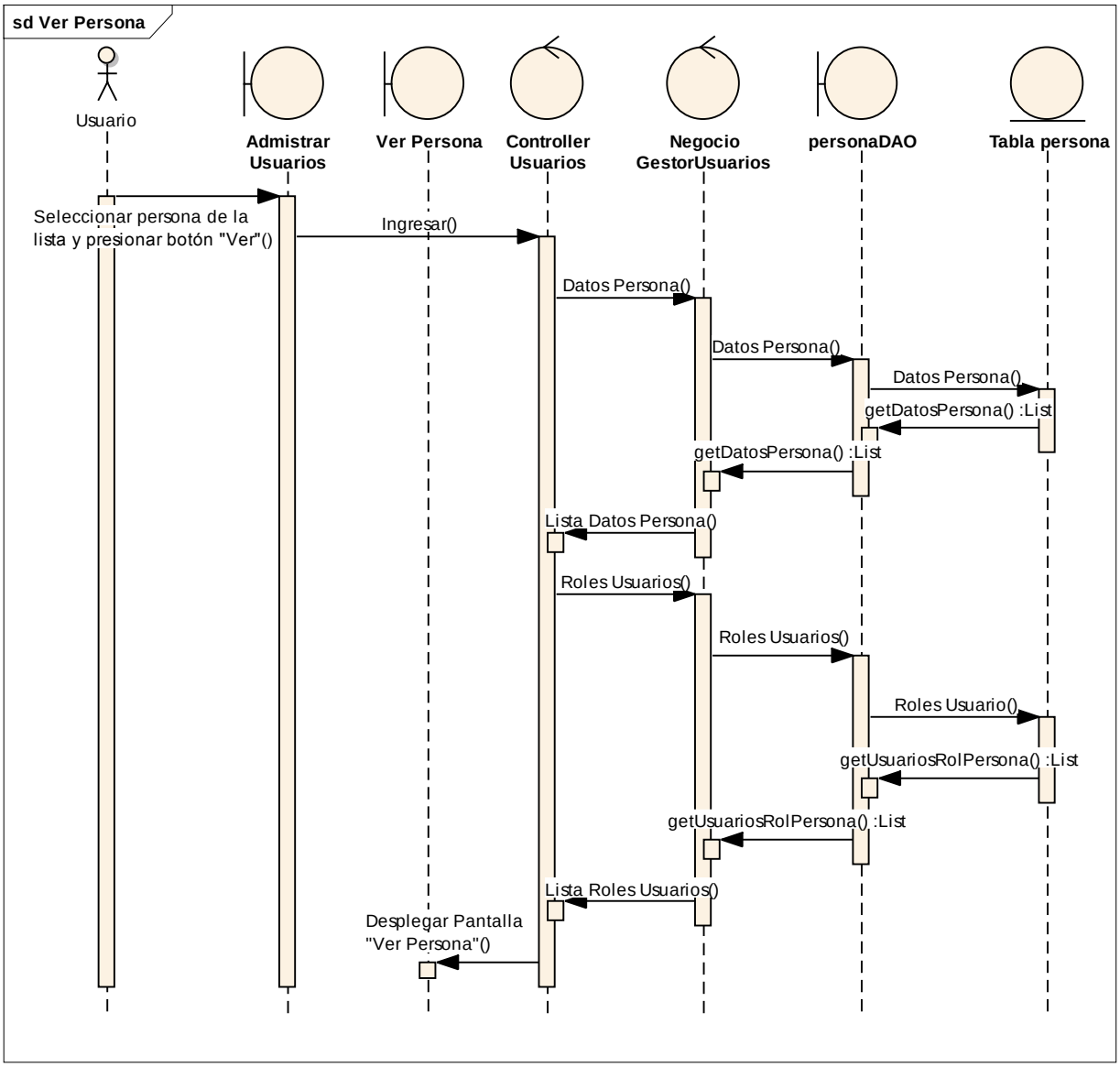


Figura 25. Diagrama de Secuencia Ver Persona

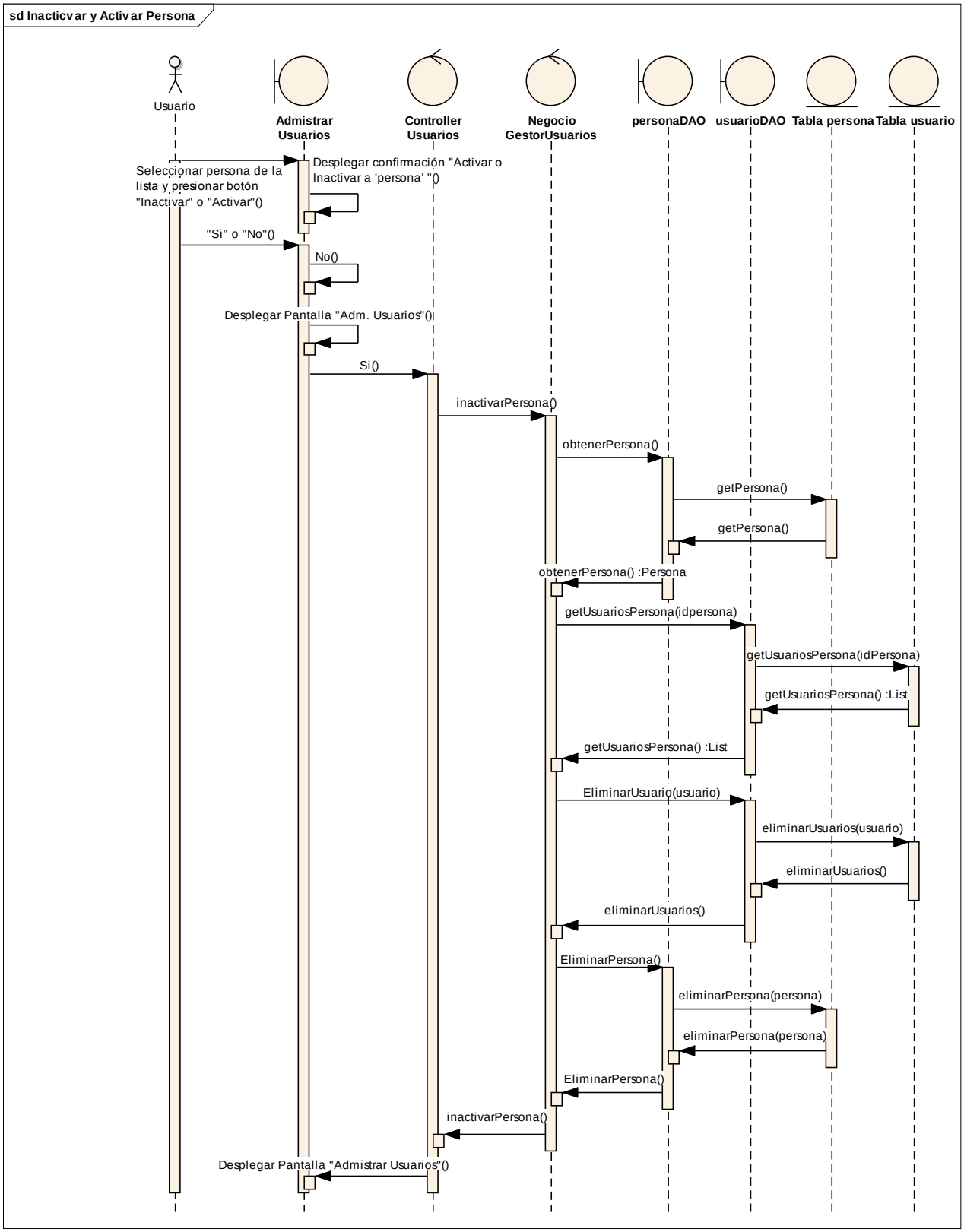


Figura 26. Diagrama de Secuencia Acción (Activar-Inactivar Persona)

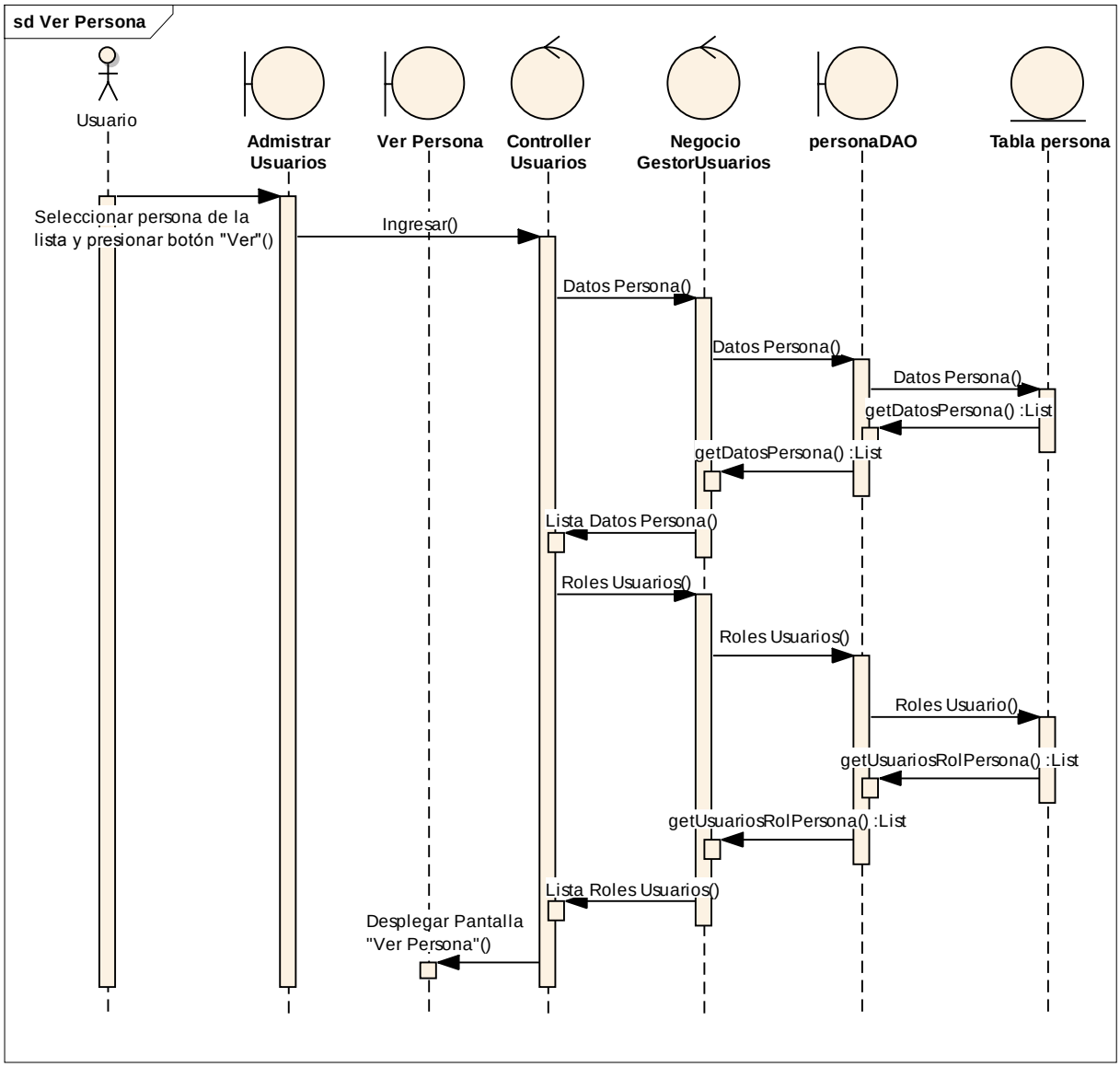


Figura 27. Diagrama de Secuencia Ver Persona

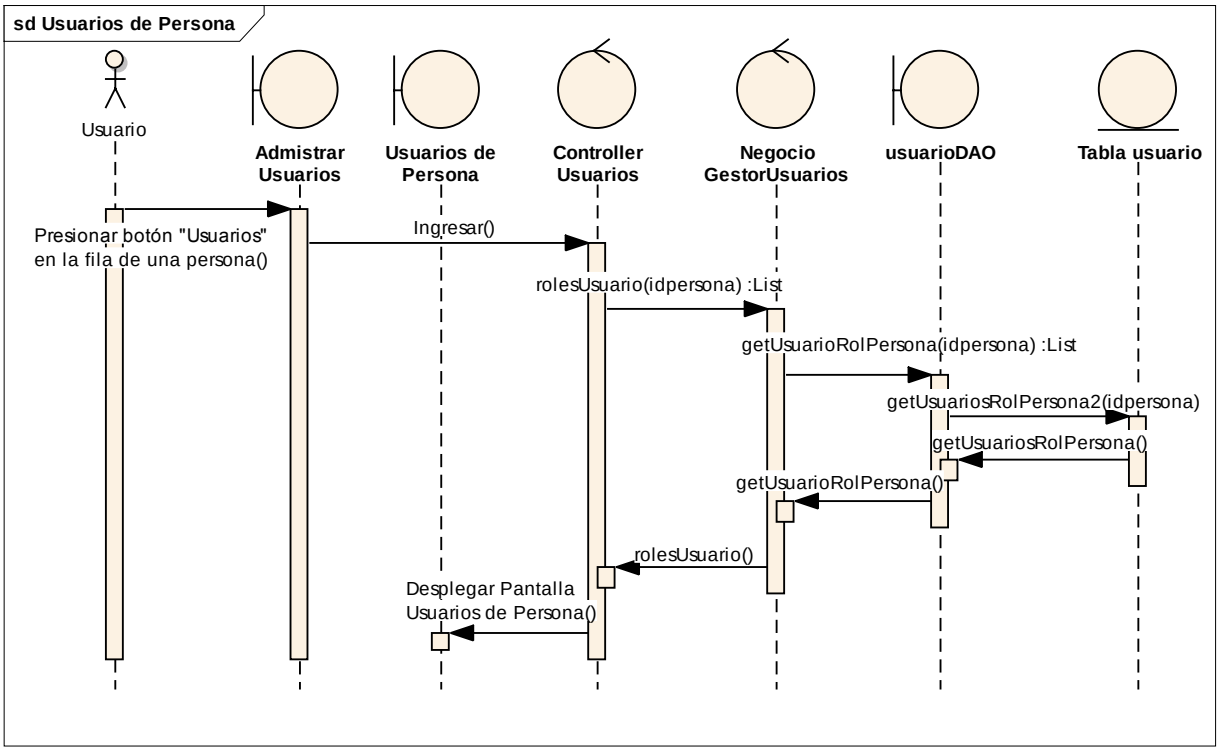


Figura 28. Diagrama de Secuencia Usuarios de Persona

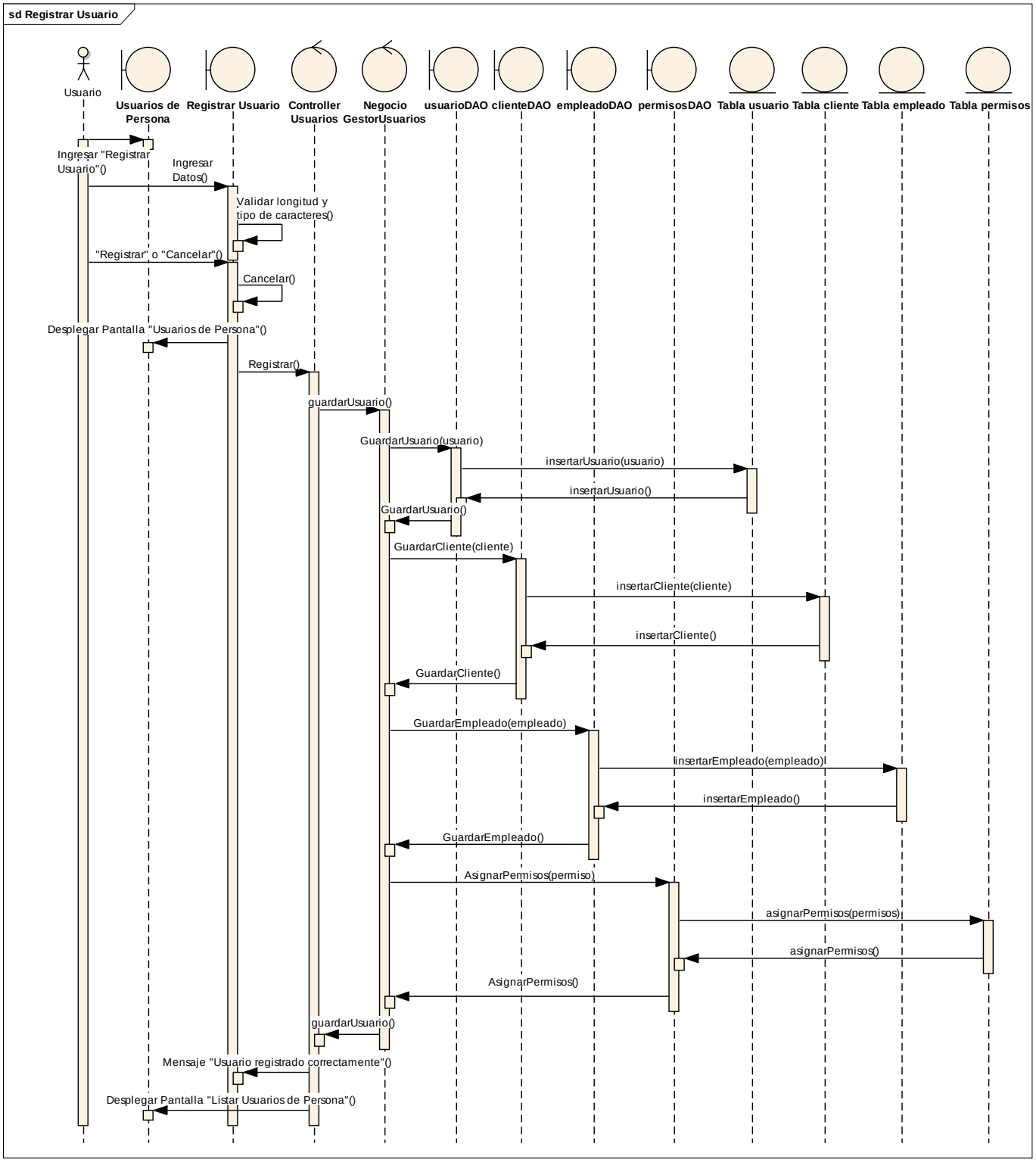


Figura 29. Diagrama de Secuencia Registrar Usuario

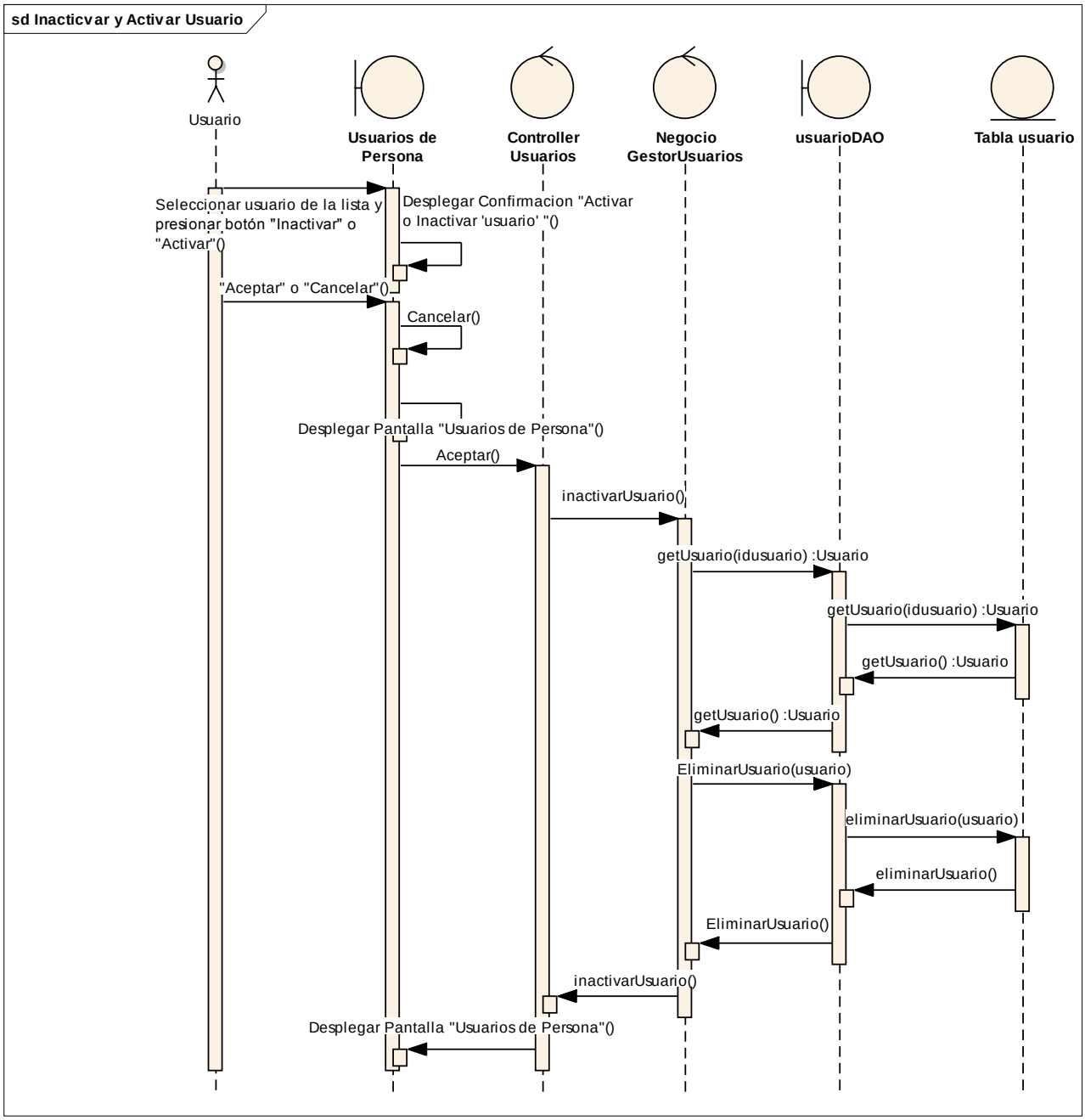


Figura 30. Diagrama de Secuencia Acción (Activar – Inactivar) Usuario

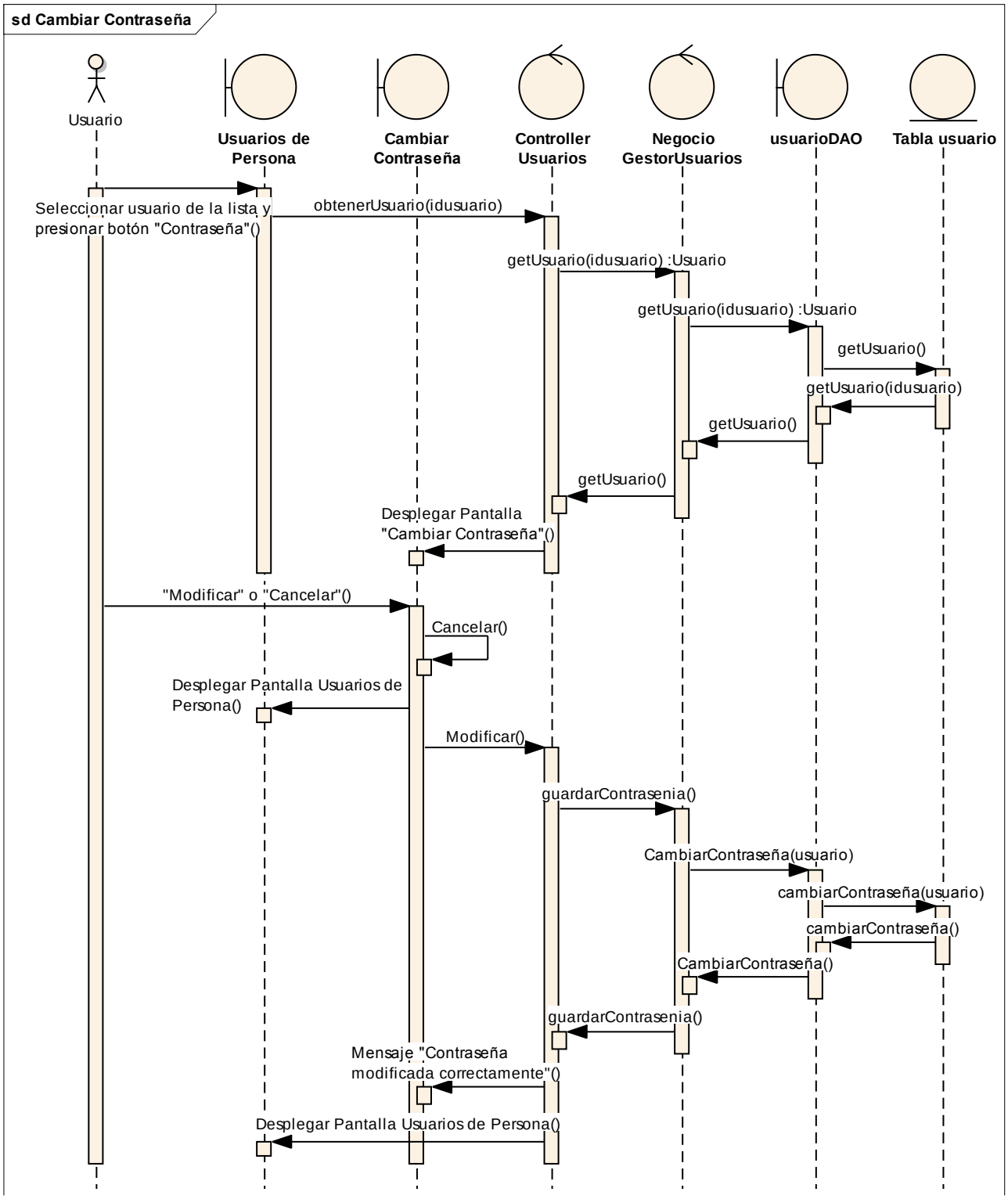


Figura 31. Diagrama de Secuencia Cambiar Contraseña

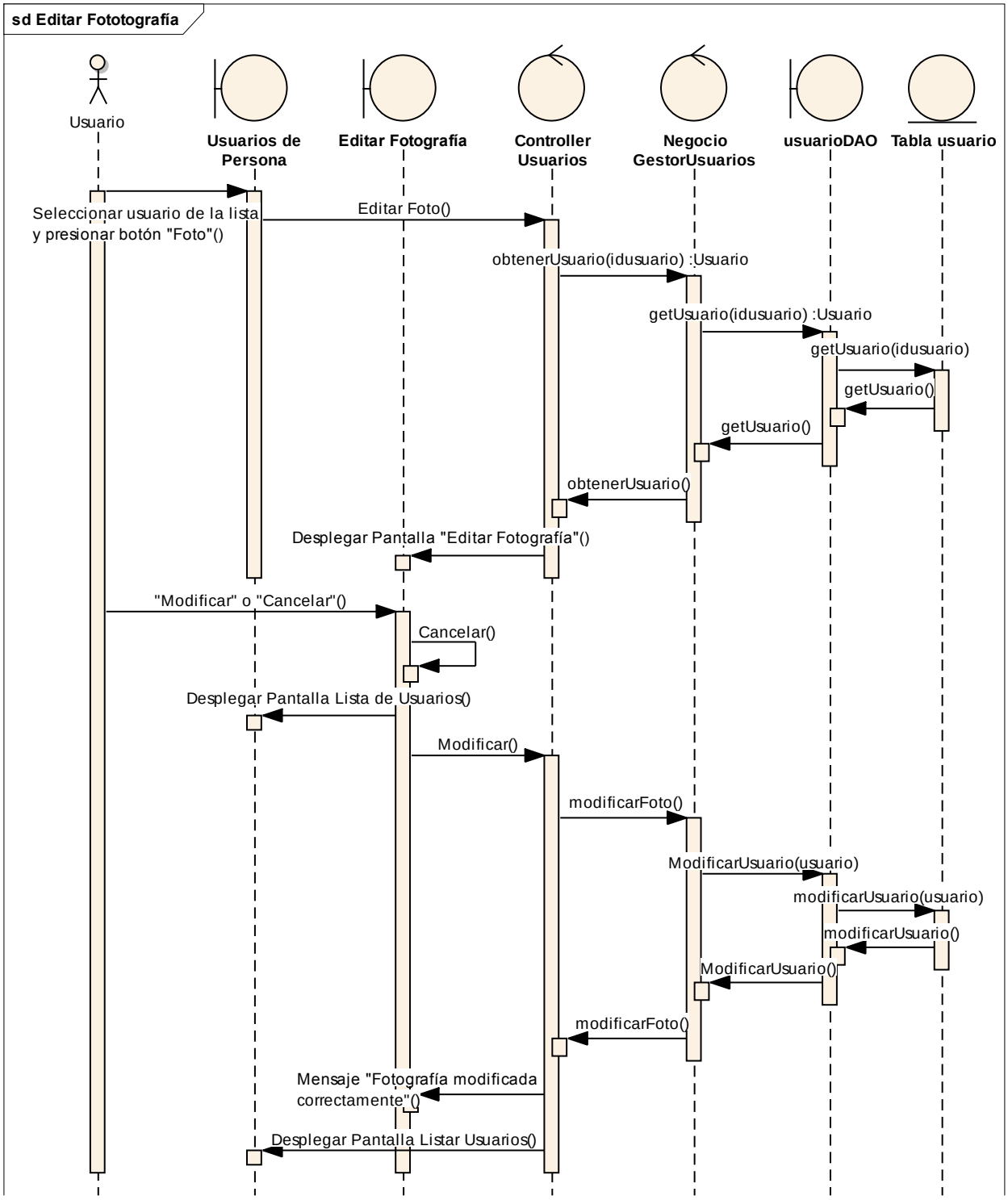


Figura 32. Diagrama de Secuencia Editar Fotografía

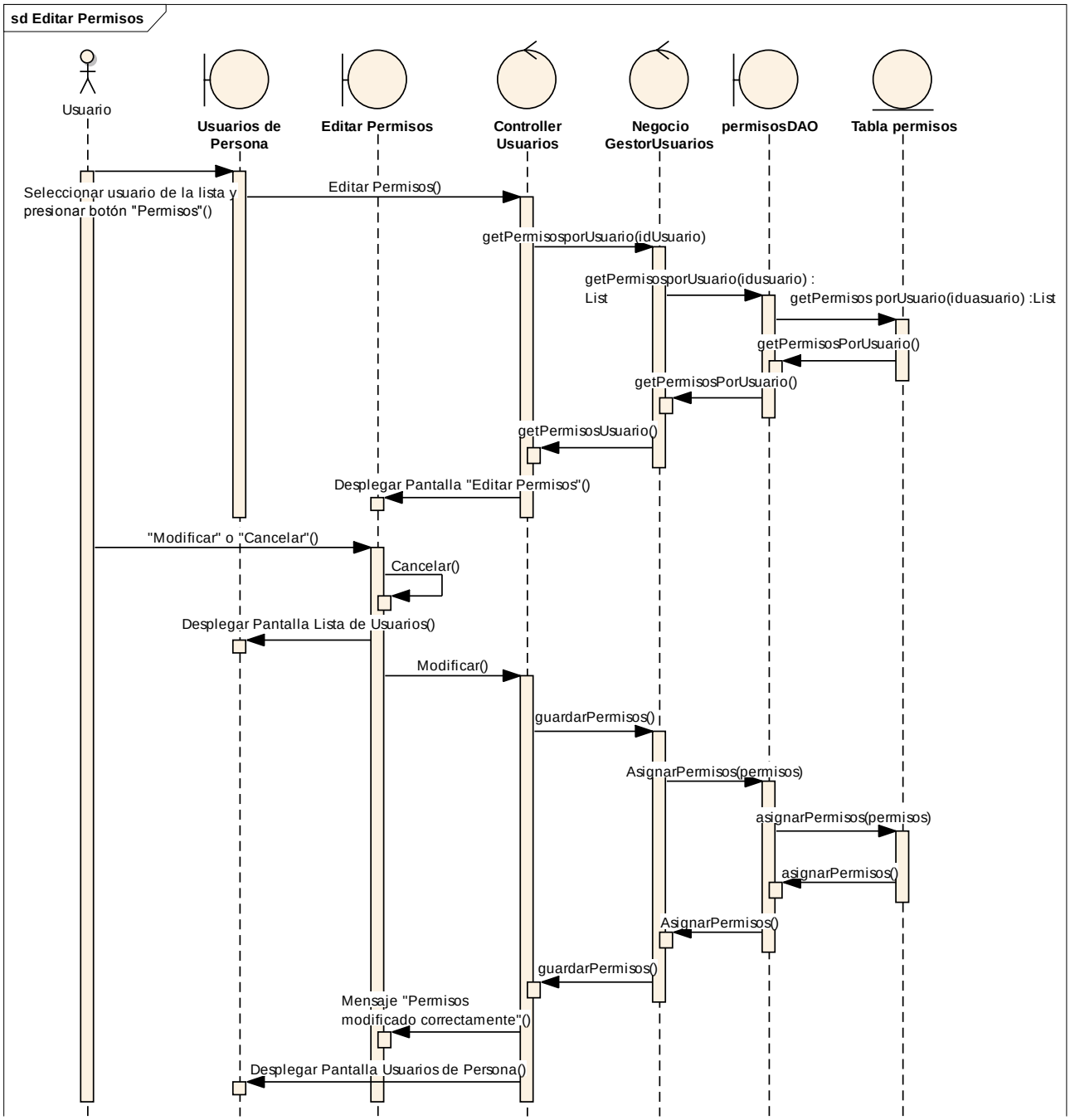


Figura 33. Diagrama de Secuencia Editar Permisos

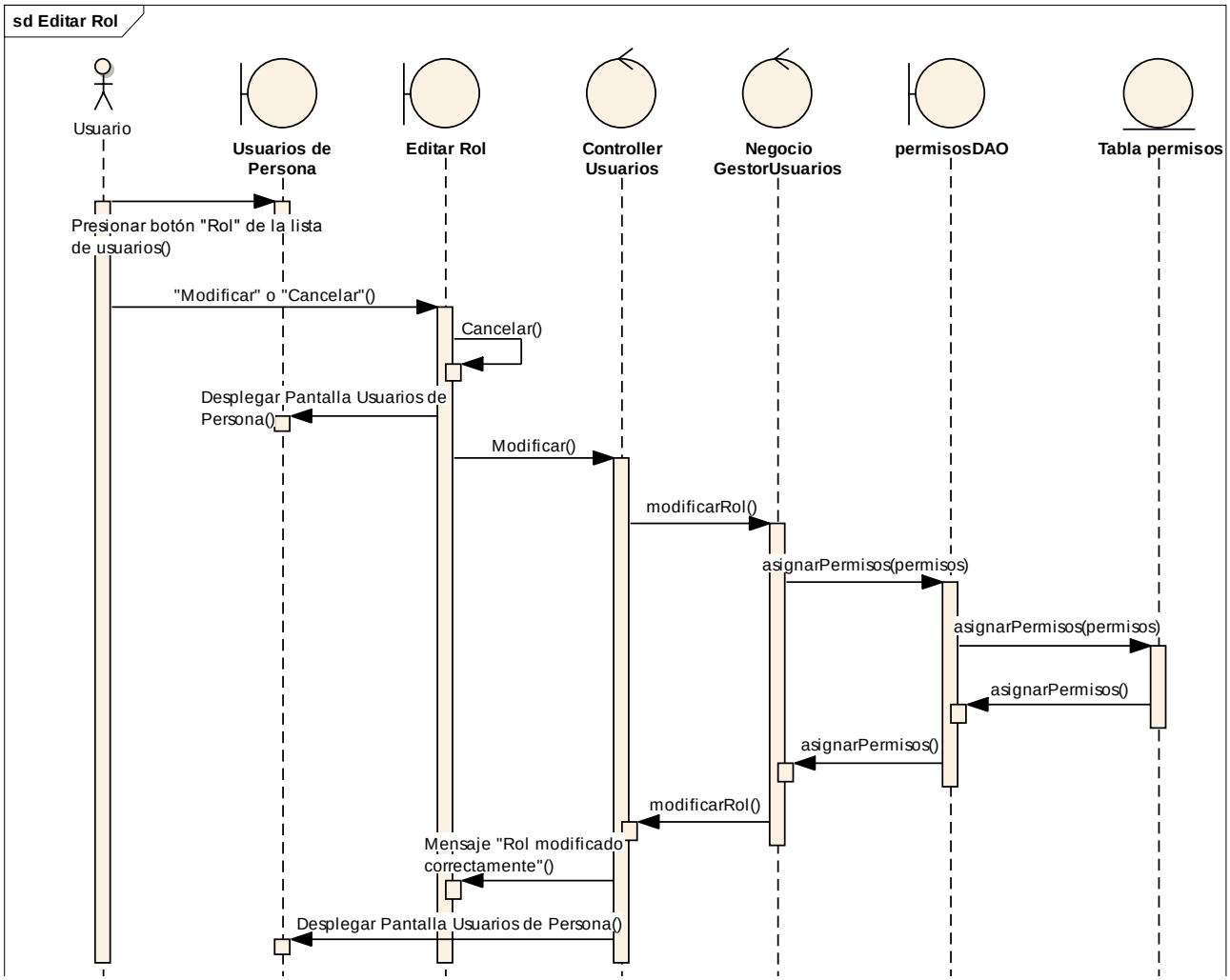


Figura 34. Diagrama de Secuencia Editar Rol

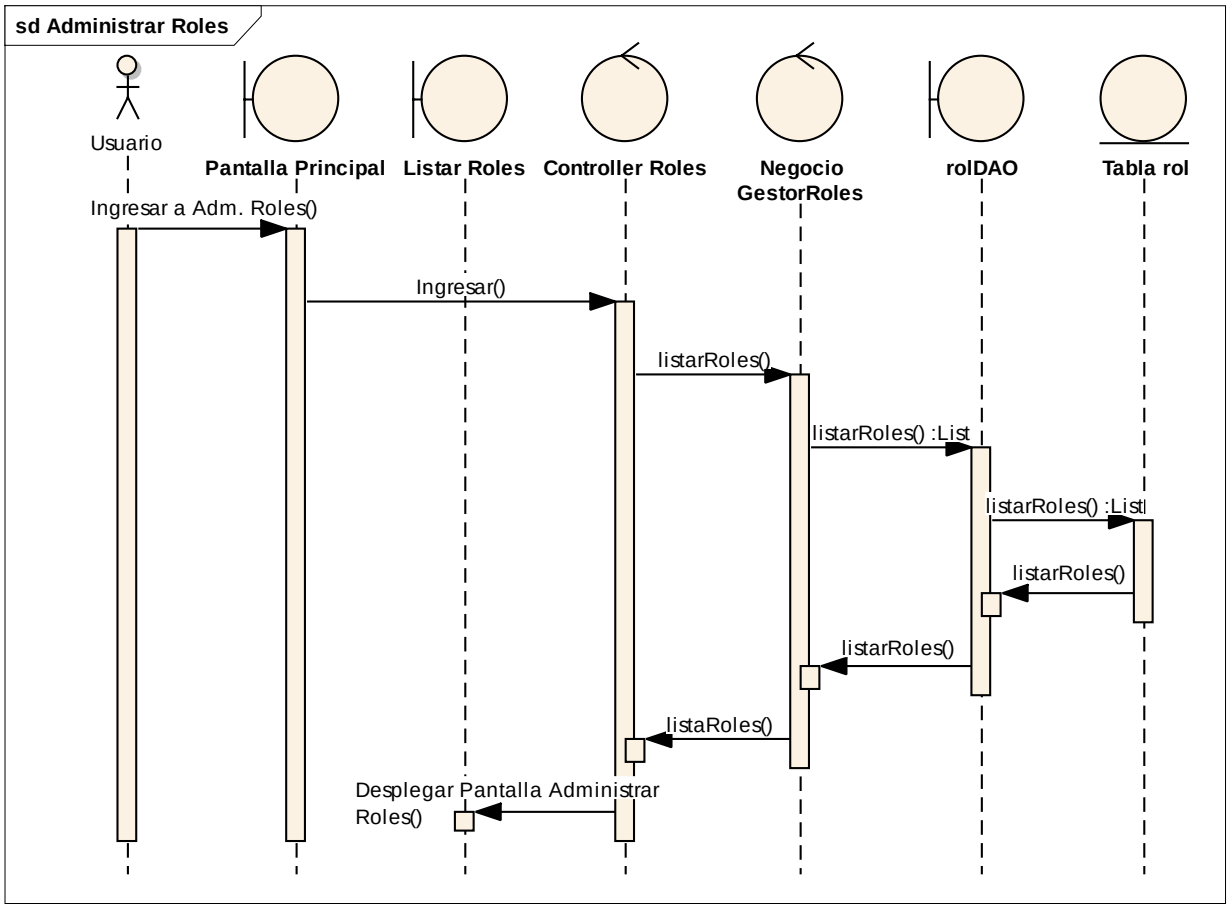


Figura 35. Diagrama de Secuencia Administrar Roles

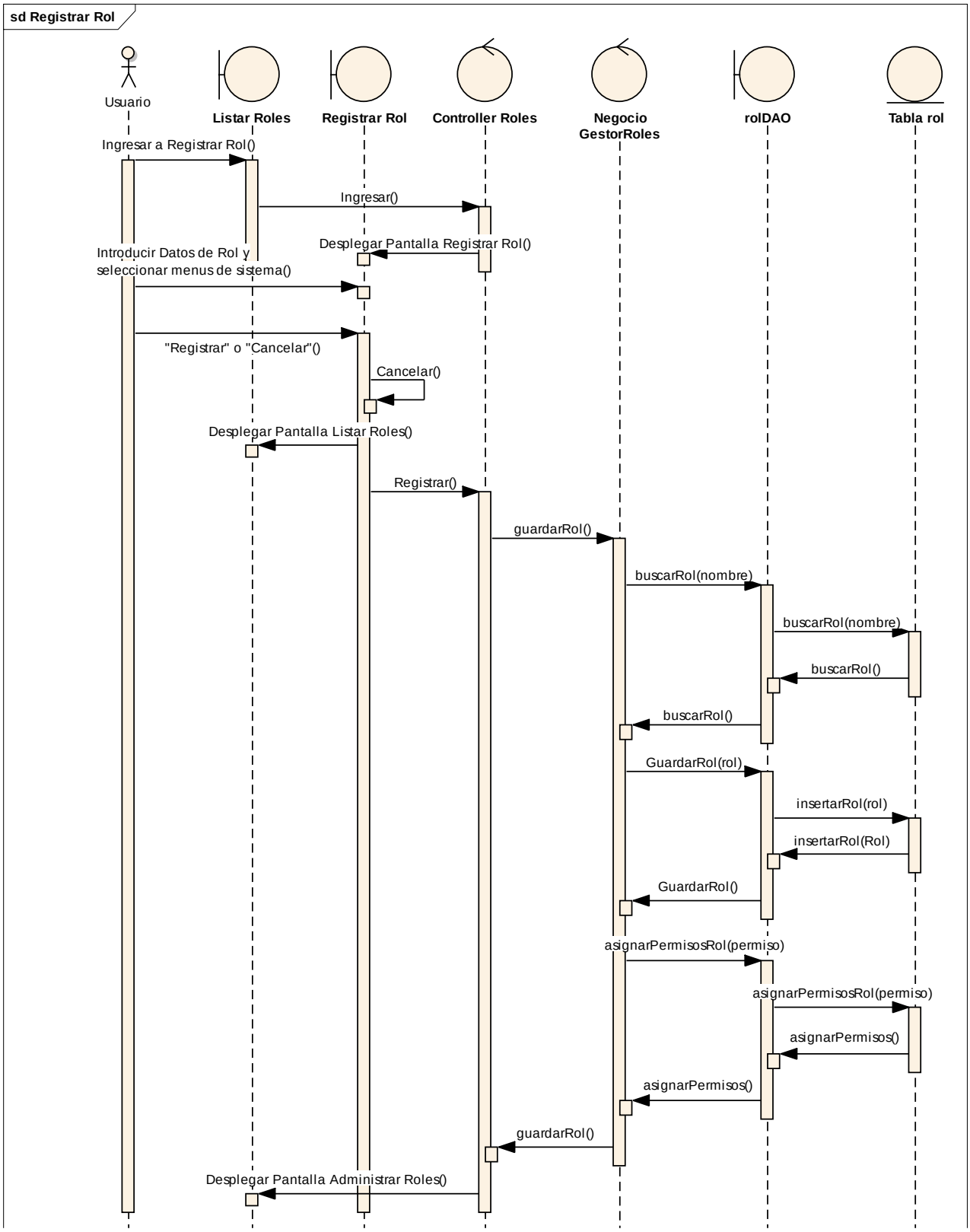


Figura 36. Diagrama de Secuencia Registrar Rol

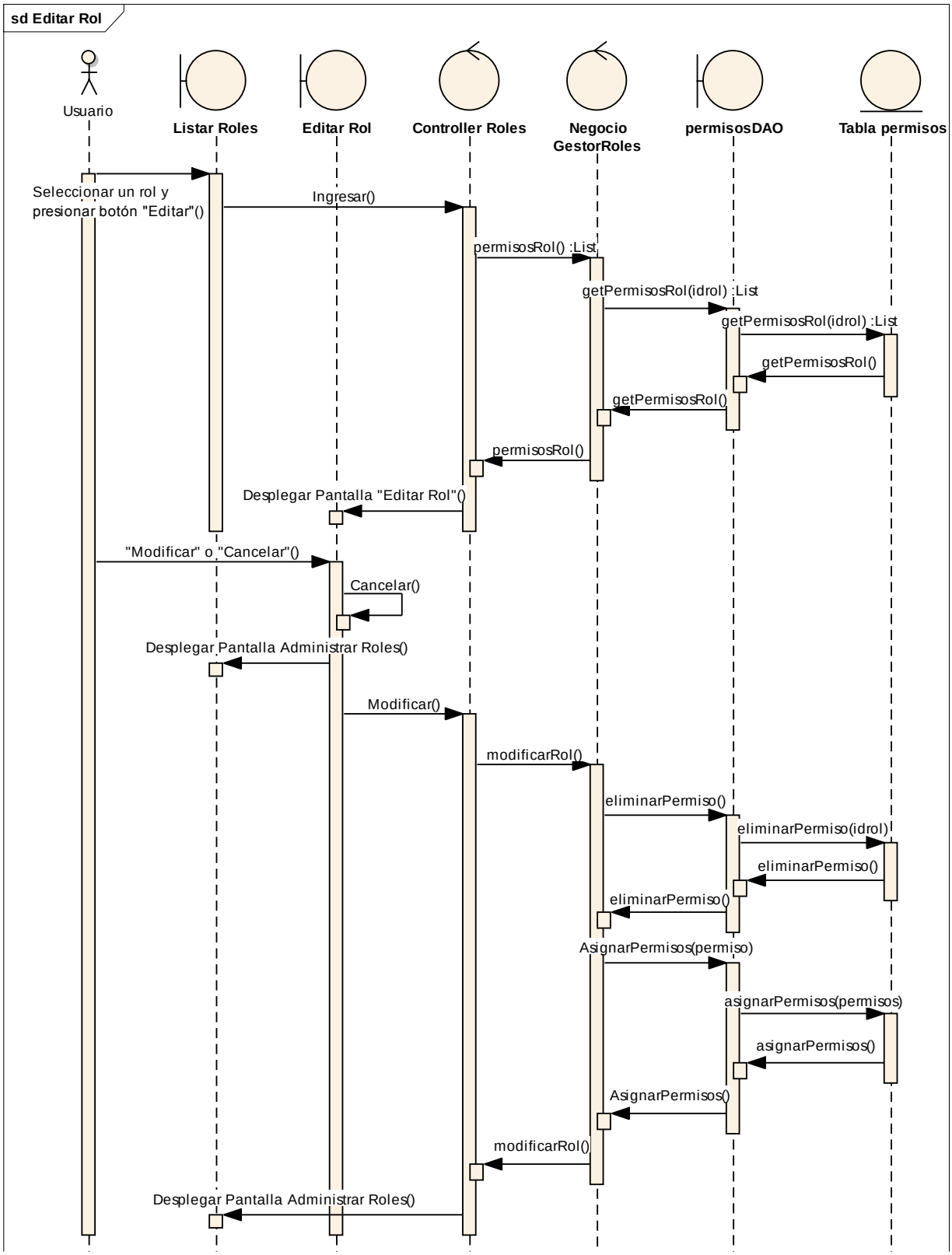


Figura 37. Diagrama de Secuencia Editar Rol

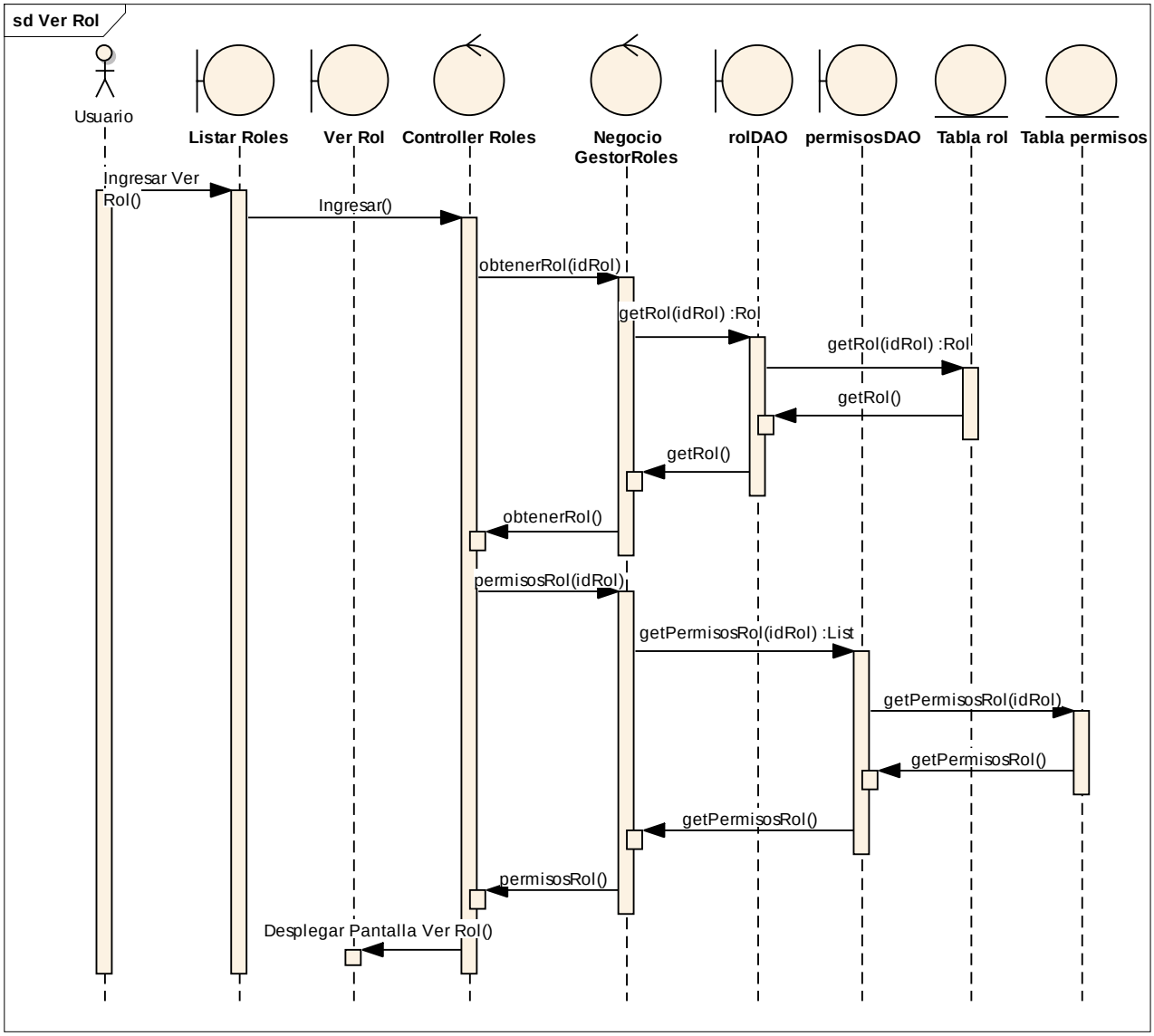


Figura 38. Diagrama de Secuencia Ver Rol

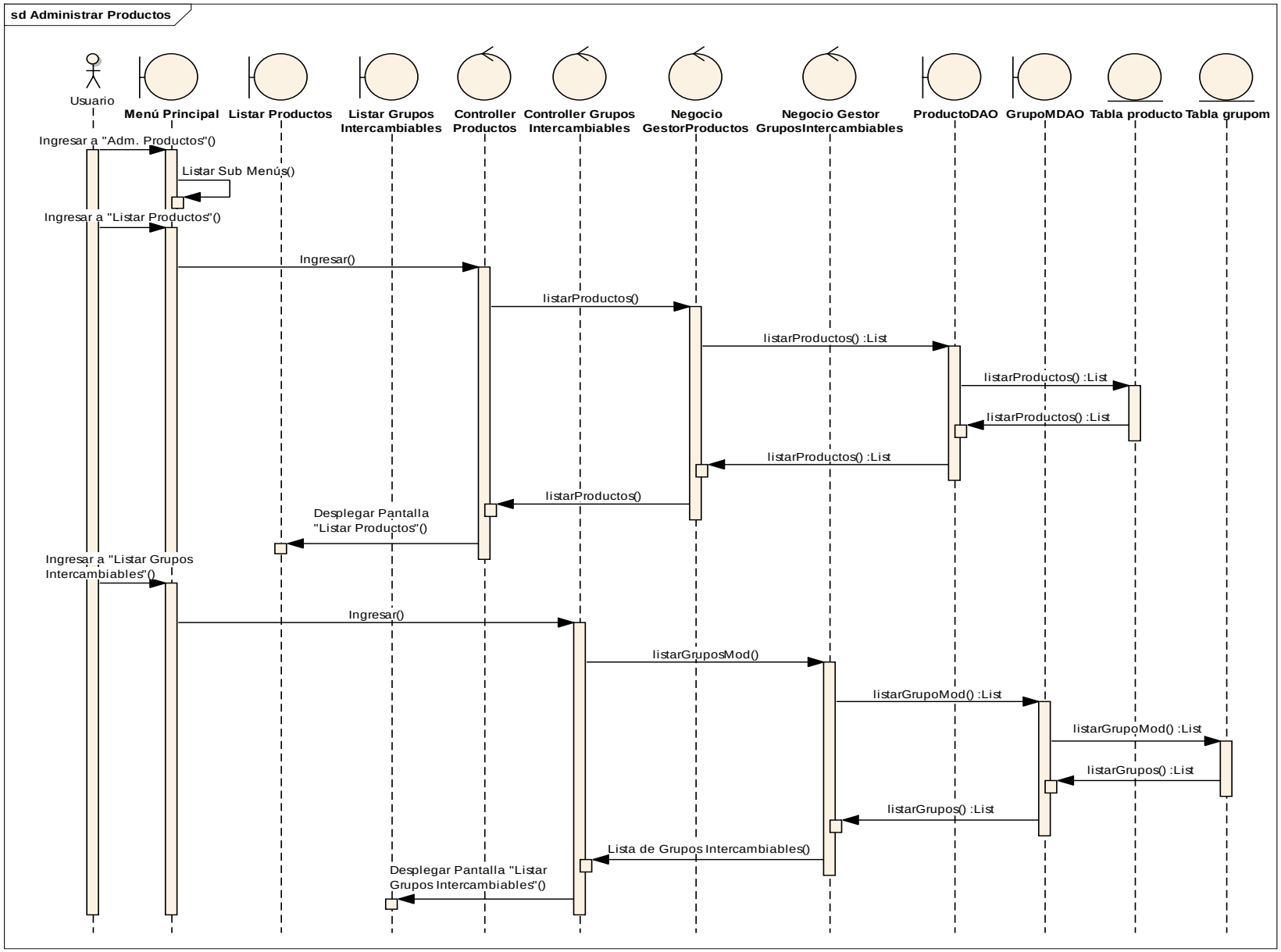


Figura 39. Diagrama de Secuencia Administrar Productos

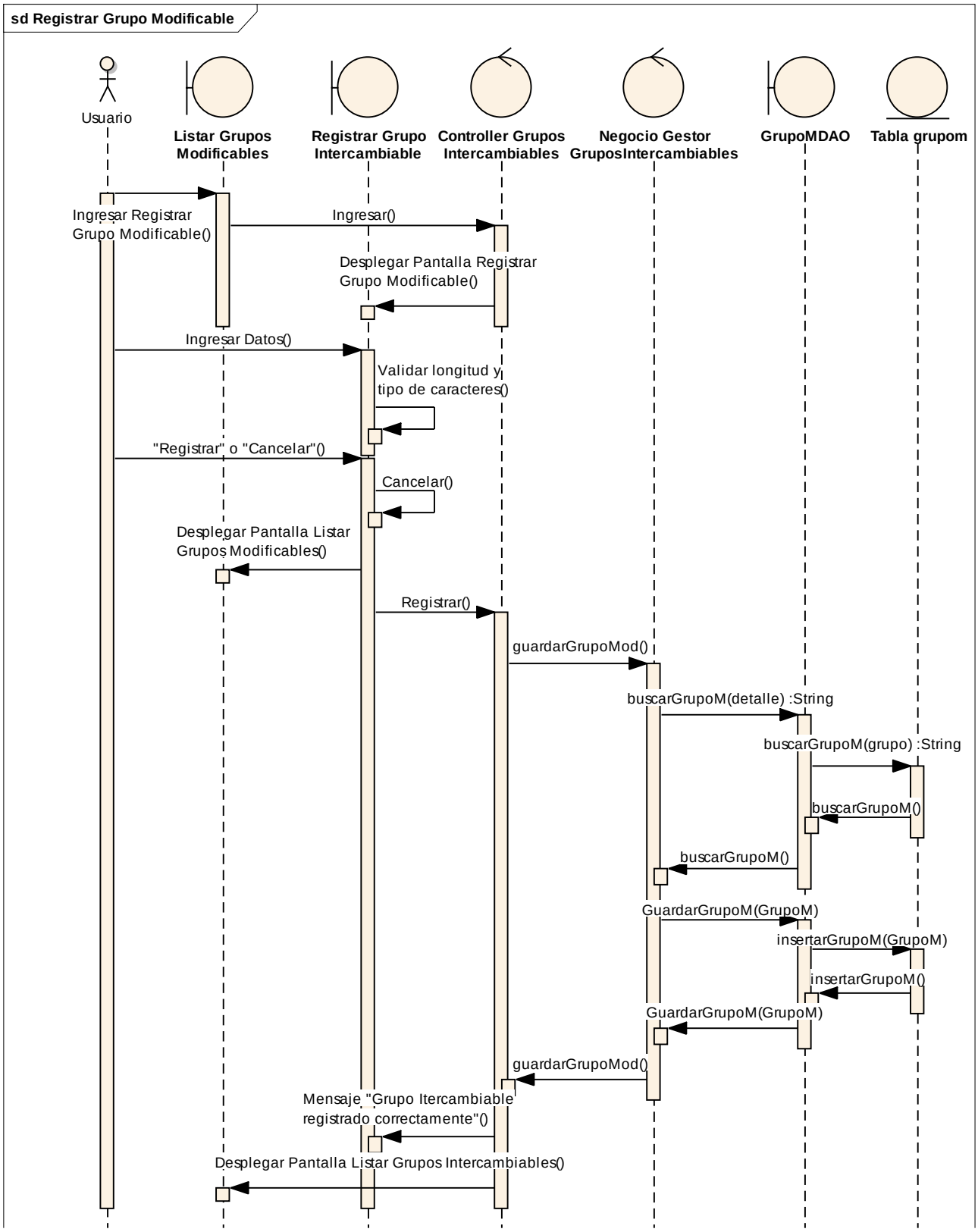


Figura 40. Diagrama de Secuencia Registrar Grupo Modificable

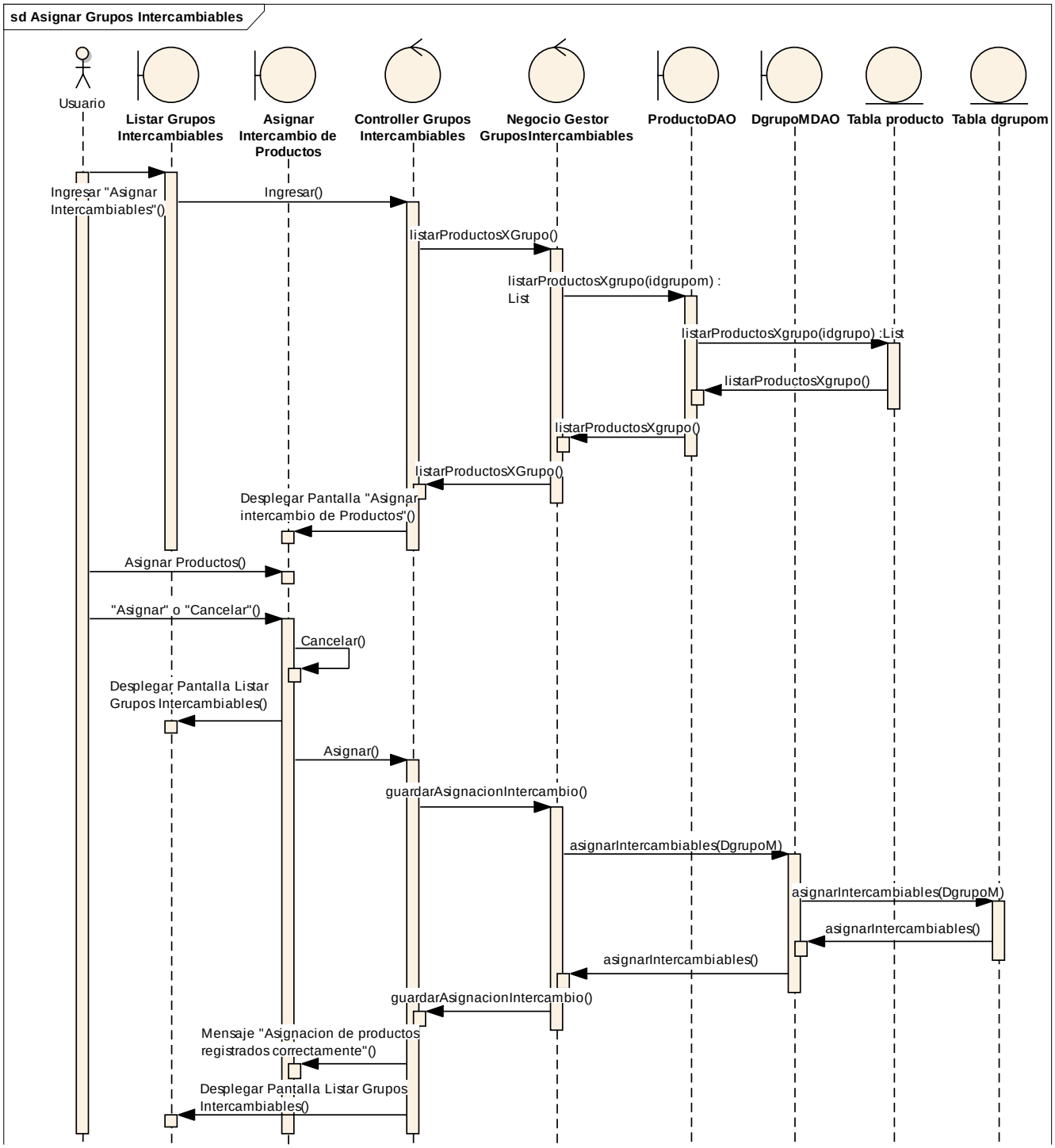


Figura 41. Diagrama de Secuencia Asignar Grupos Intercambiables

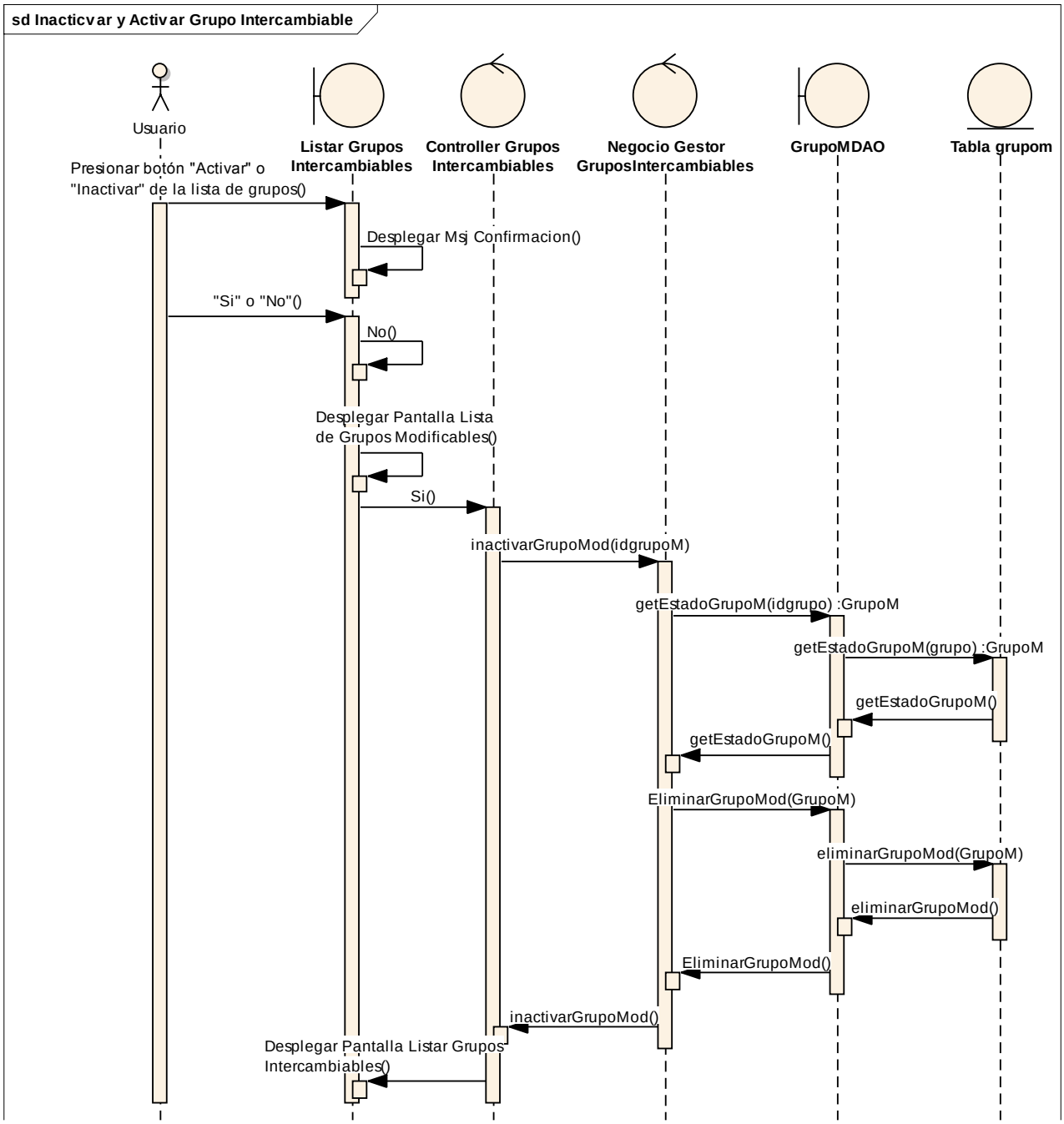


Figura 42. Diagrama de Secuencia Acción (Activar-Inactivar Grupo Intercambiable)

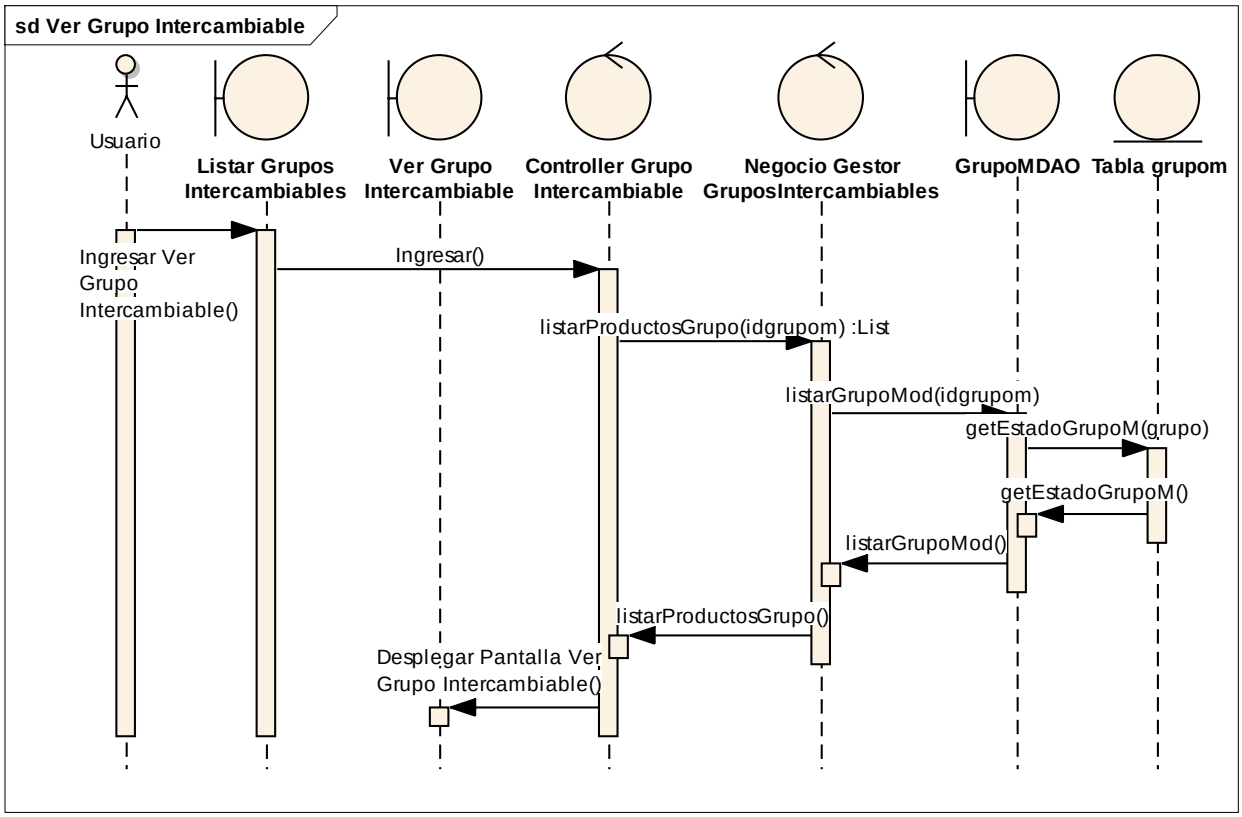


Figura 43. Diagrama de Secuencia Ver Grupo Intercambiable

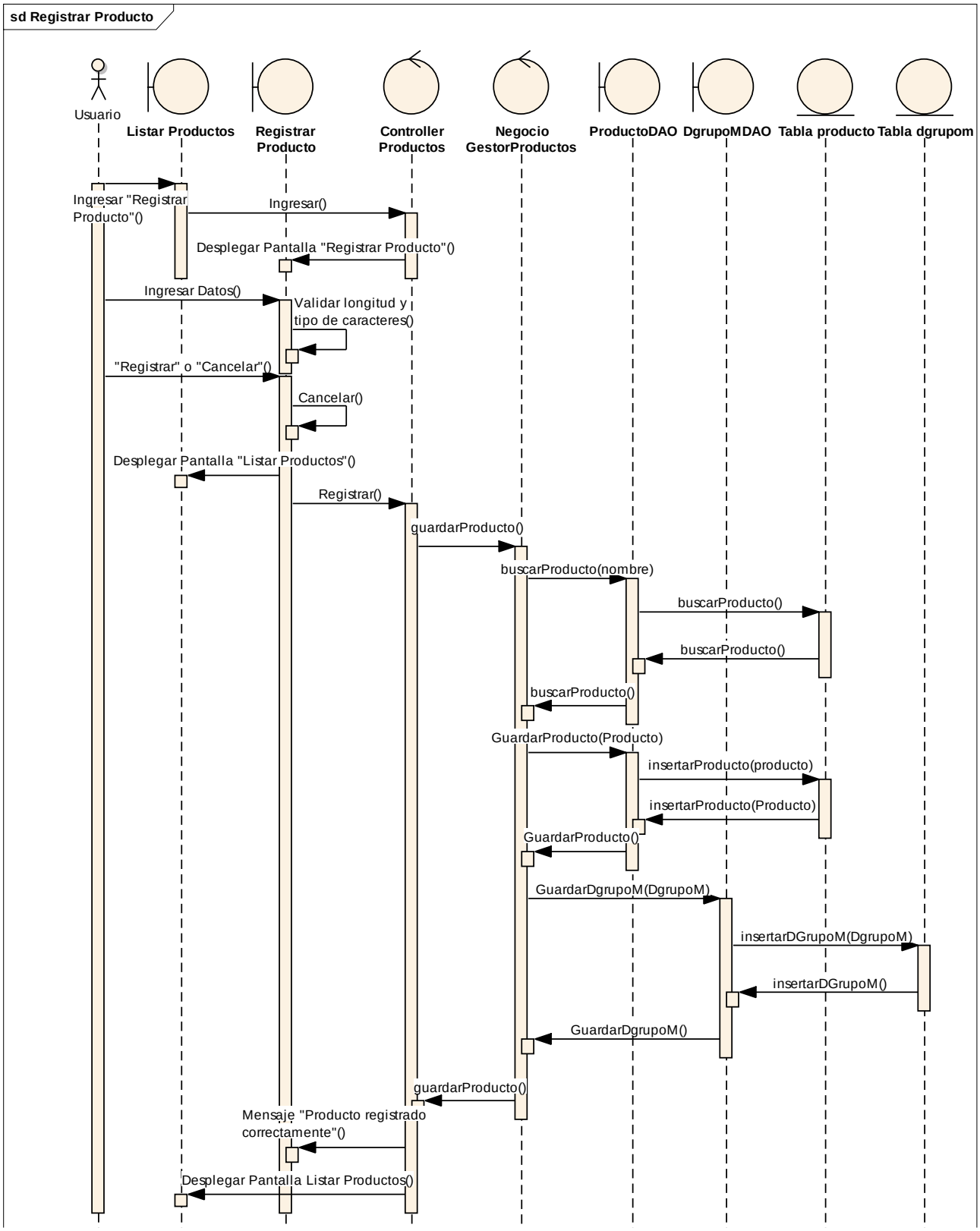


Figura 44. Diagrama de Secuencia Registrar Producto

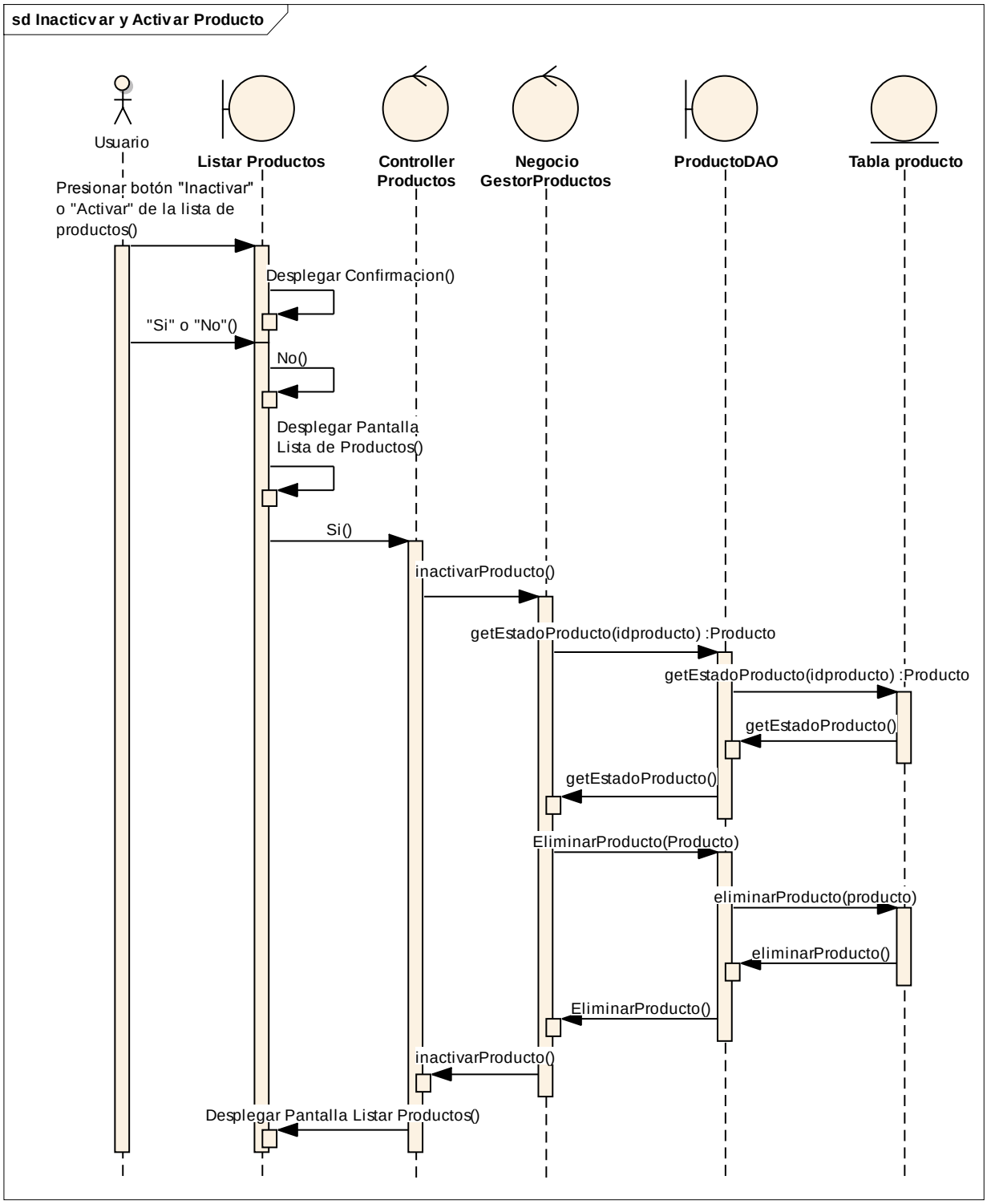


Figura 45. Diagrama de Secuencia Acción (Activar-Inactivar Producto)

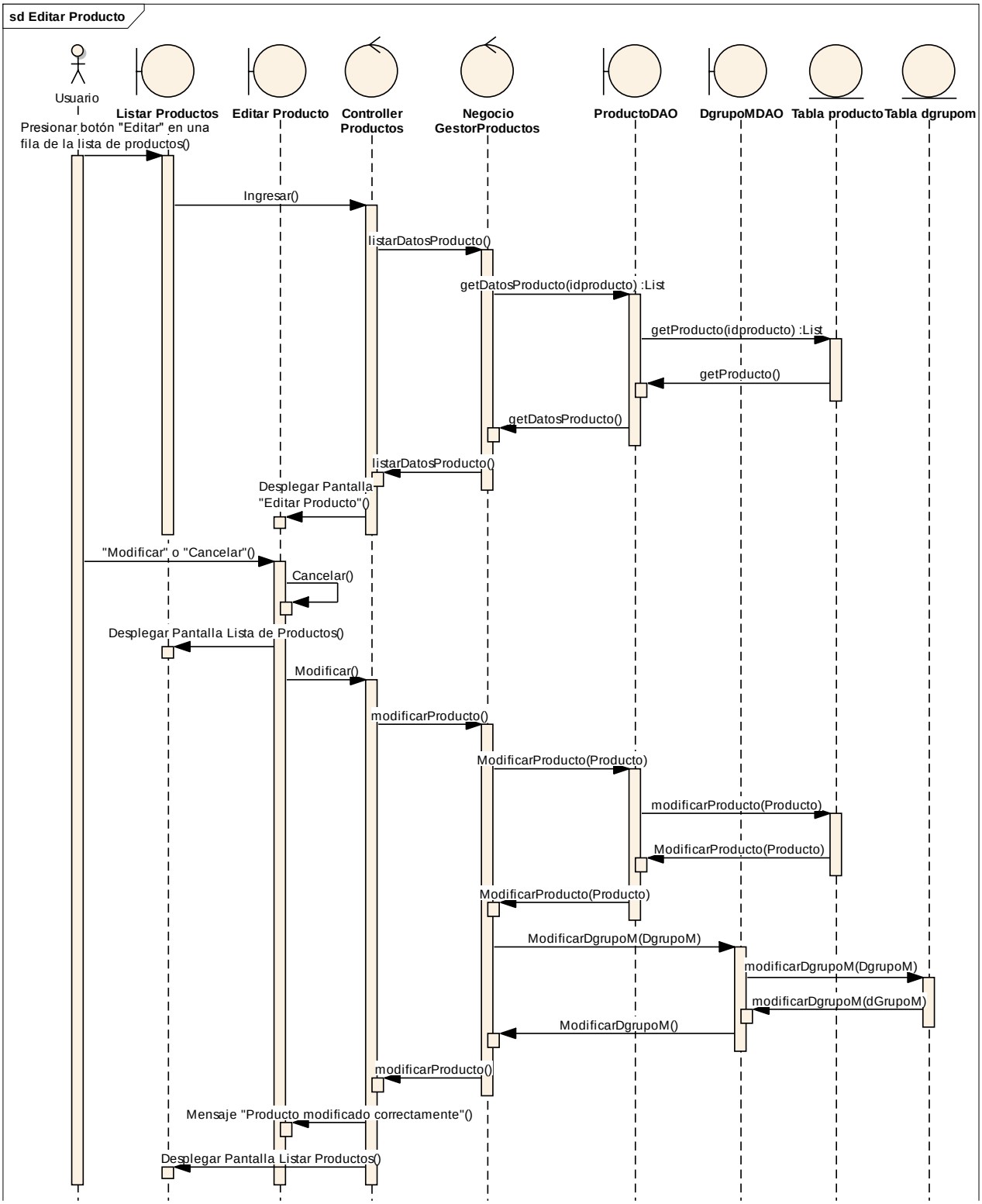


Figura 46. Diagrama de Secuencia Editar Producto

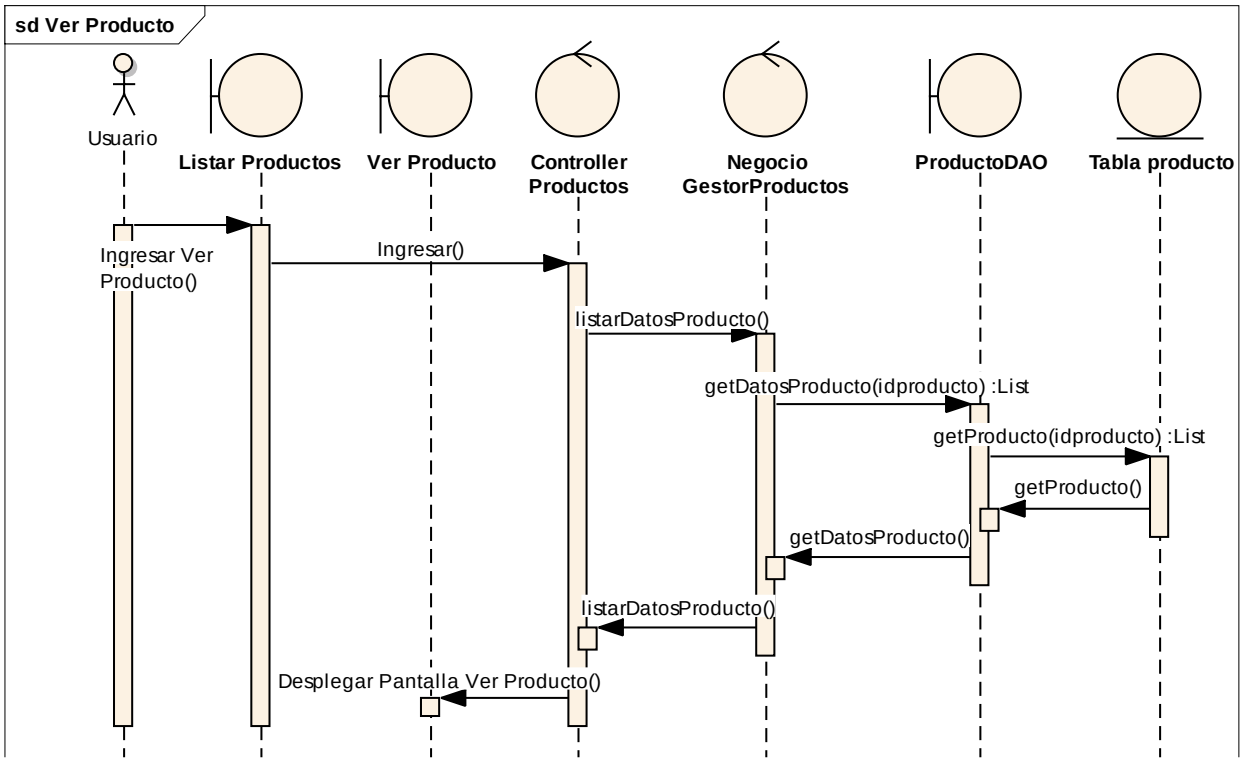


Figura 47. Diagrama de Secuencia Ver Producto

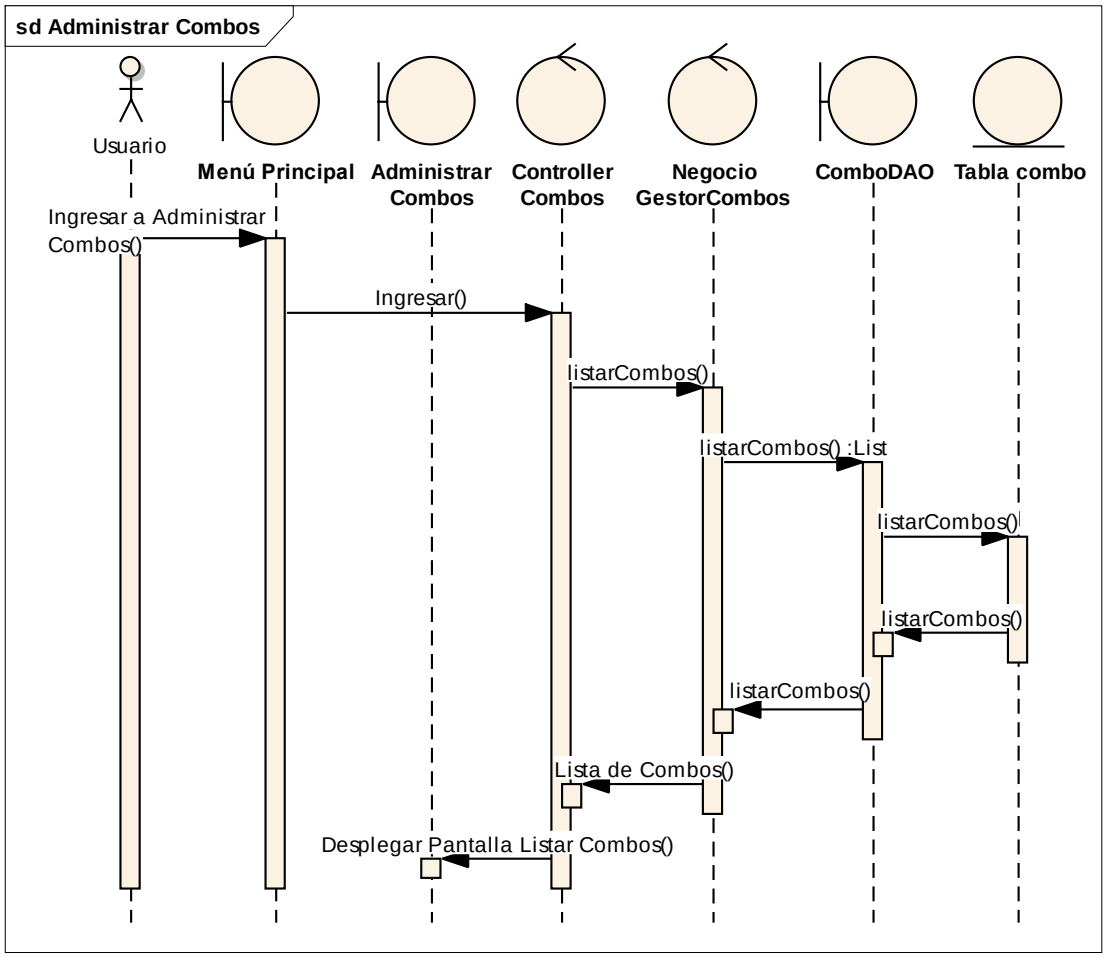


Figura 48. Diagrama de Secuencia Administrar Combos

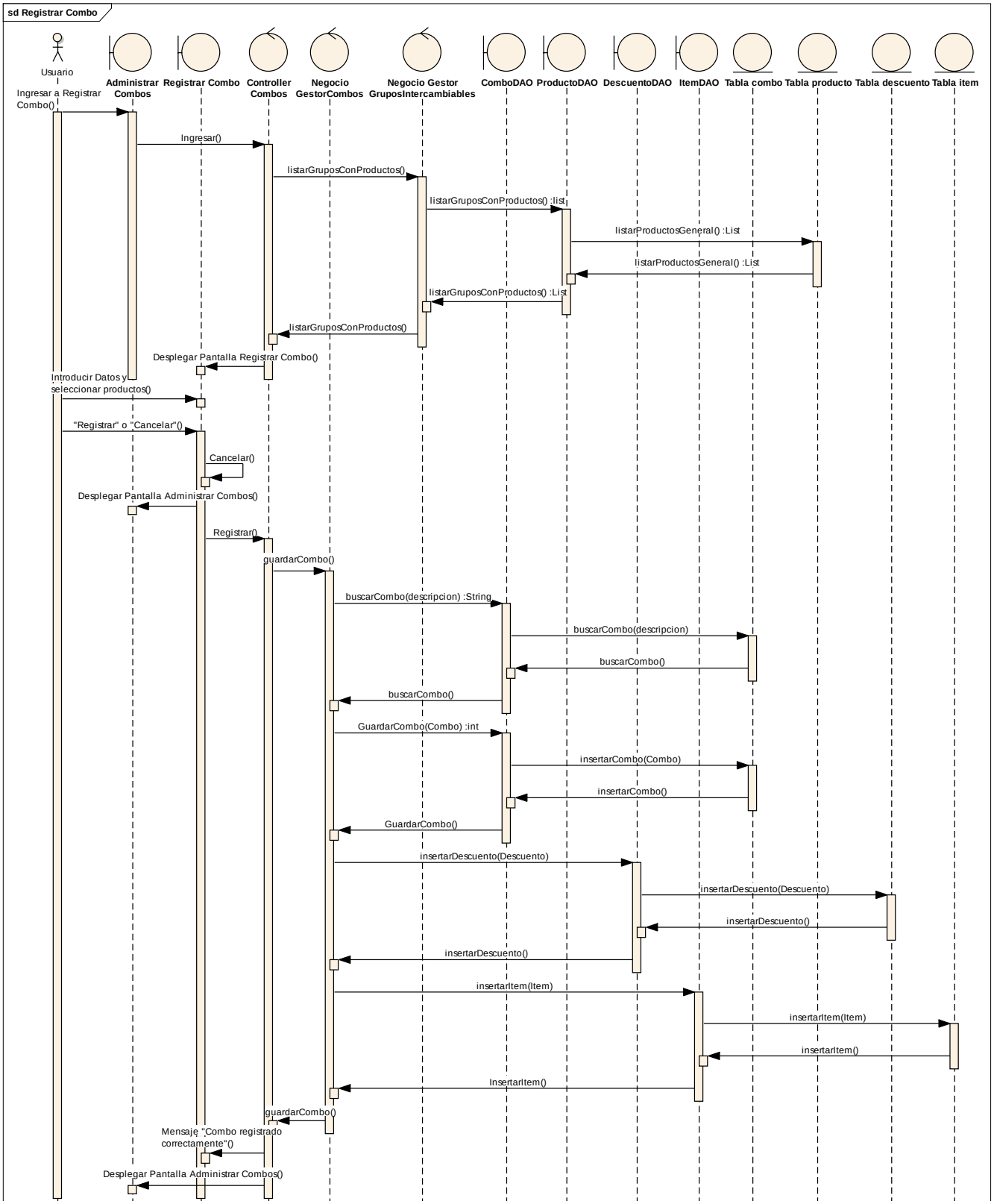


Figura 49. Diagrama de Secuencia Registrar Combo

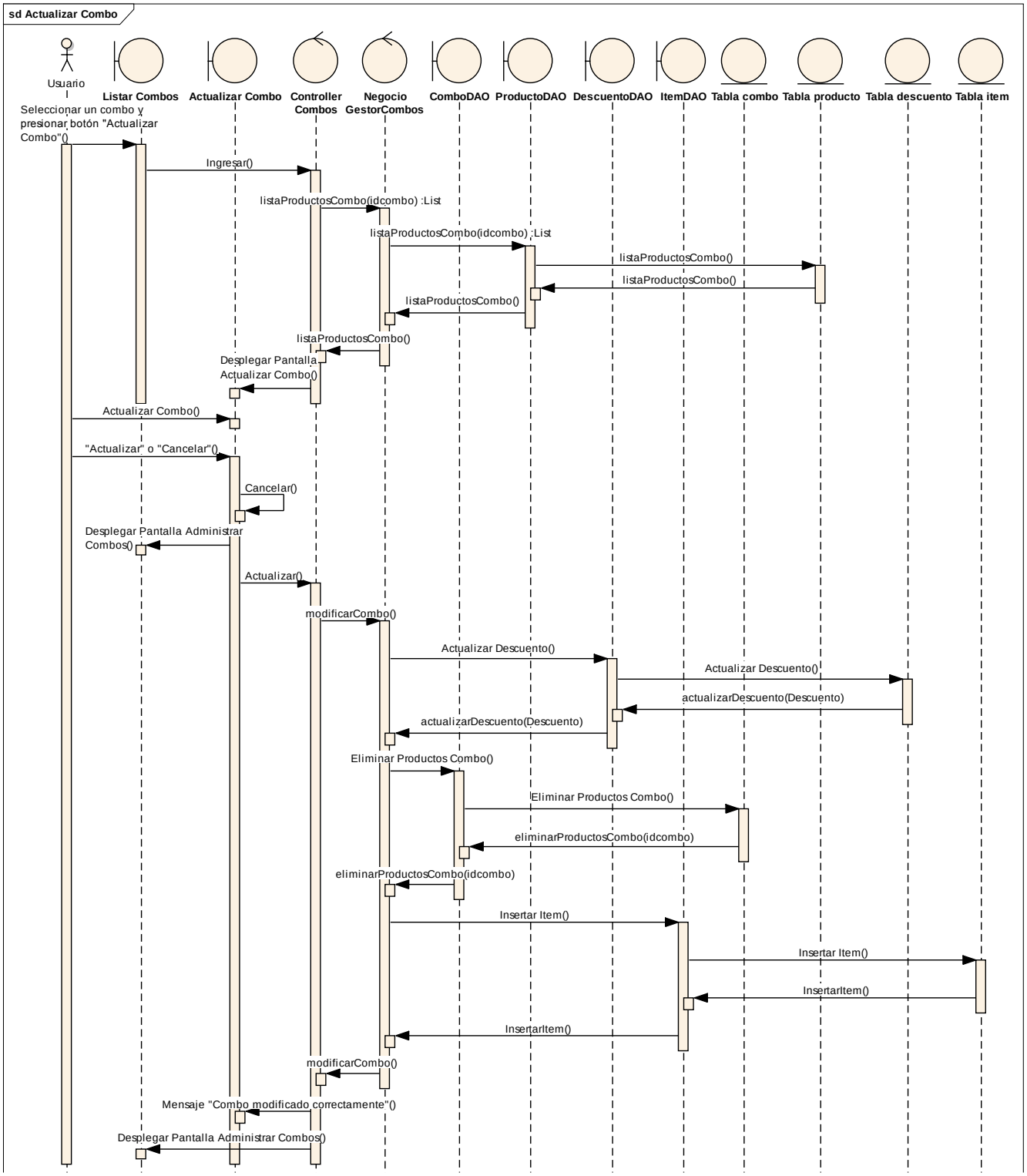


Figura 50. Diagrama de Secuencia Actualizar Combo

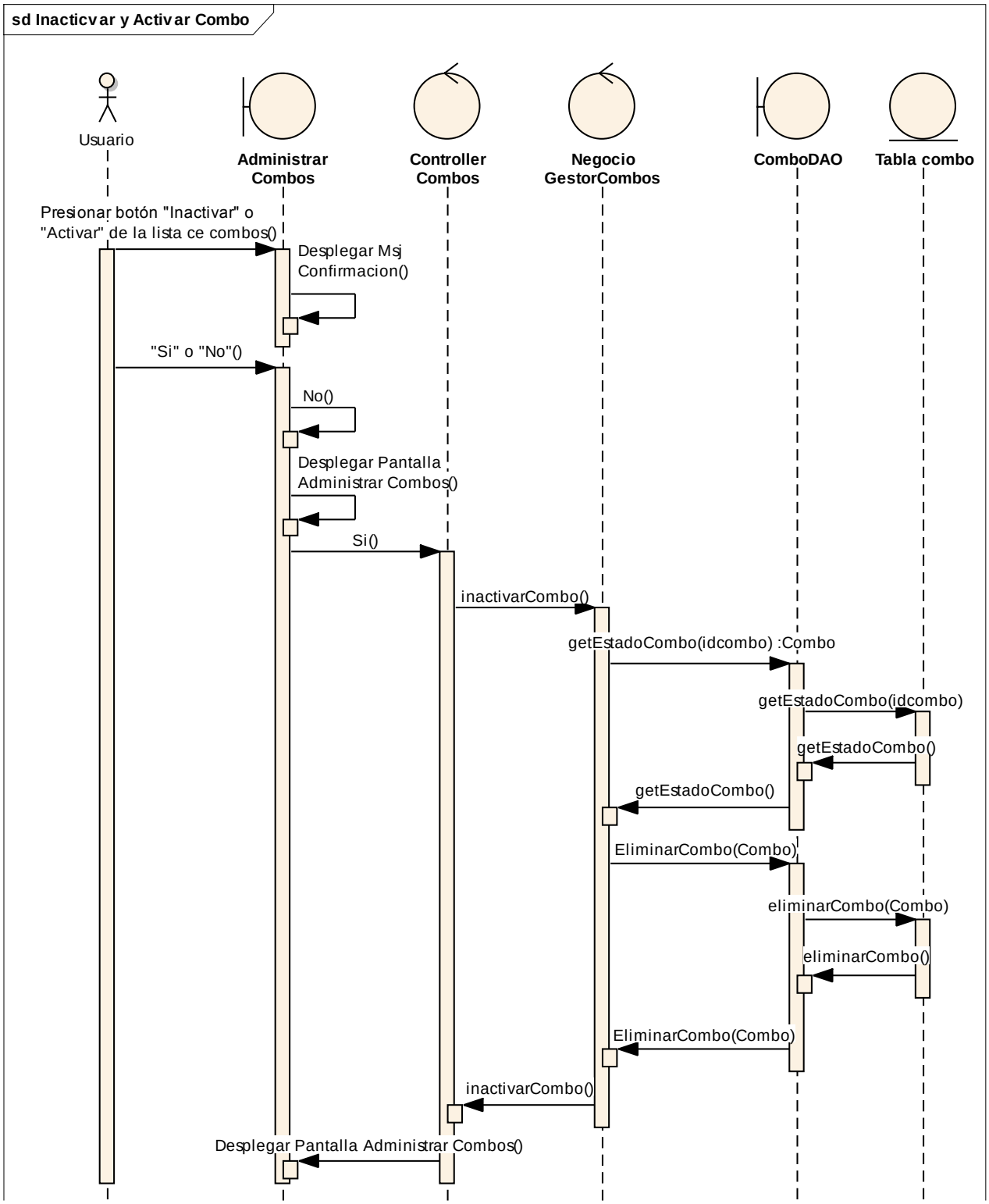


Figura 51. Diagrama de Secuencia Acción (Activar-Inactivar Combo)

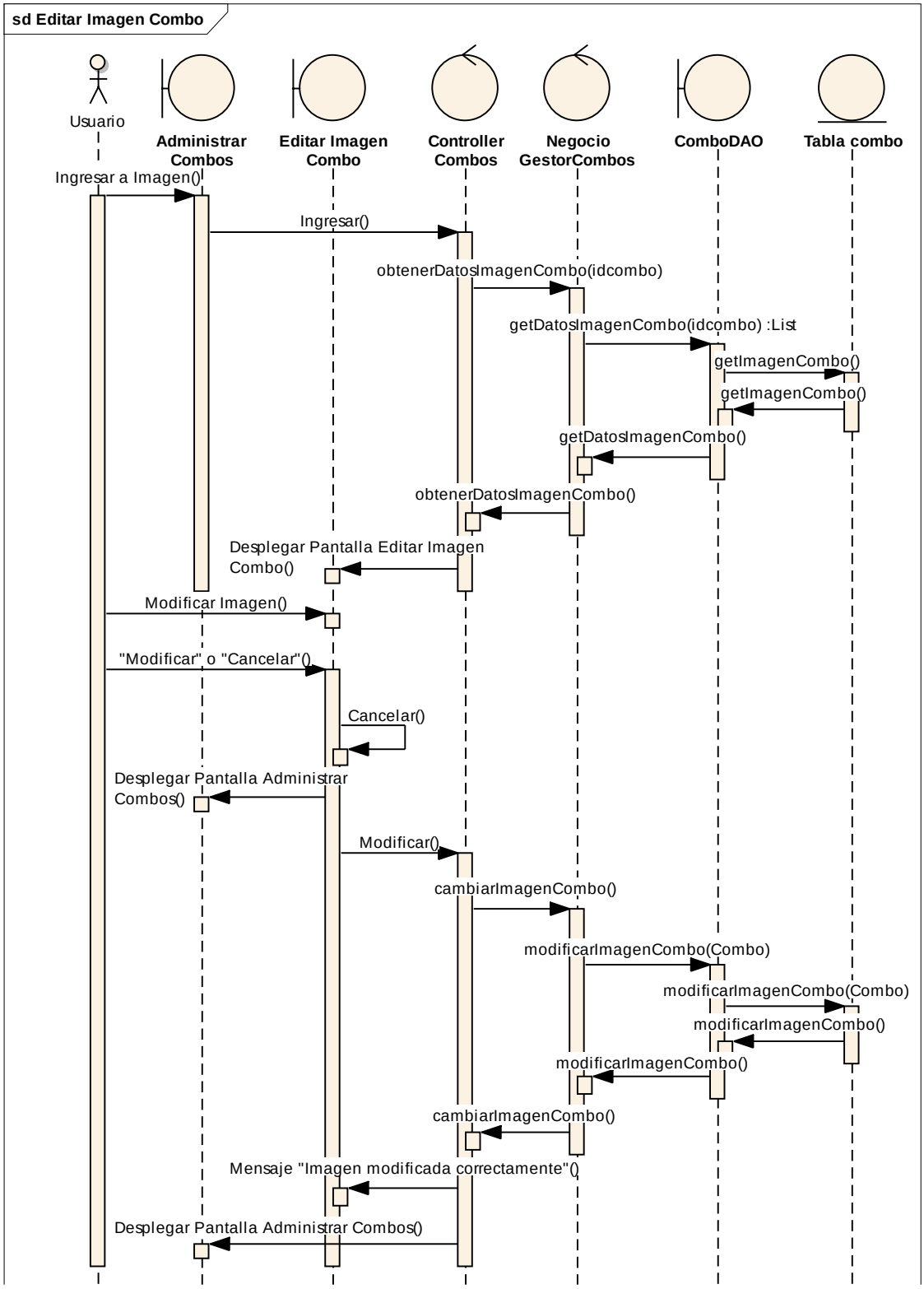


Figura 52. Diagrama de Secuencia Editar Imagen Combo

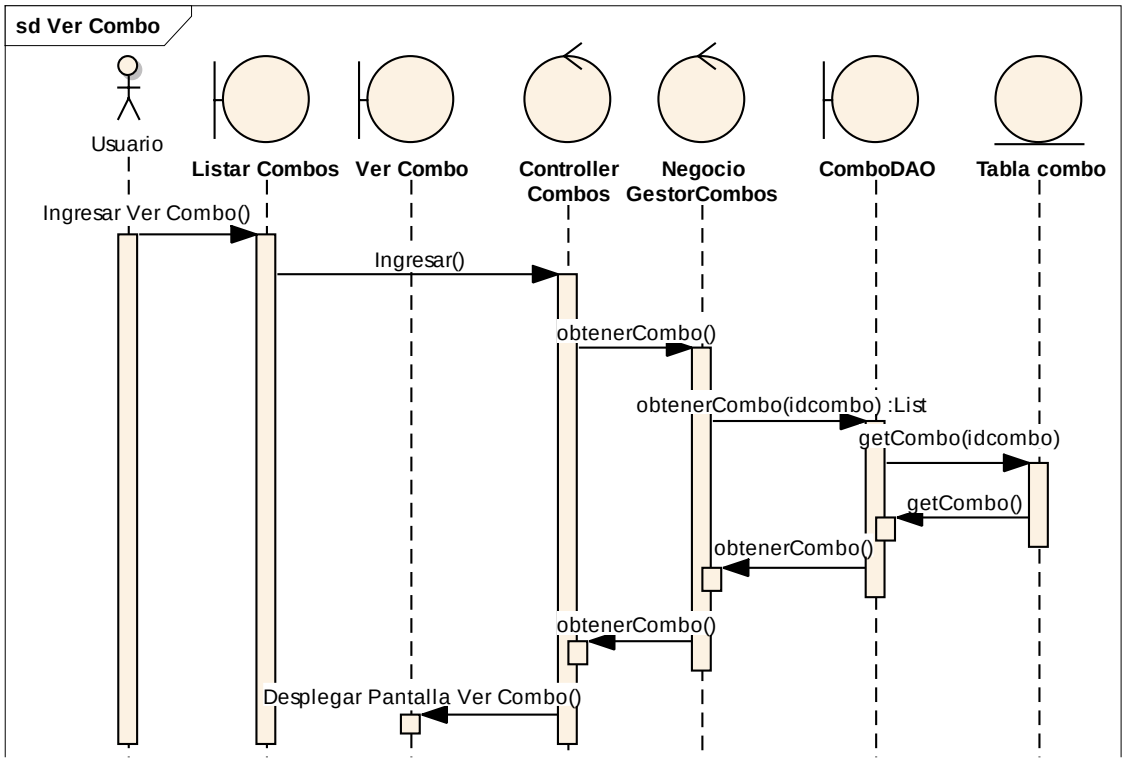


Figura 53. Diagrama de Secuencia Ver Combo

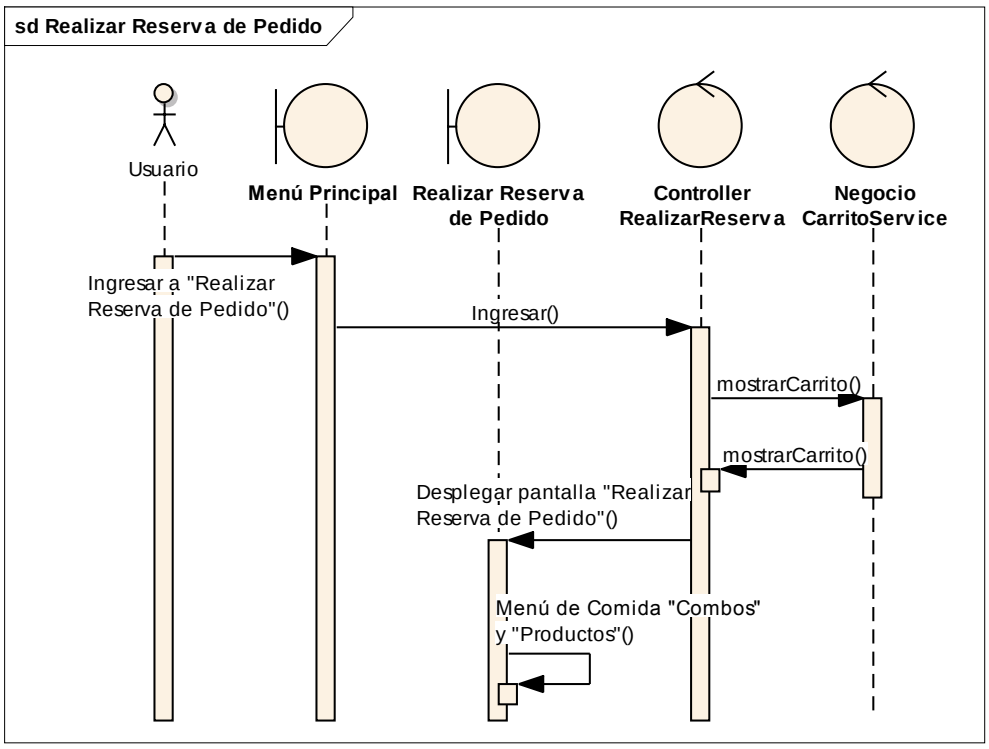


Figura 54. Diagrama de Secuencia Realizar Reserva de Pedido

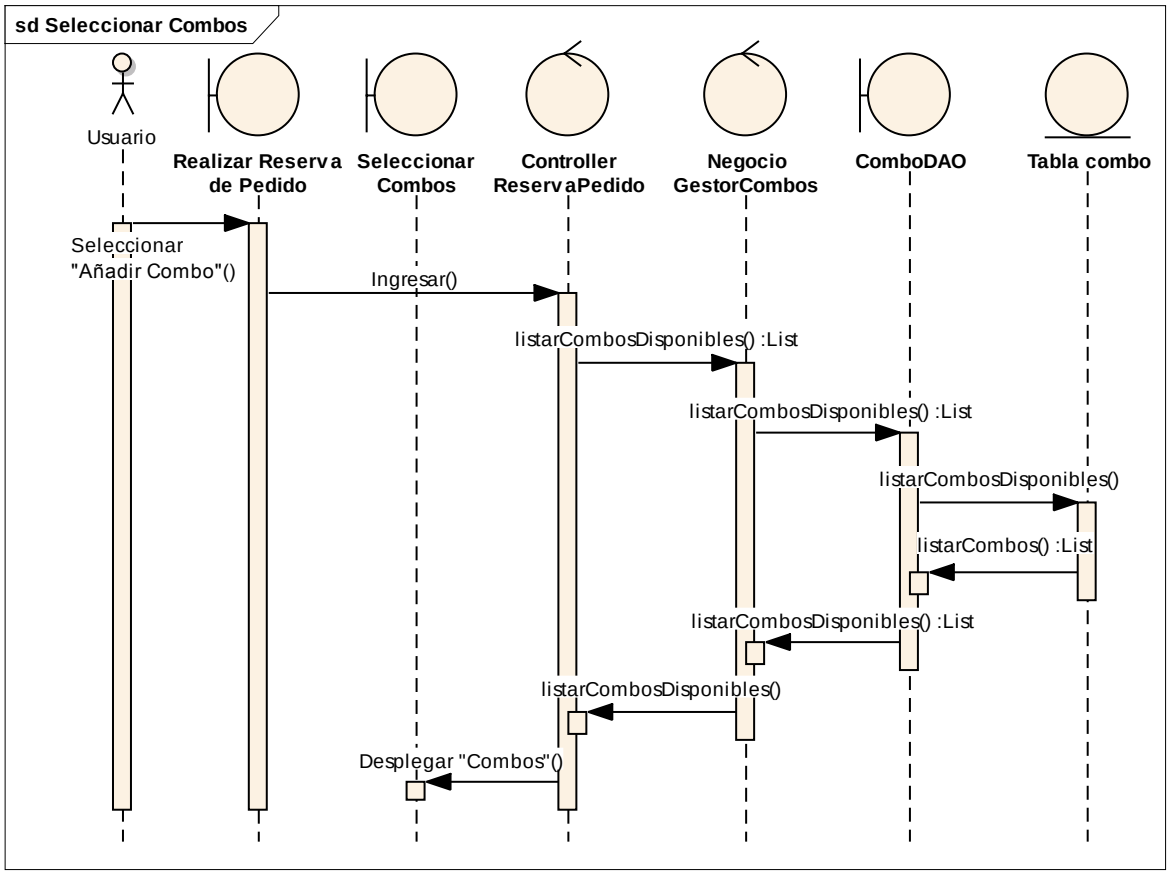


Figura 55. Diagrama de Secuencia Seleccionar Combos

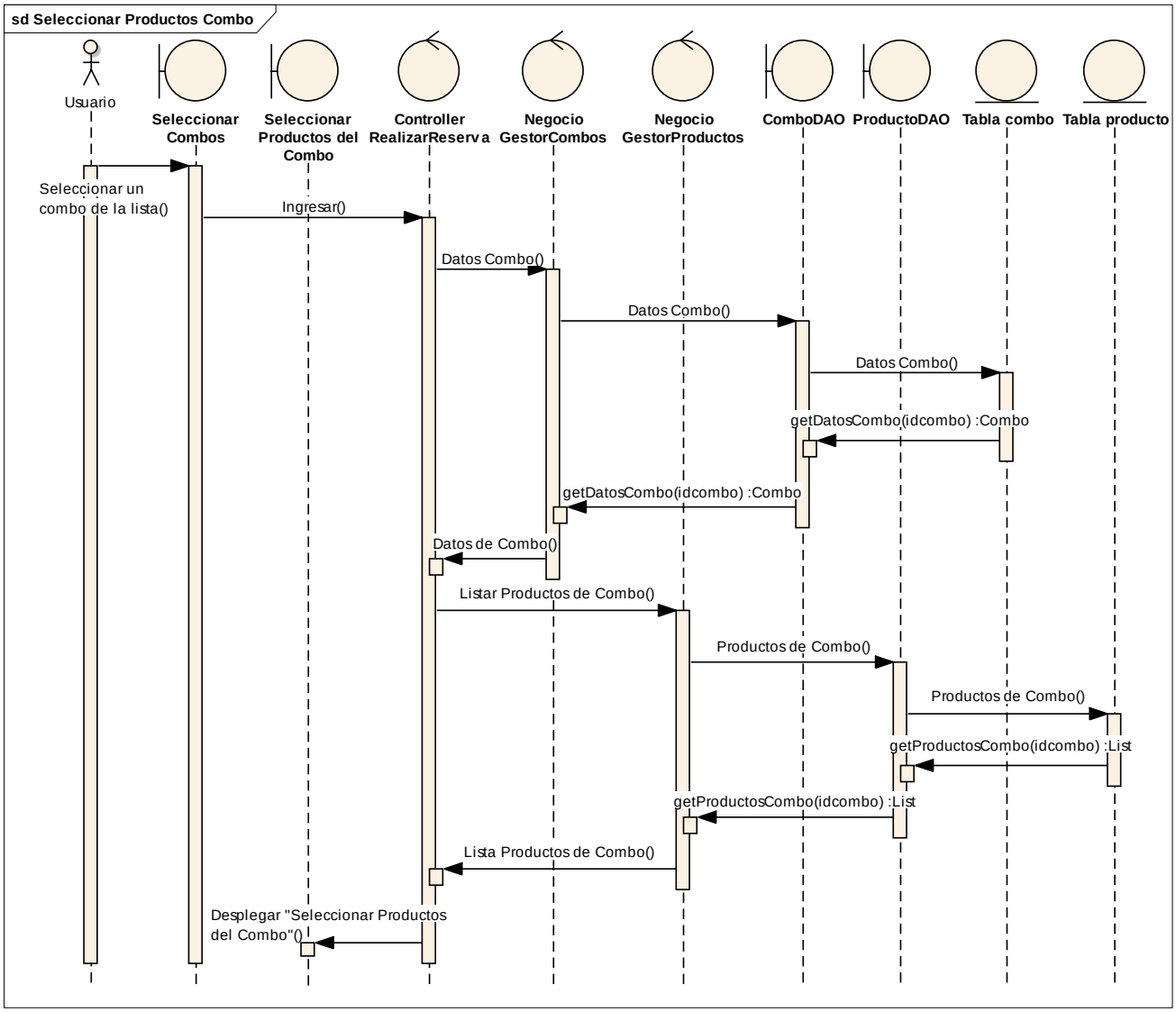


Figura 56. Diagrama de Secuencia Seleccionar Productos Combos

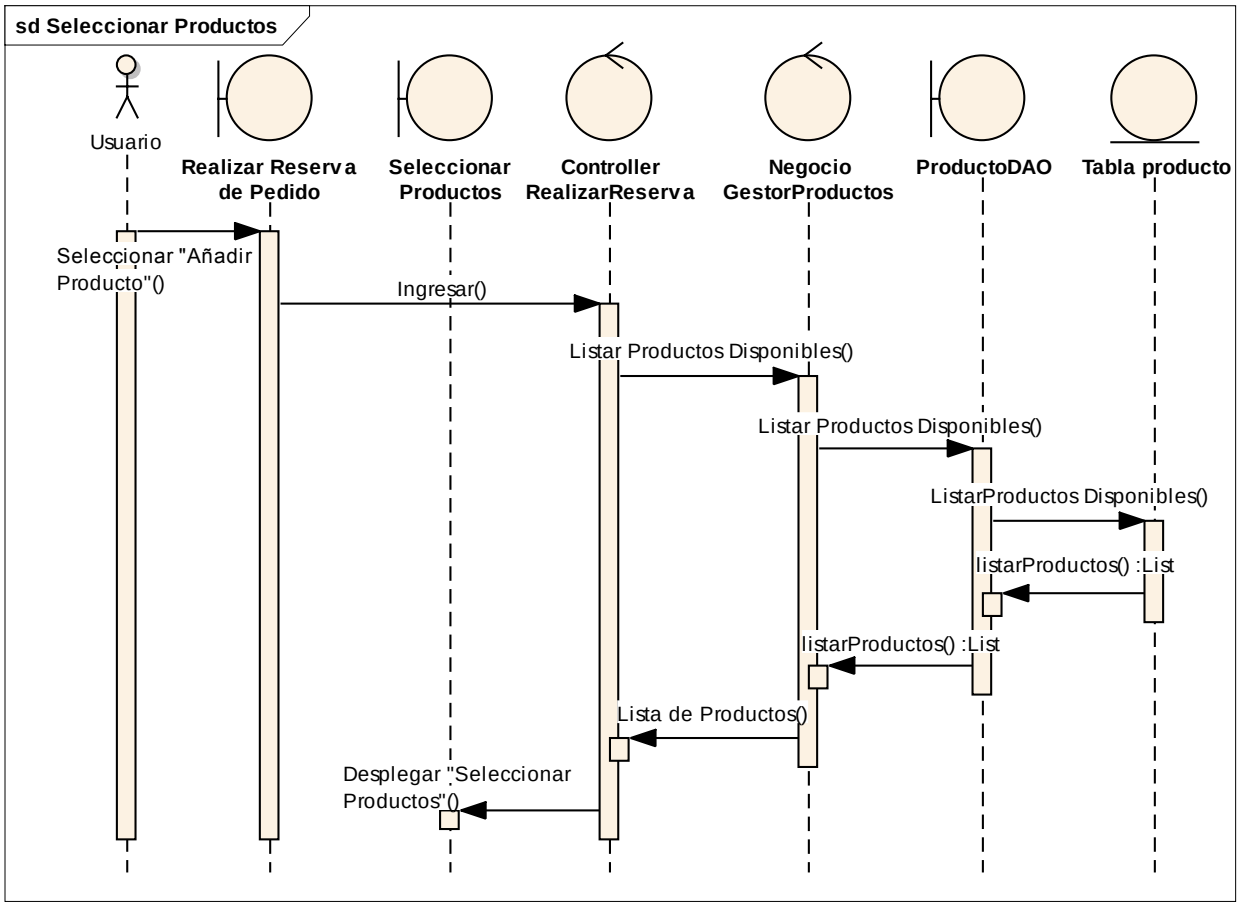


Figura 57. Diagrama de Secuencia Seleccionar Productos

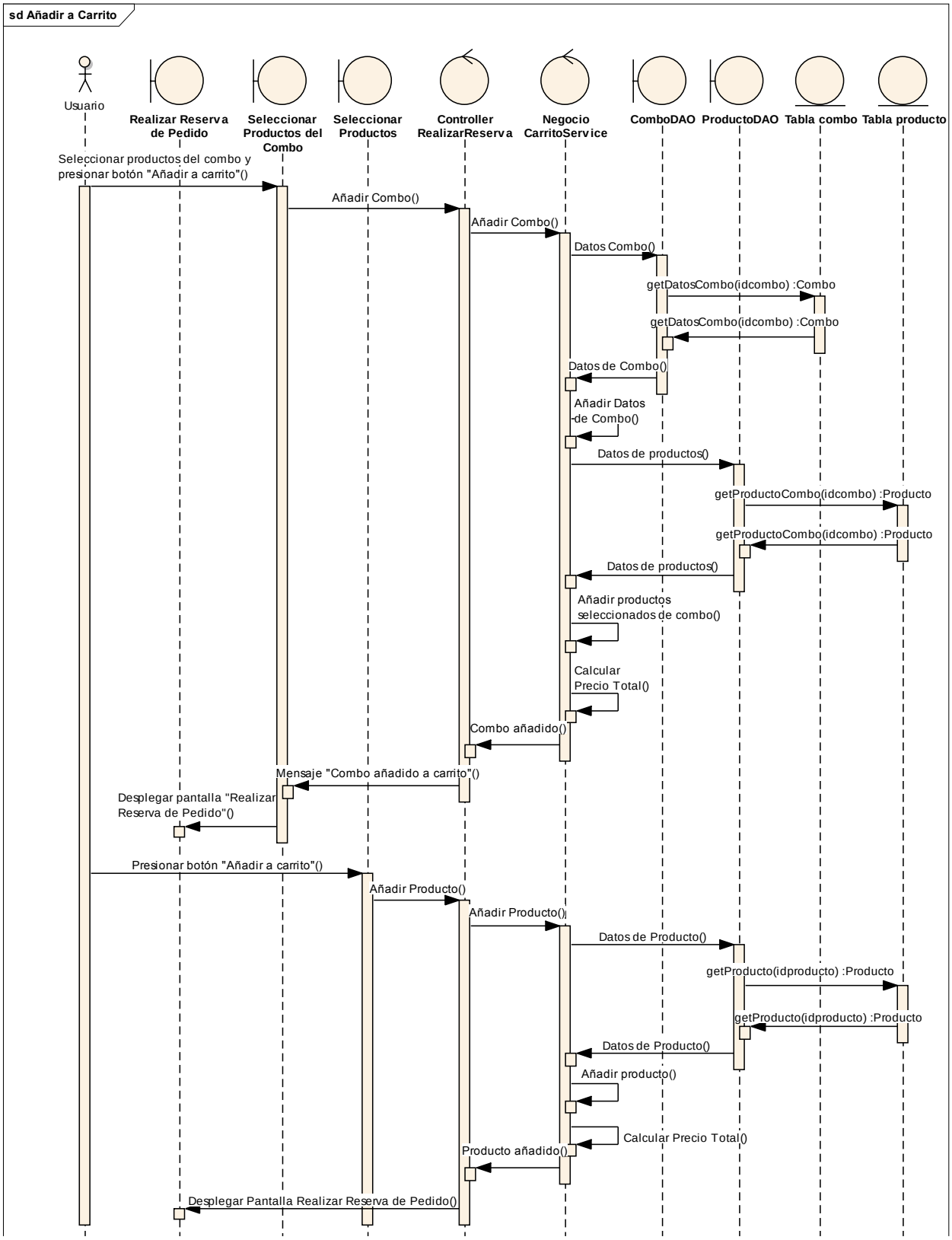


Figura 58. Diagrama de Secuencia Añadir a Carrito

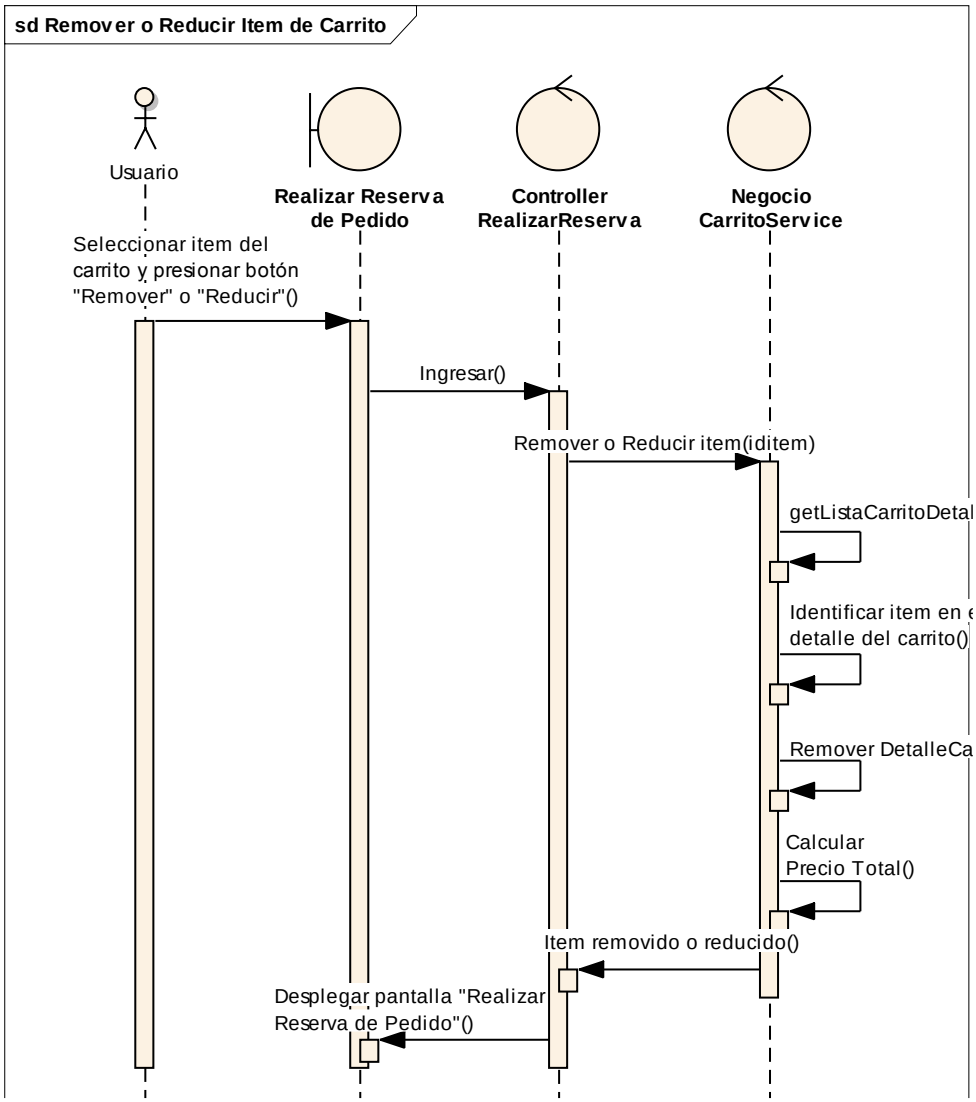


Figura 59. Diagrama de Secuencia Remover o reducir Ítem

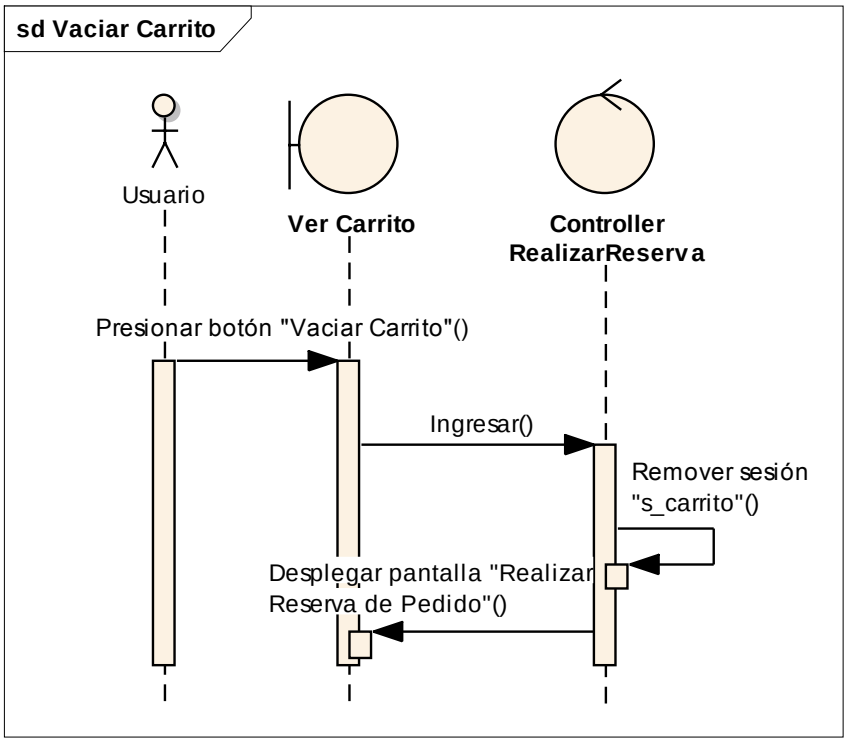


Figura 60. Diagrama de Secuencia Vaciar Carrito

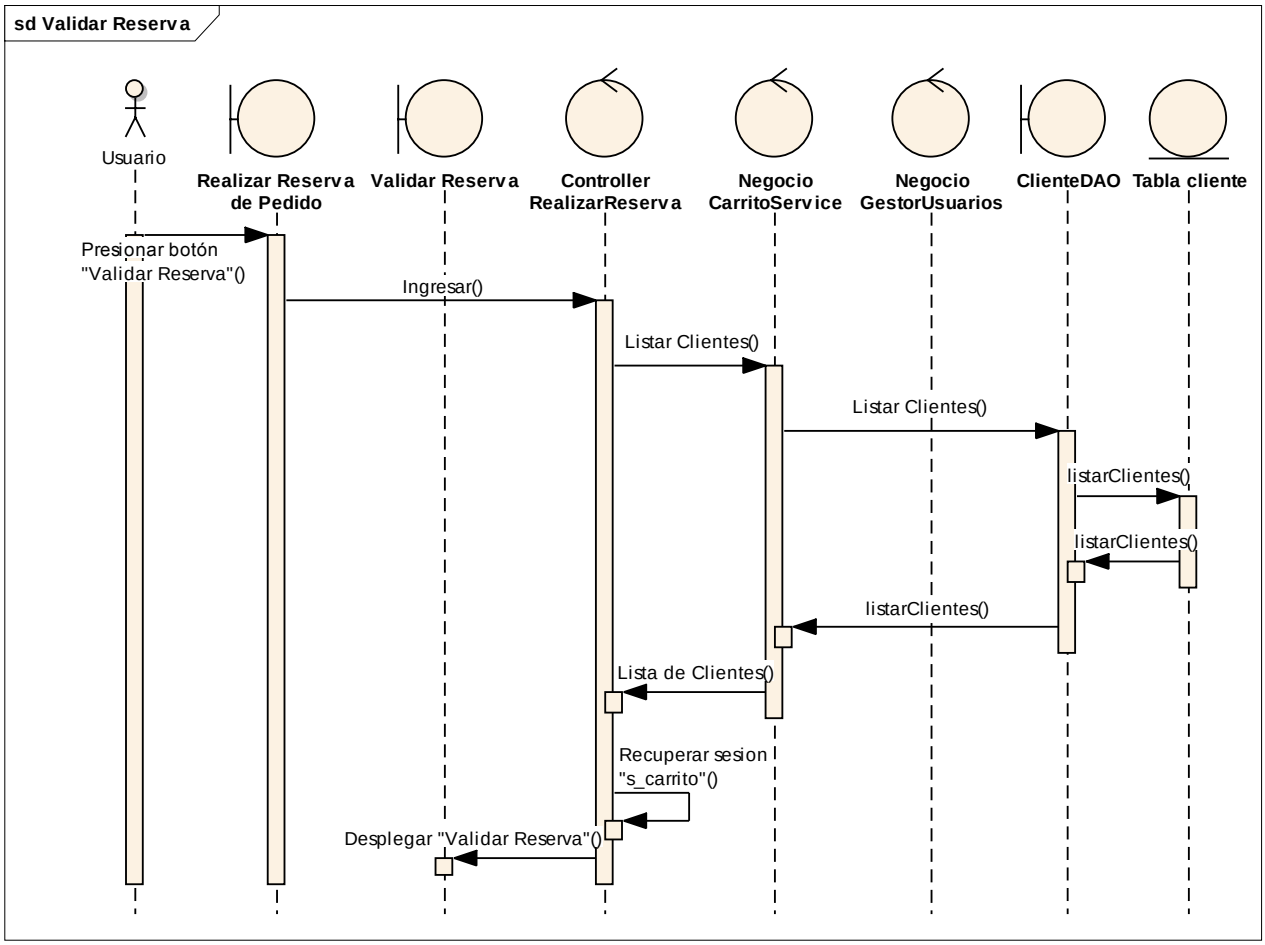


Figura 61. Diagrama de Secuencia Validar Reserva

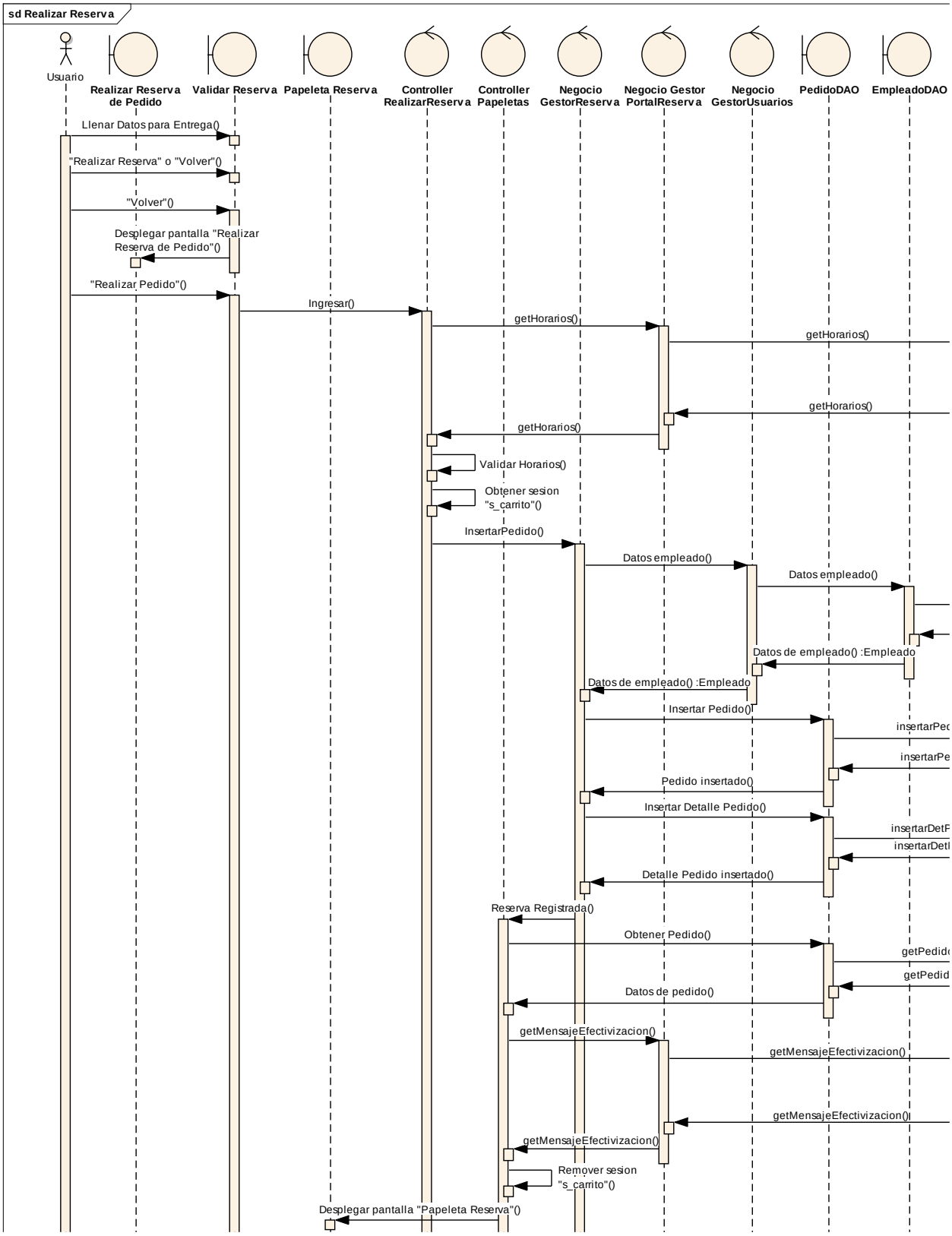


Figura 62. Diagrama de Secuencia Realizar Reserva

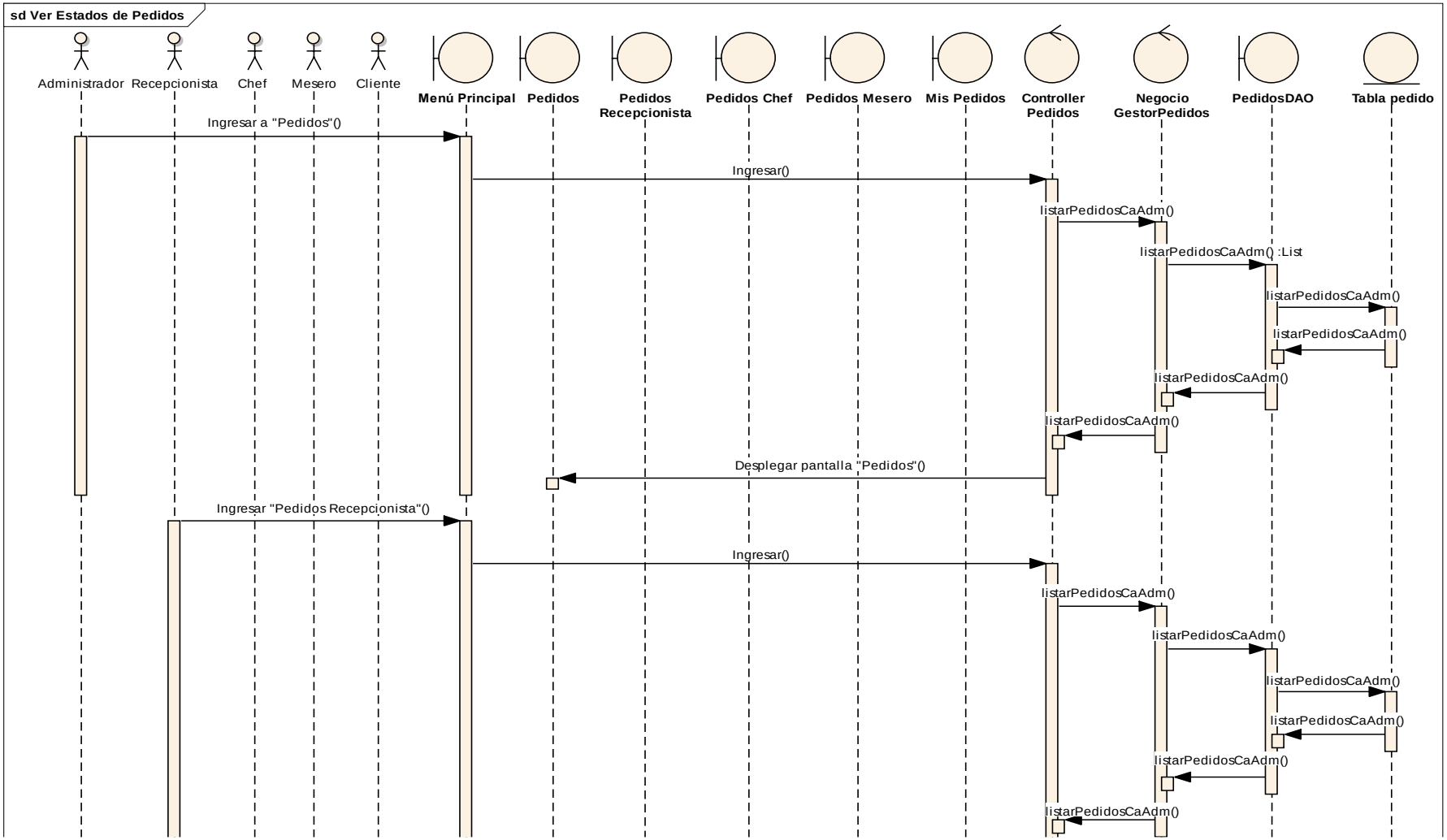


Figura 63. Diagrama de Secuencia Ver Estados de Pedidos

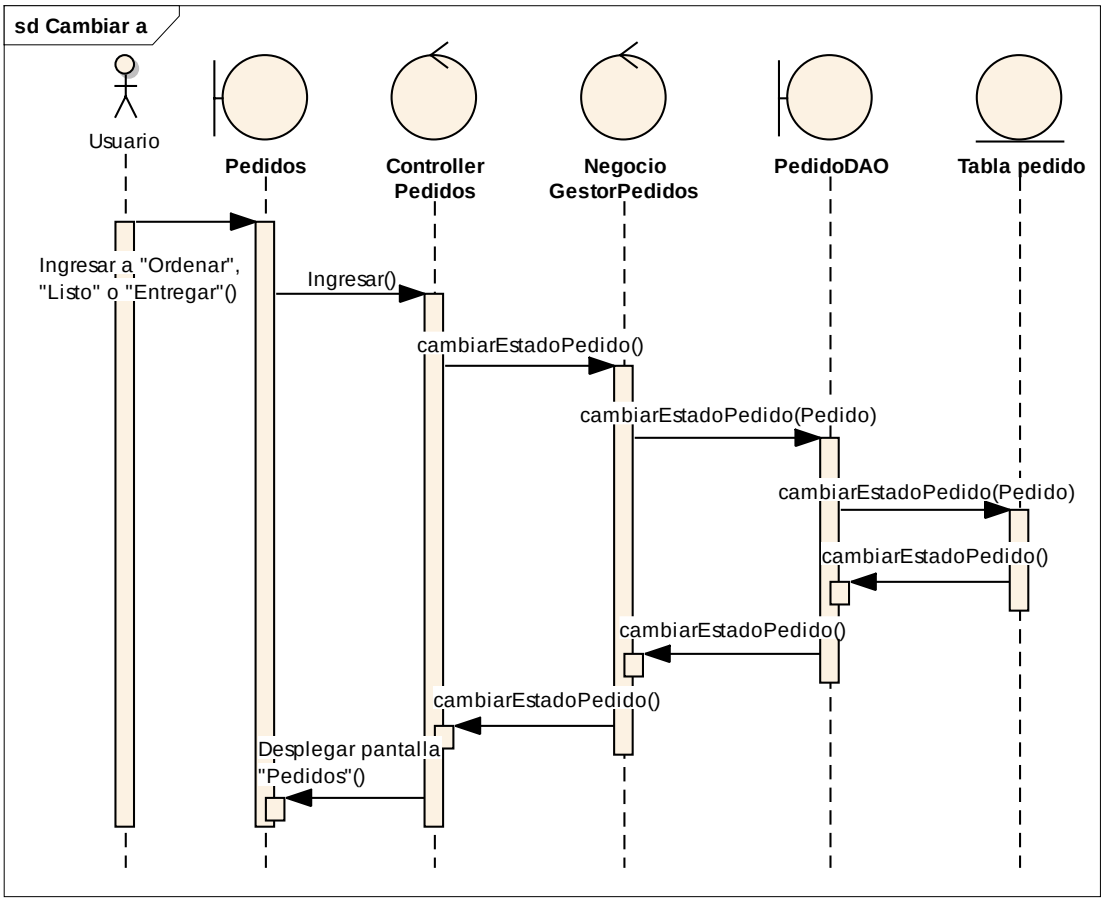


Figura 64. Diagrama de Secuencia Cambiar a

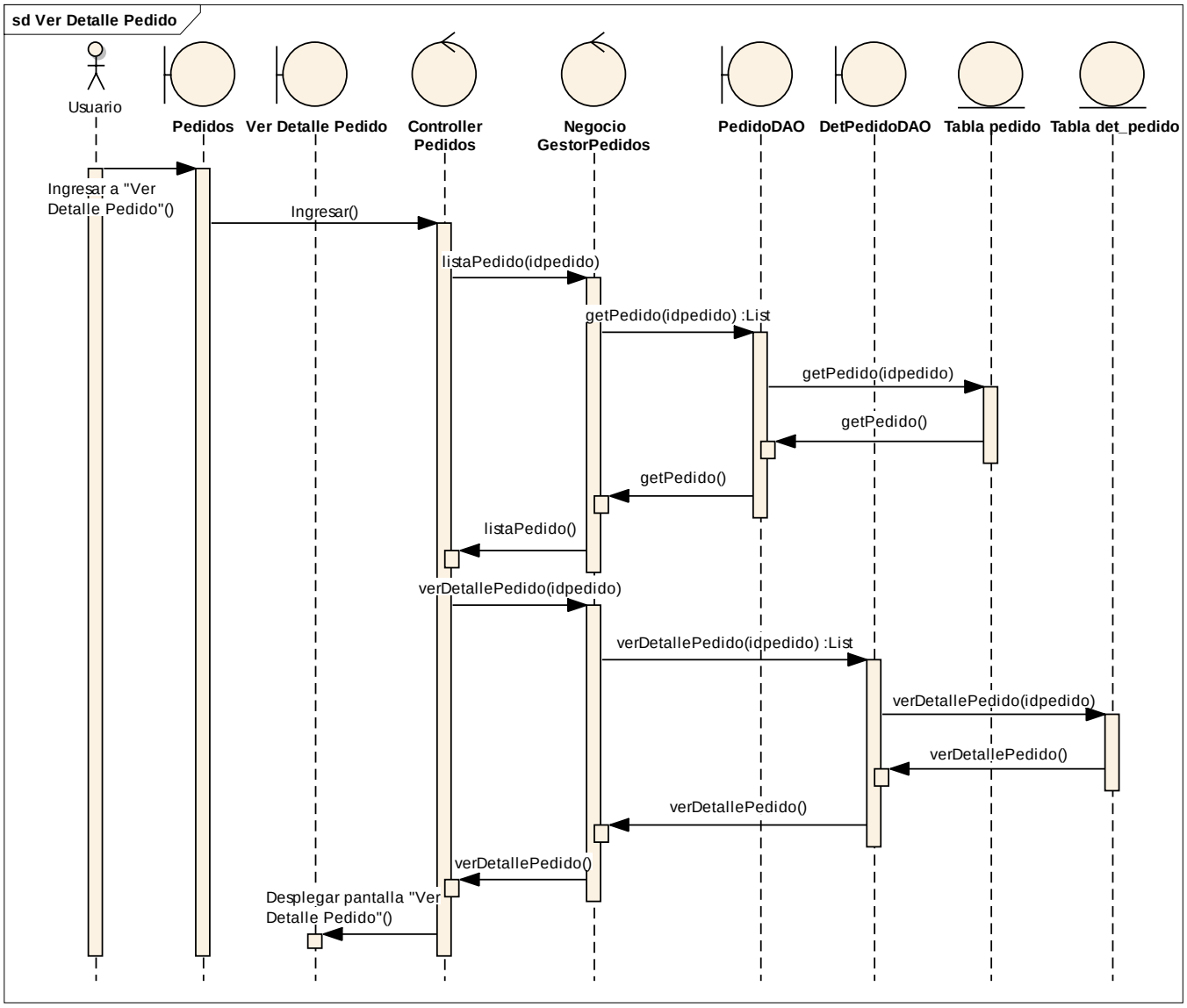


Figura 65. Diagrama de Secuencia Ver Detalle Pedido

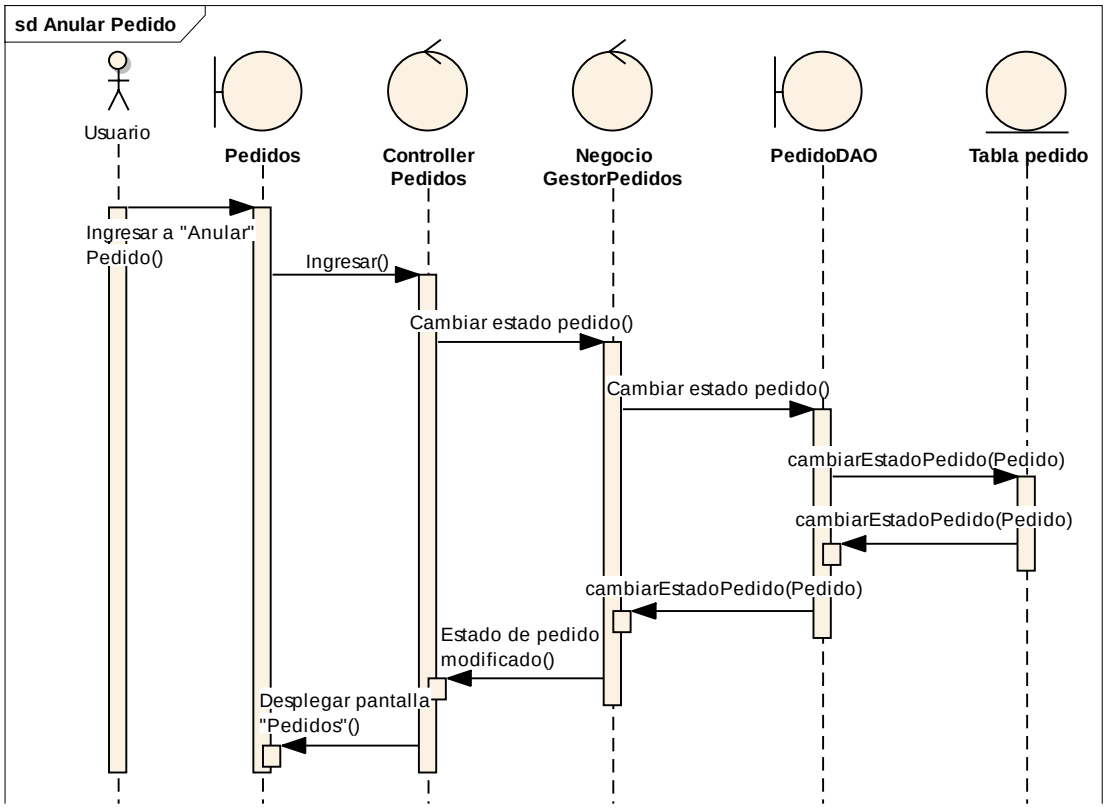


Figura 66. Diagrama de Secuencia Anular Pedido

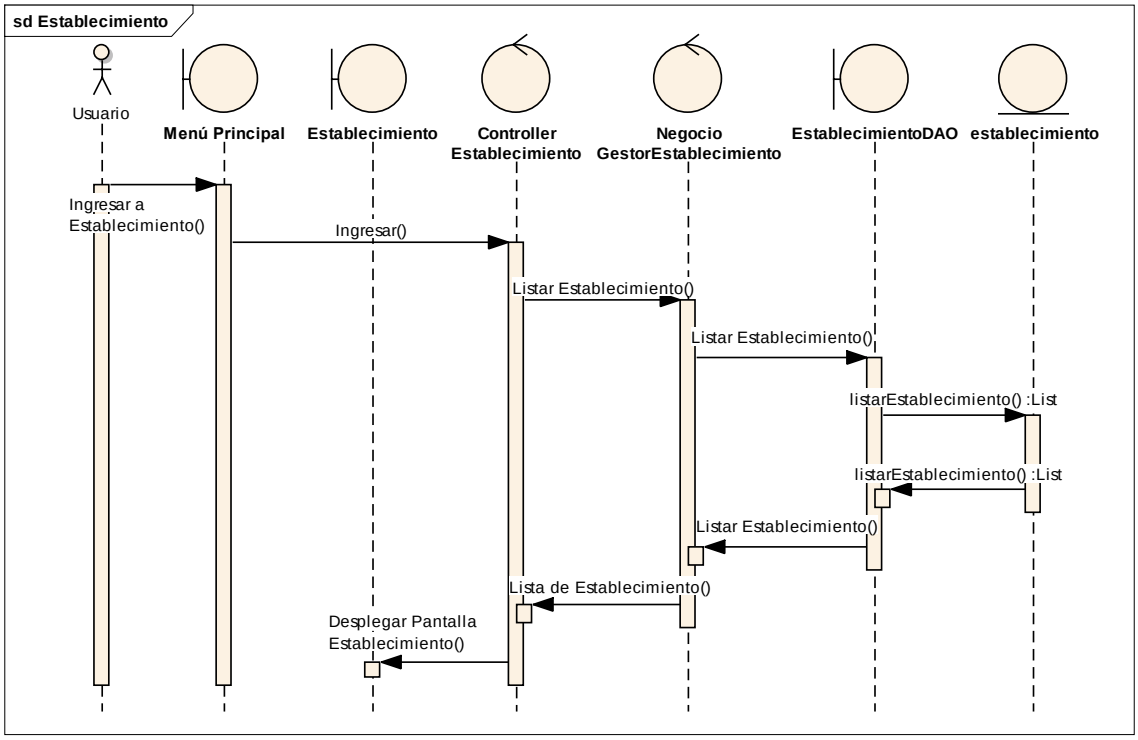


Figura 67. Diagrama de Secuencia Establecimiento

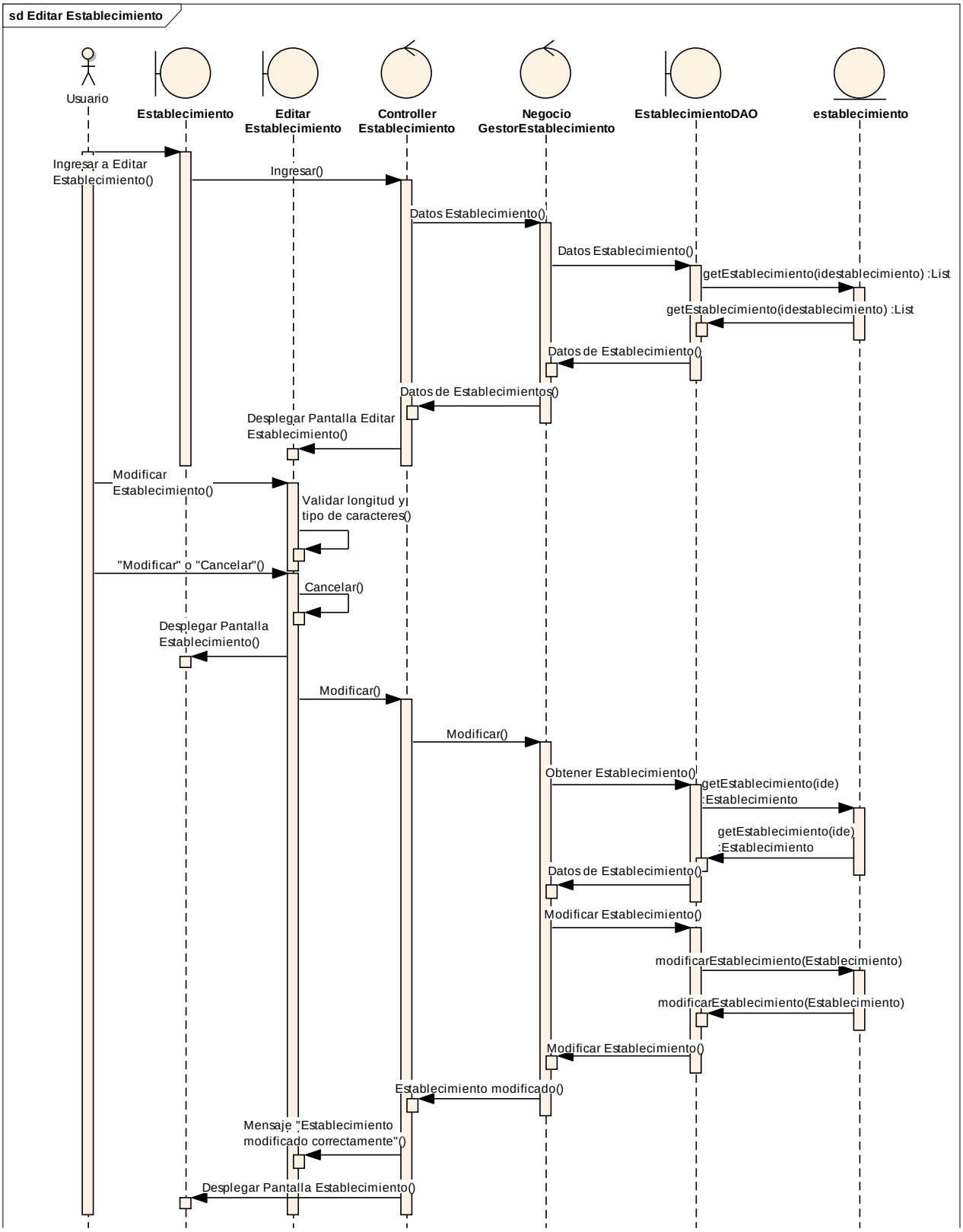


Figura 68. Diagrama de Secuencia Editar Establecimiento

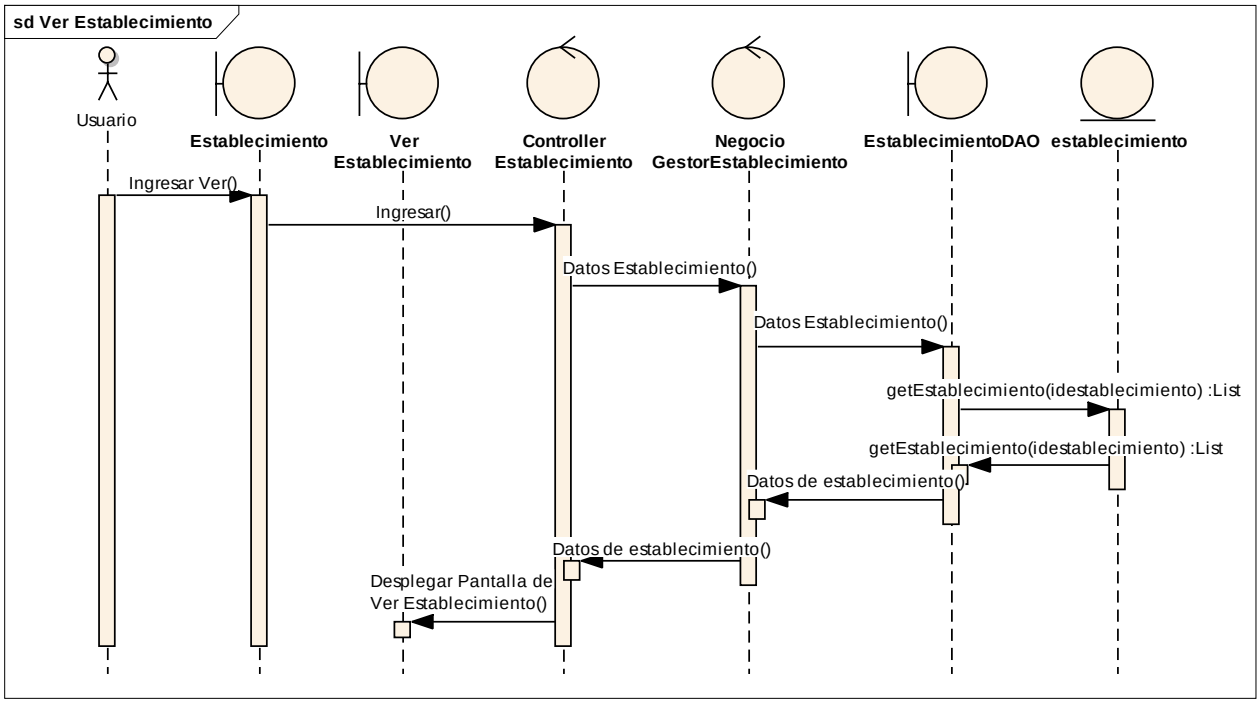


Figura 69. Diagrama de Secuencia Ver Establecimiento

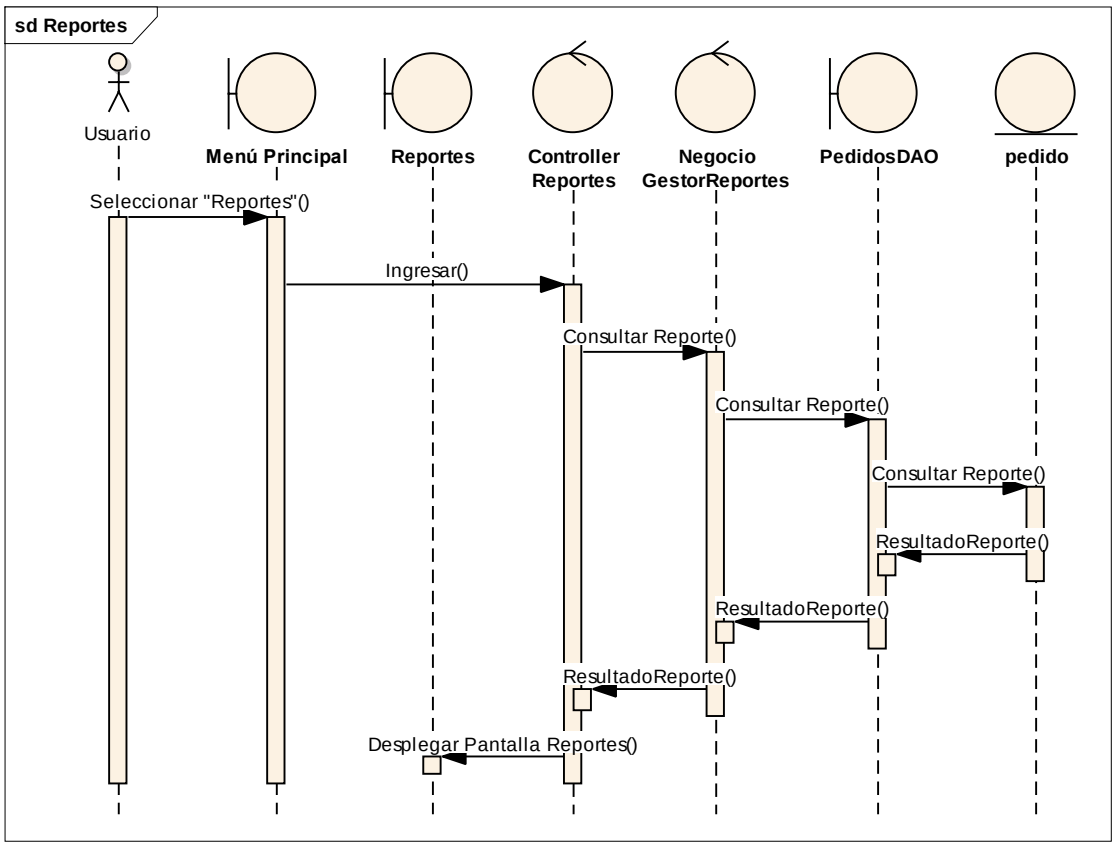


Figura 70. Diagrama de Secuencia Reportes

4.10.2 Diagrama de actividad

4.10.2.1 Introducción

Mediante el uso de los diagramas de actividad podemos modelar el flujo de control entre actividades del sistema. La idea es generar una especie de diagrama Pert, en el que se puede ver el flujo de actividades que tienen lugar a lo largo del tiempo, así como las tareas concurrentes que pueden realizarse a la vez. Gráficamente es un conjunto de arcos y nodos. Desde un punto de vista conceptual, El diagrama de actividad muestra como fluye el control de unas clases a otras con la finalidad de culminar con un flujo de control total que se corresponde con la consecución de un proceso más completo, Por este motivo, aparecerán acciones y actividades correspondientes a distintas clases, colaborando todas ellas para conseguir un mismo fin.

4.10.2.2 Propósito

- ✓ Comprender la estructura del sistema deseado para la organización
- ✓ Identificar posibles mejoras en el sistema
- ✓ Modelar aspectos dinámicos del sistema
- ✓ Mostrar operaciones que se pasan entre objetivos
- ✓ Mostrar flujos de actividades

4.10.2.3 Alcances

- ✓ Describe los procesos del sistema y de los clientes

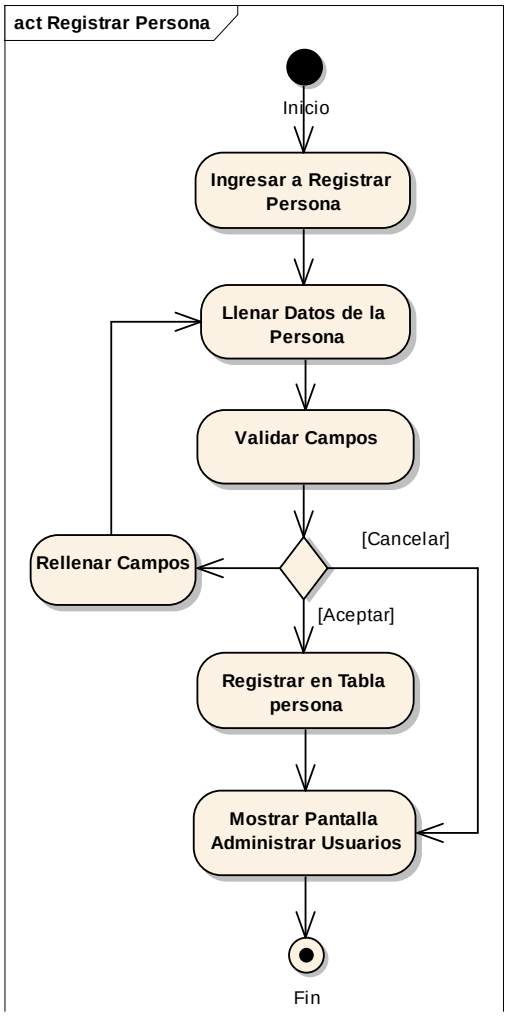


Figura 71. Registrar Persona

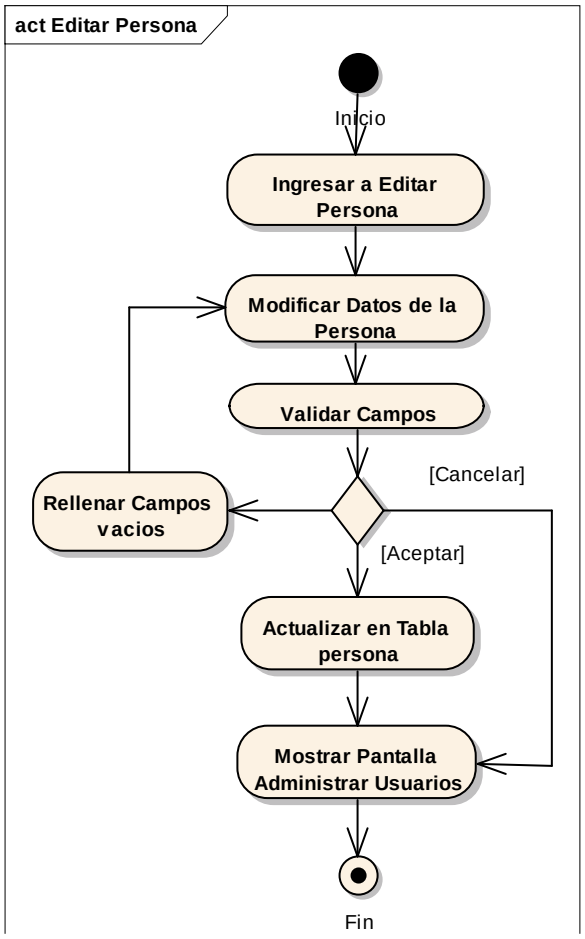


Figura 72. Editar Persona

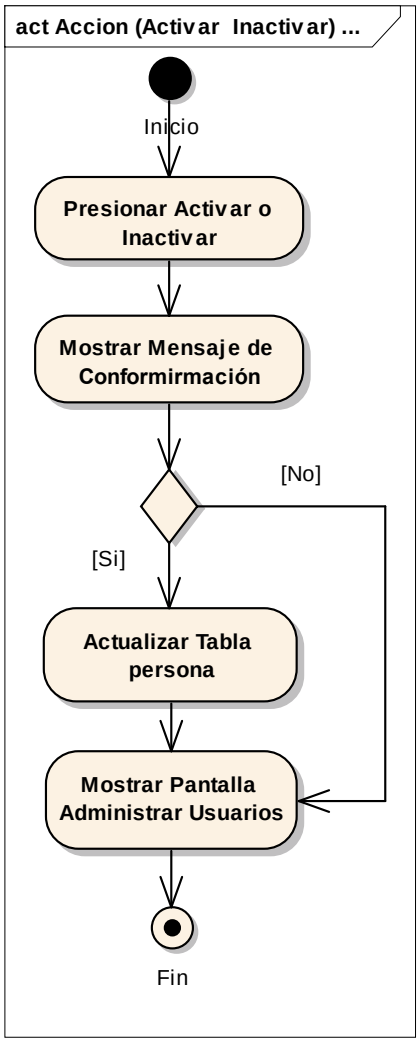


Figura 73. Acción (Activar-Inactivar) Persona

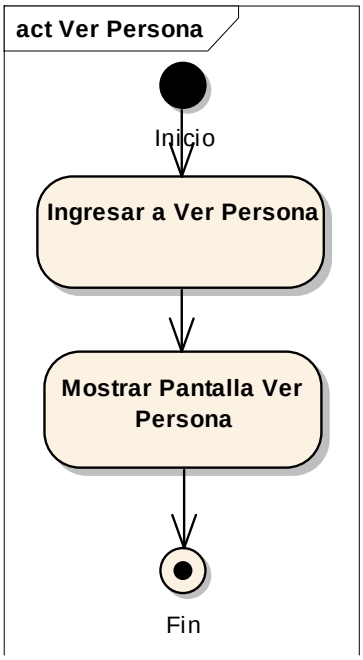


Figura 74. Ver Persona

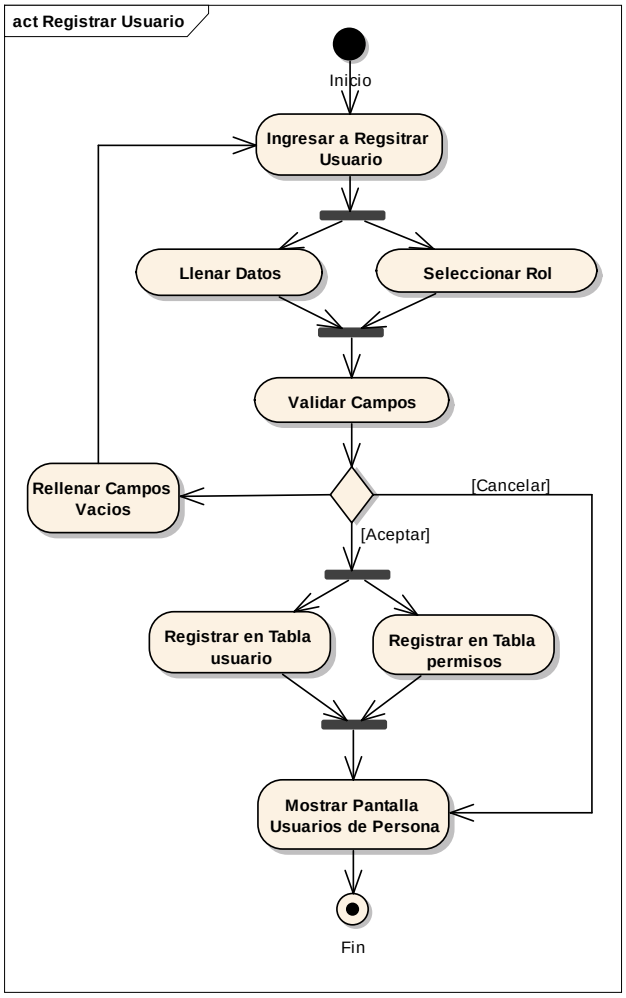


Figura 75 Registrar Usuario

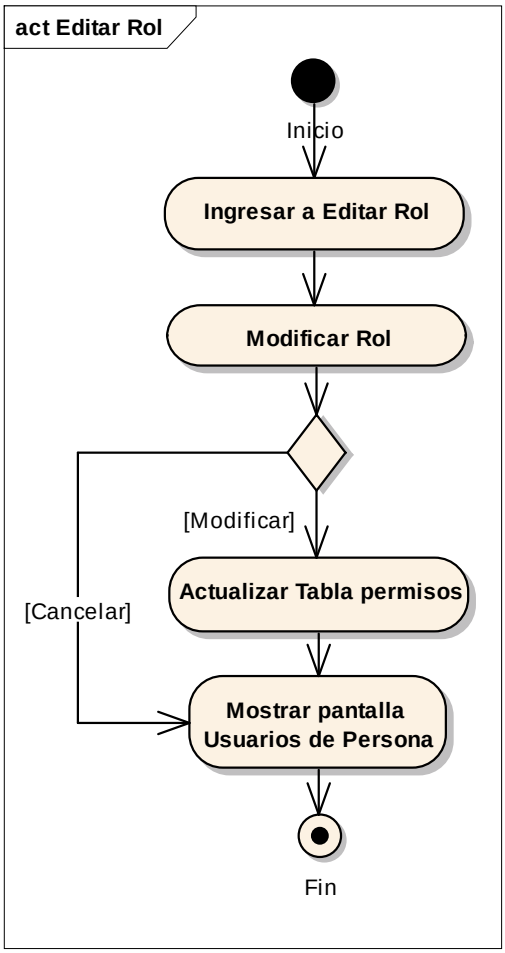


Figura 76. Editar Rol Usuario

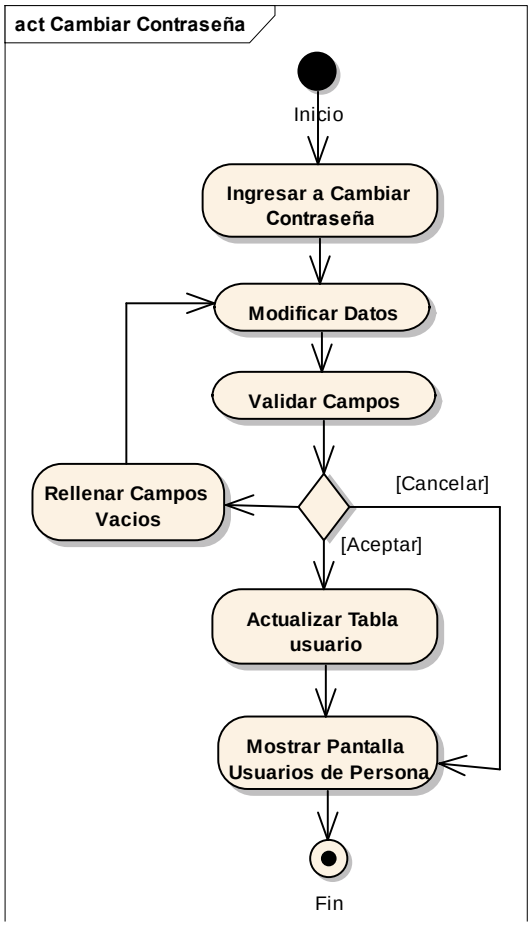


Figura 77. Cambiar Contraseña

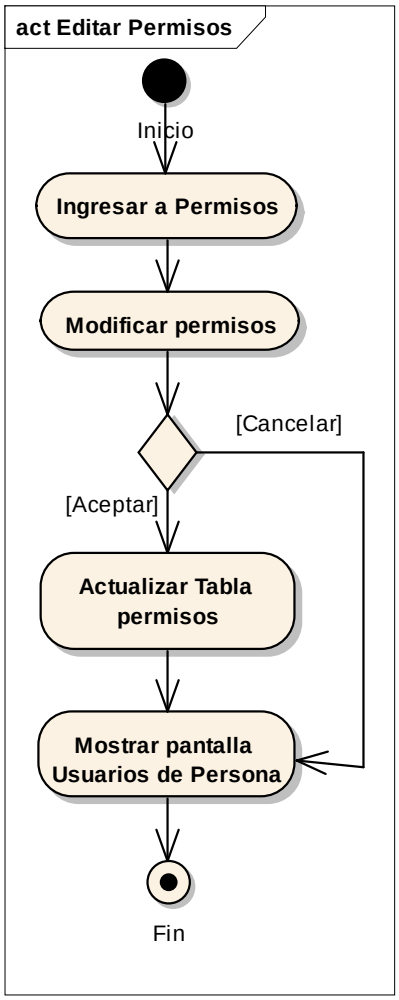


Figura 78. Editar Permisos

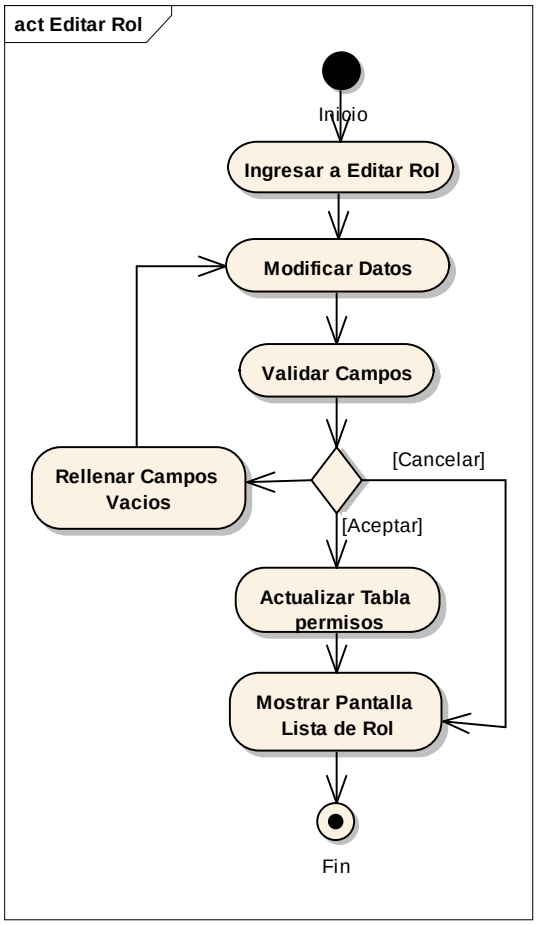


Figura 79. Editar Rol

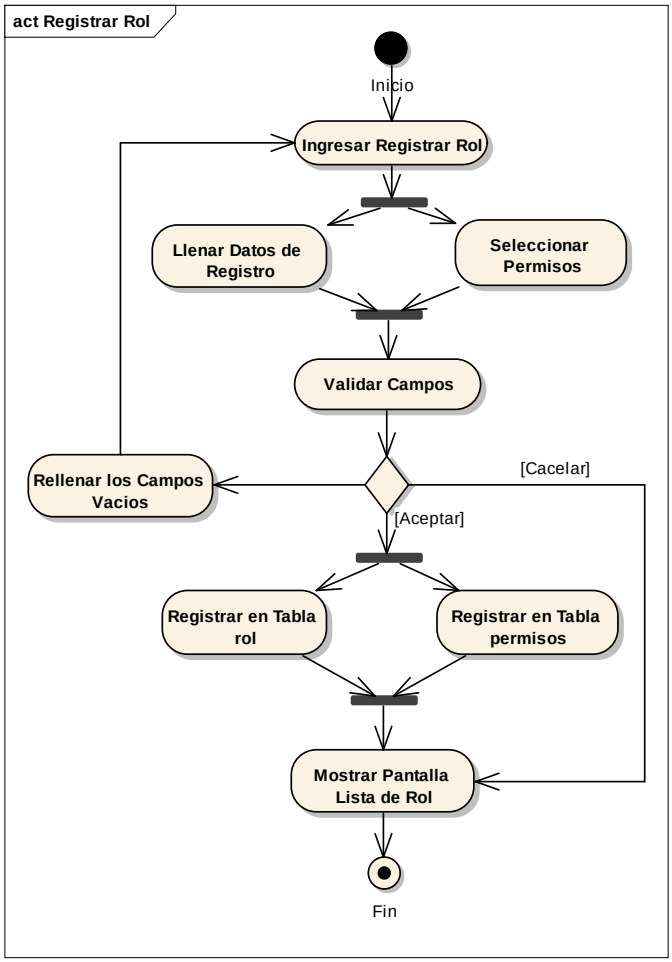


Figura 80. Registrar Rol

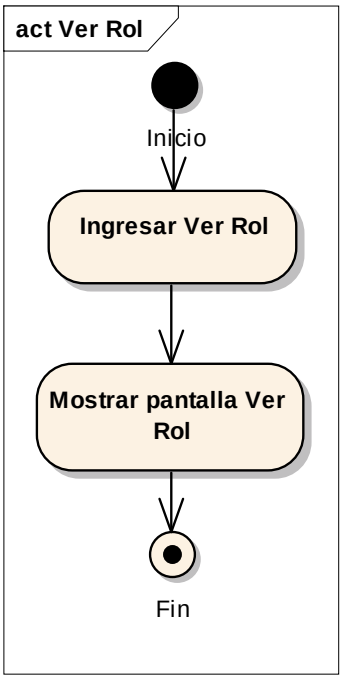


Figura 81. Ver Rol

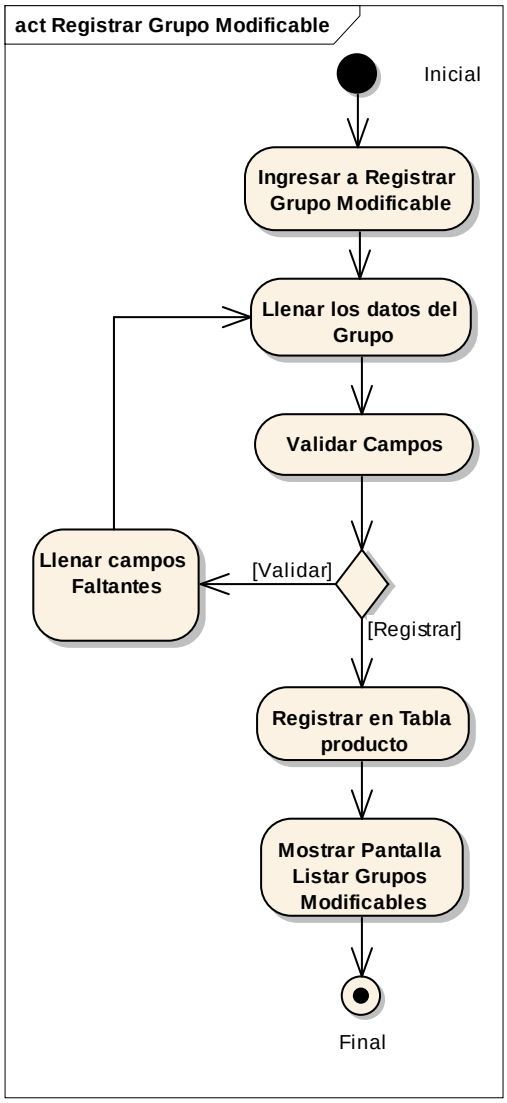


Figura 82. Registrar Grupo Intercambiable

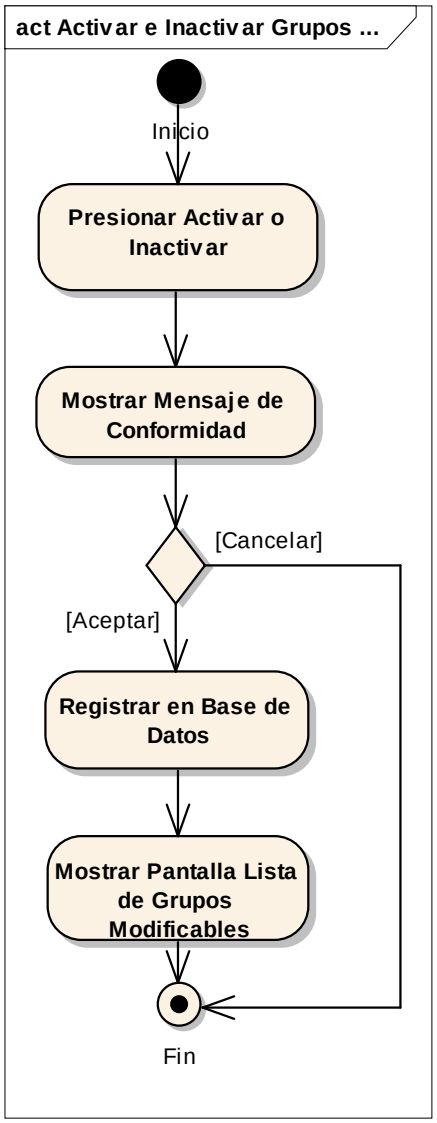


Figura 83. Inactivar y Activar Grupo Modificable

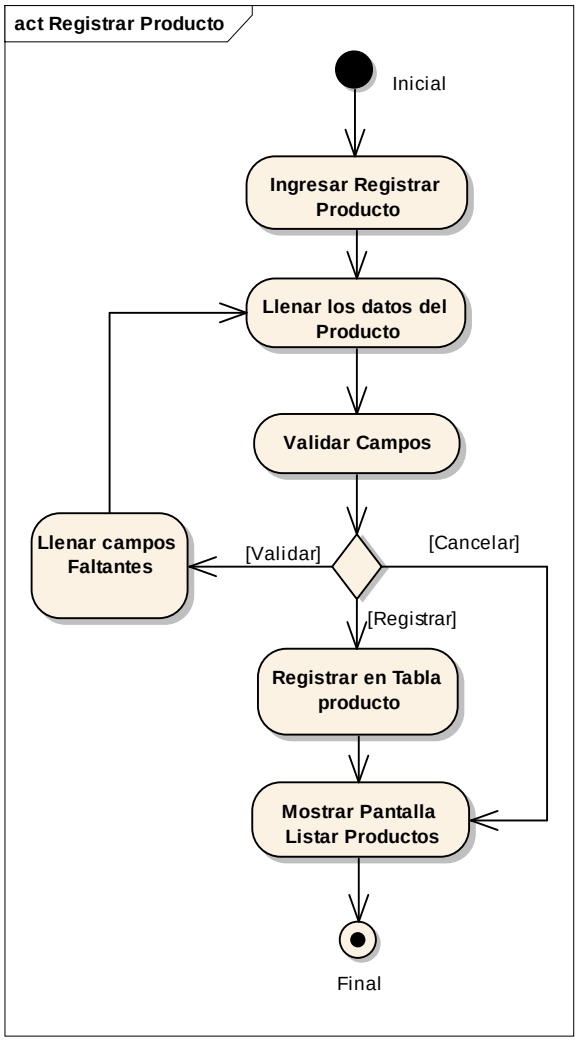


Figura 84. Registrar Producto

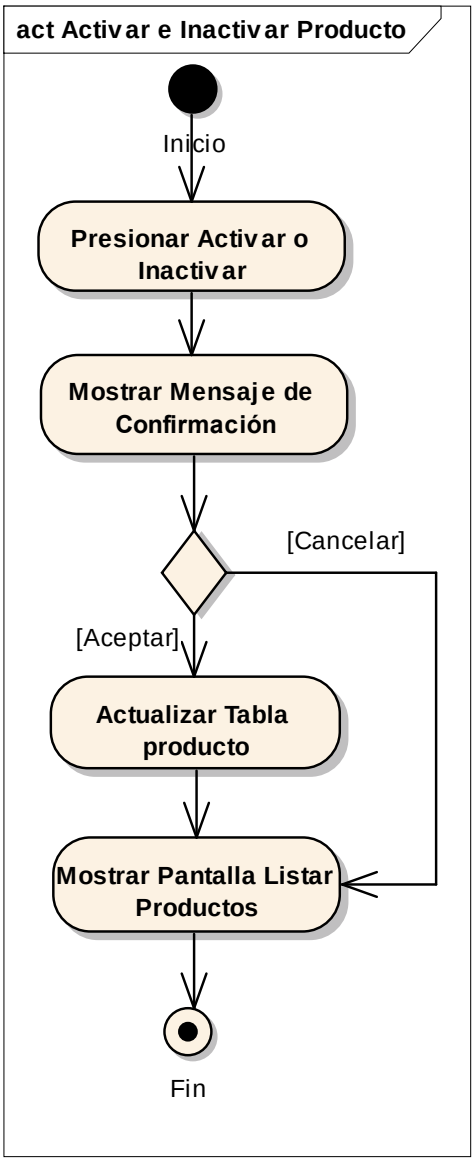


Figura 85. Acción (Activar-Inactivar) Producto

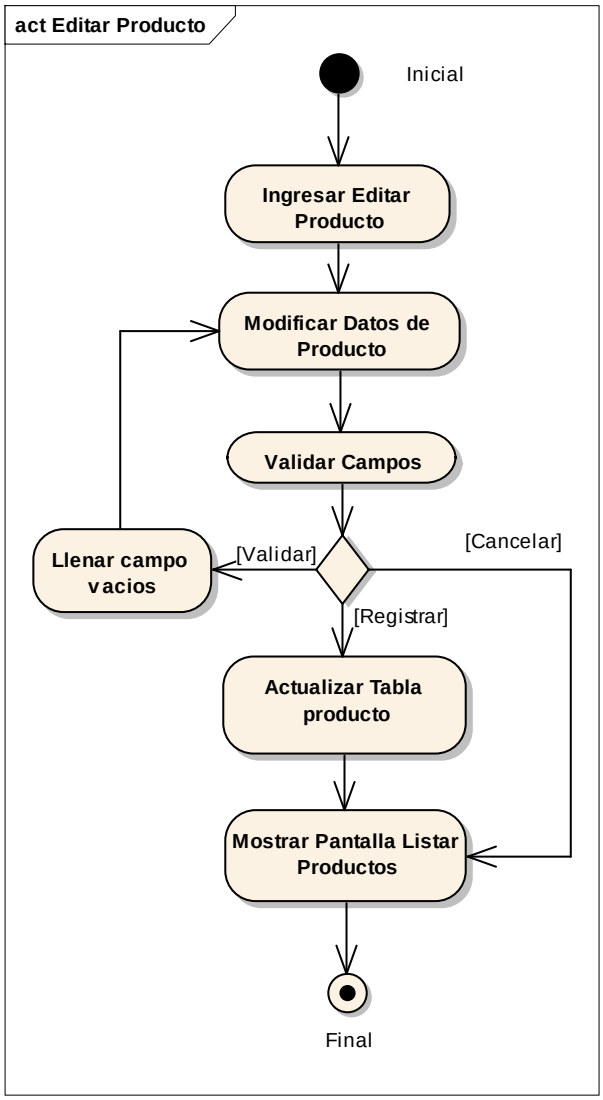


Figura 86. Editar Producto

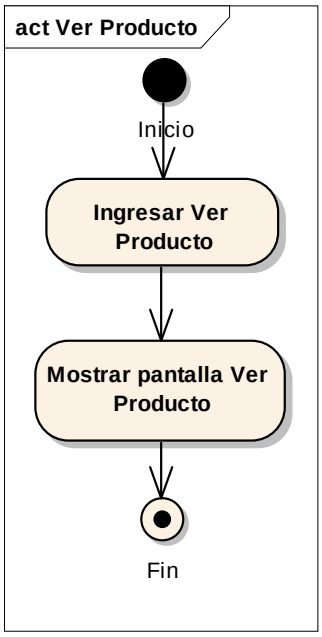


Figura 87. Ver Producto

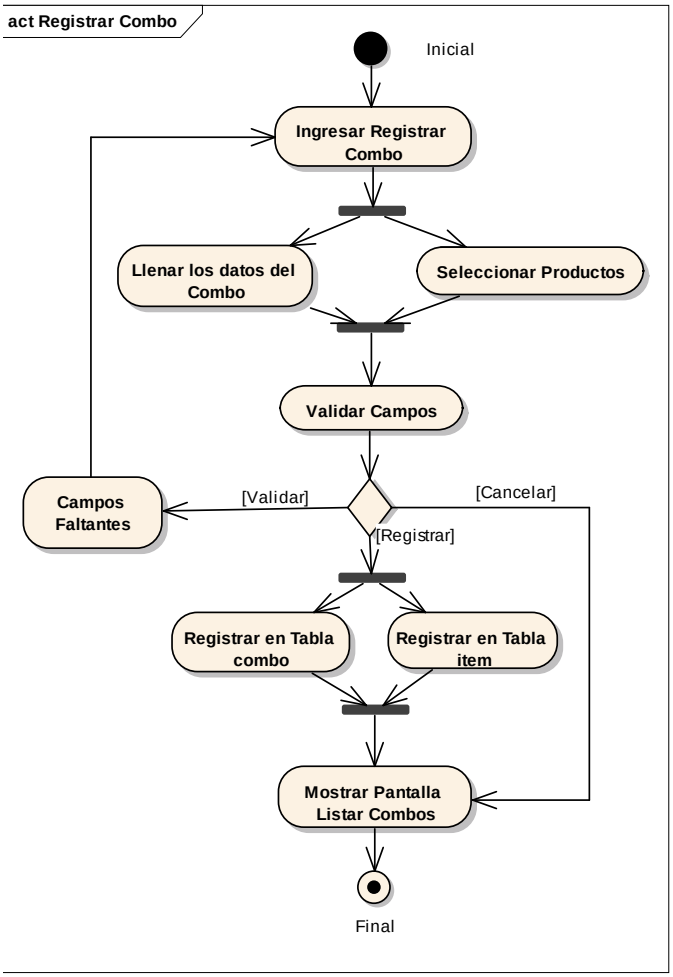


Figura 88. Registrar Combo

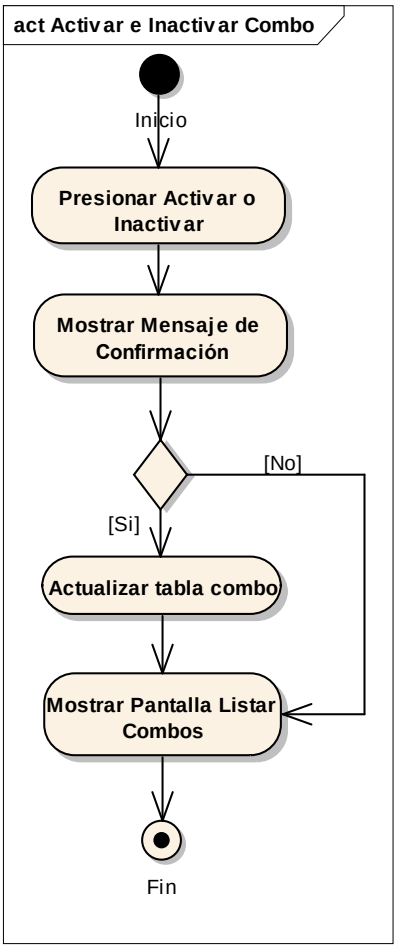


Figura 89. Acción (Activar-Inactivar) Combo

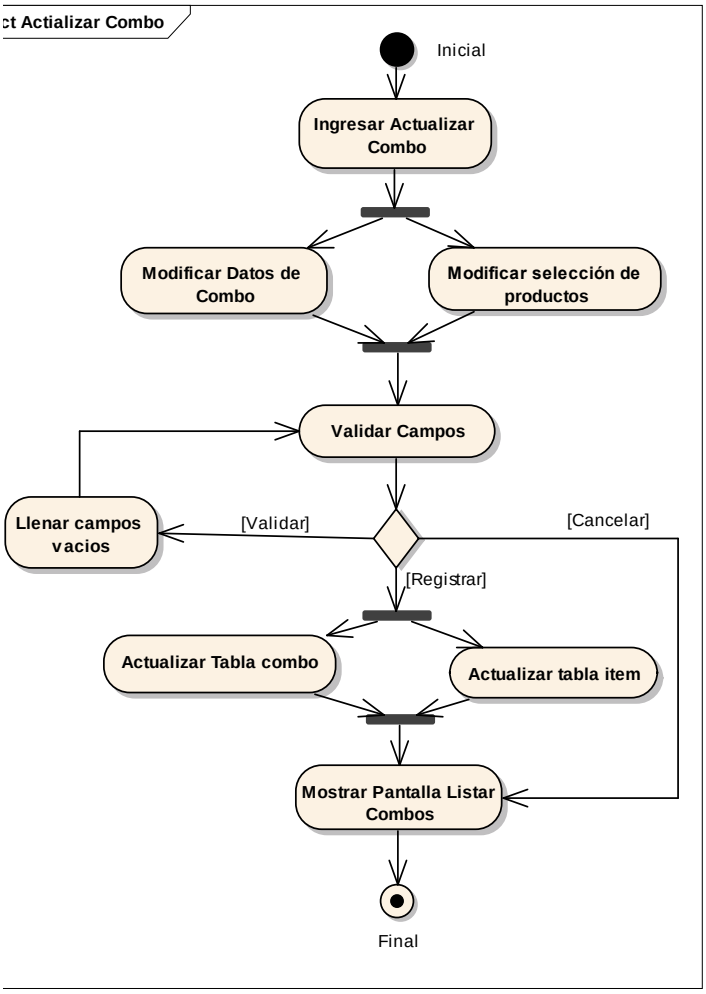


Figura 90. Actualizar Combo

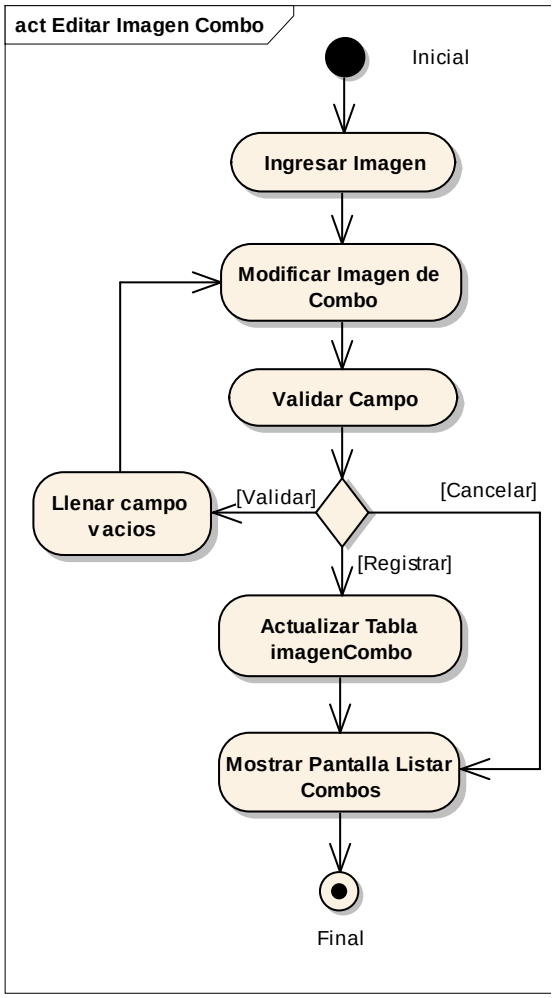


Figura 91. Editar Imagen Combo

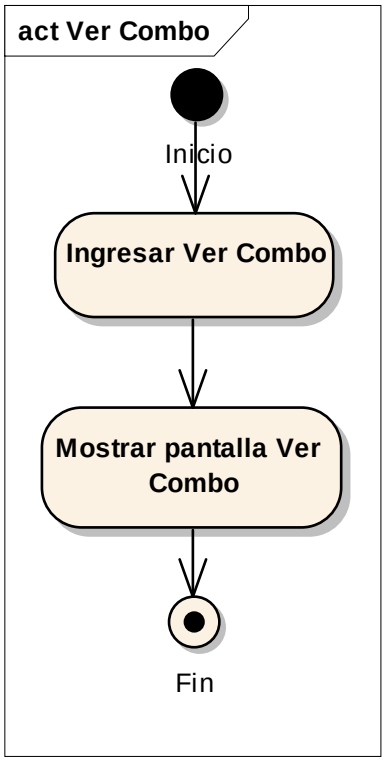


Figura 92. Ver Combo

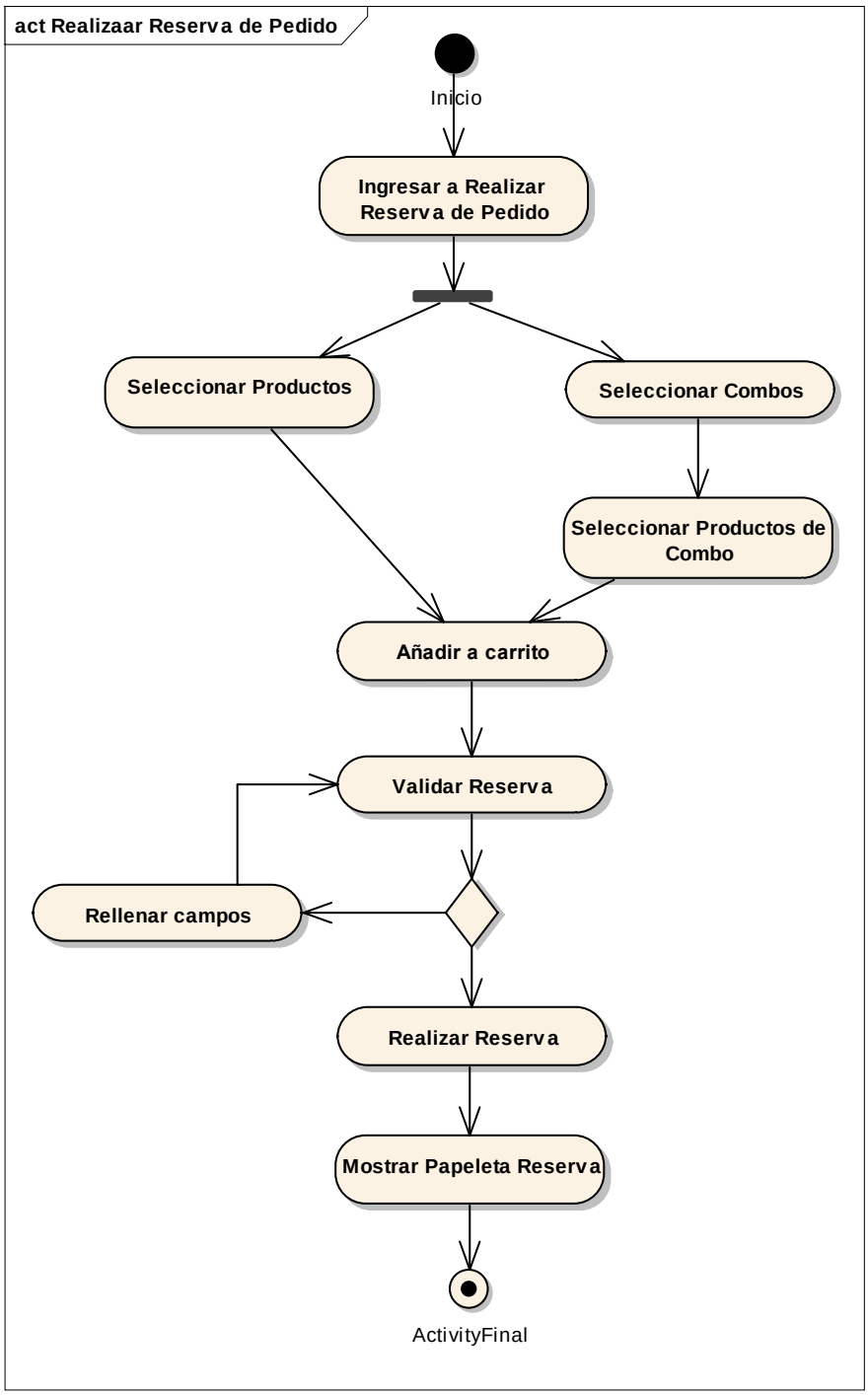


Figura 93. Realizar Reserva de Pedido

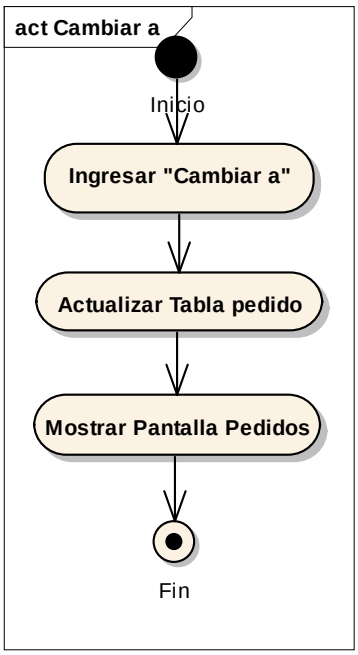


Figura 94. Cambiar a

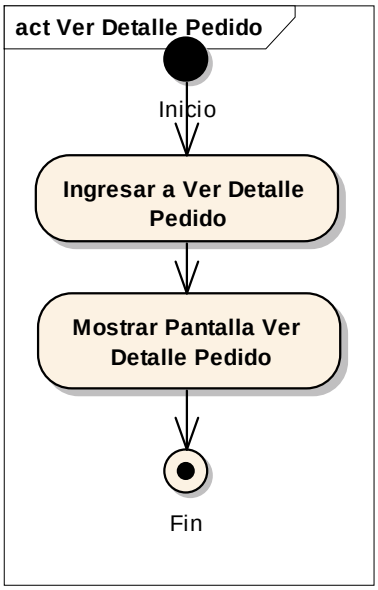


Figura 95. Detalle Pedido

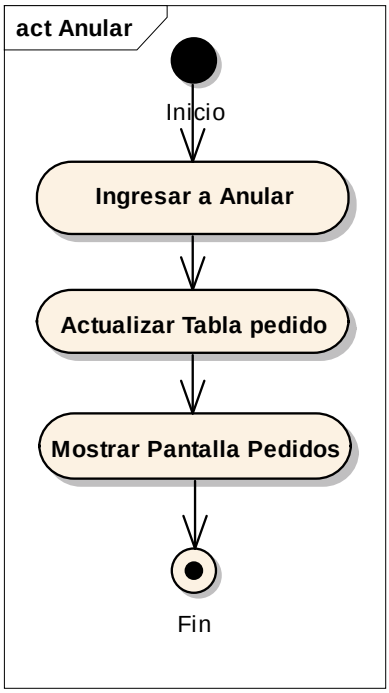


Figura 96. Anular

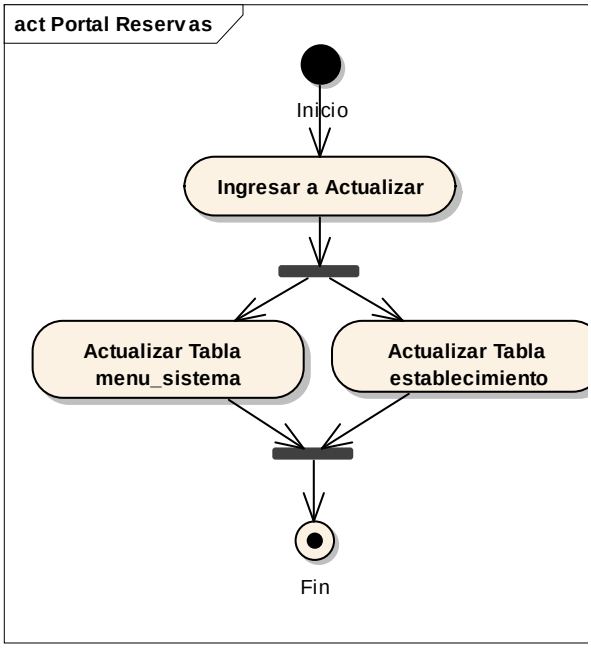


Figura 97. Portal Reservas

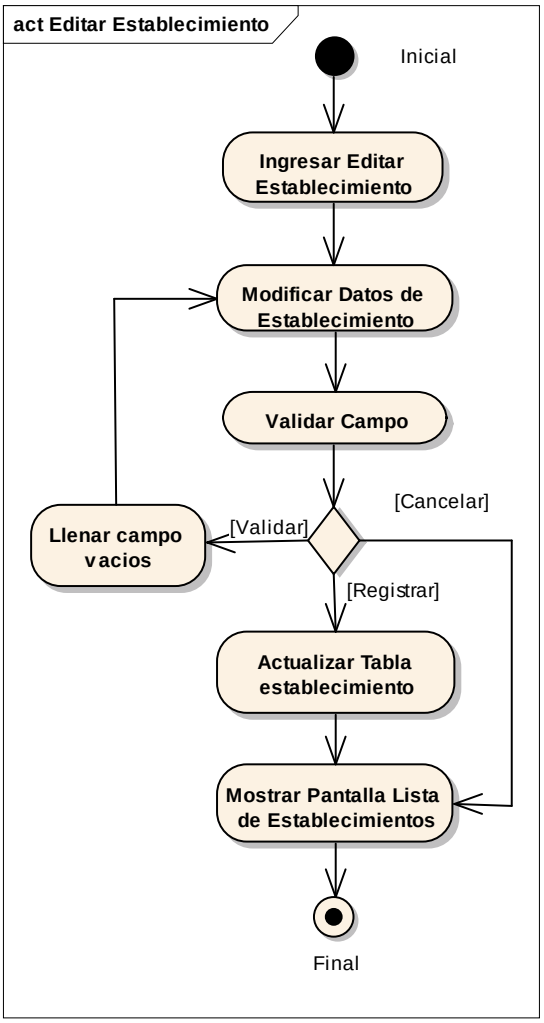


Figura 98. Editar Establecimiento

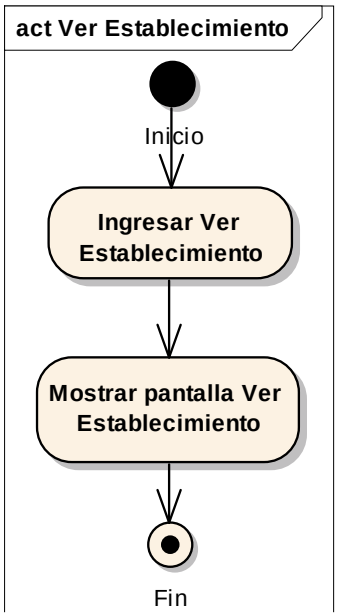


Figura 99. Ver Establecimiento

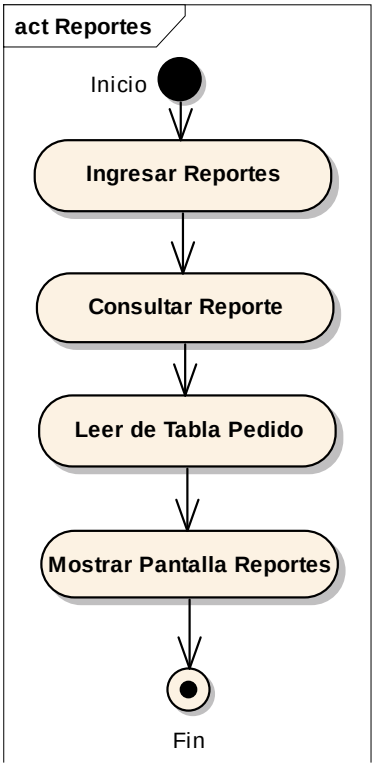


Figura 100. Reportes

4.10.3 Pantallas de interfaz de usuario

4.10.3.1 Introducción

Se trata de la presentación de las pantallas del sistema, que permiten al usuario visualizar de forma precisa lo que provee el sistema.

4.10.3.2 Propósito

Presentar las principales pantallas para que el usuario tenga una idea de la interfaz que presentará el sistema.

Pantalla presentación y acceso al sistema.

Figura 101. Pantalla Presentación y acceso al sistema

Pantalla menú principal del sistema

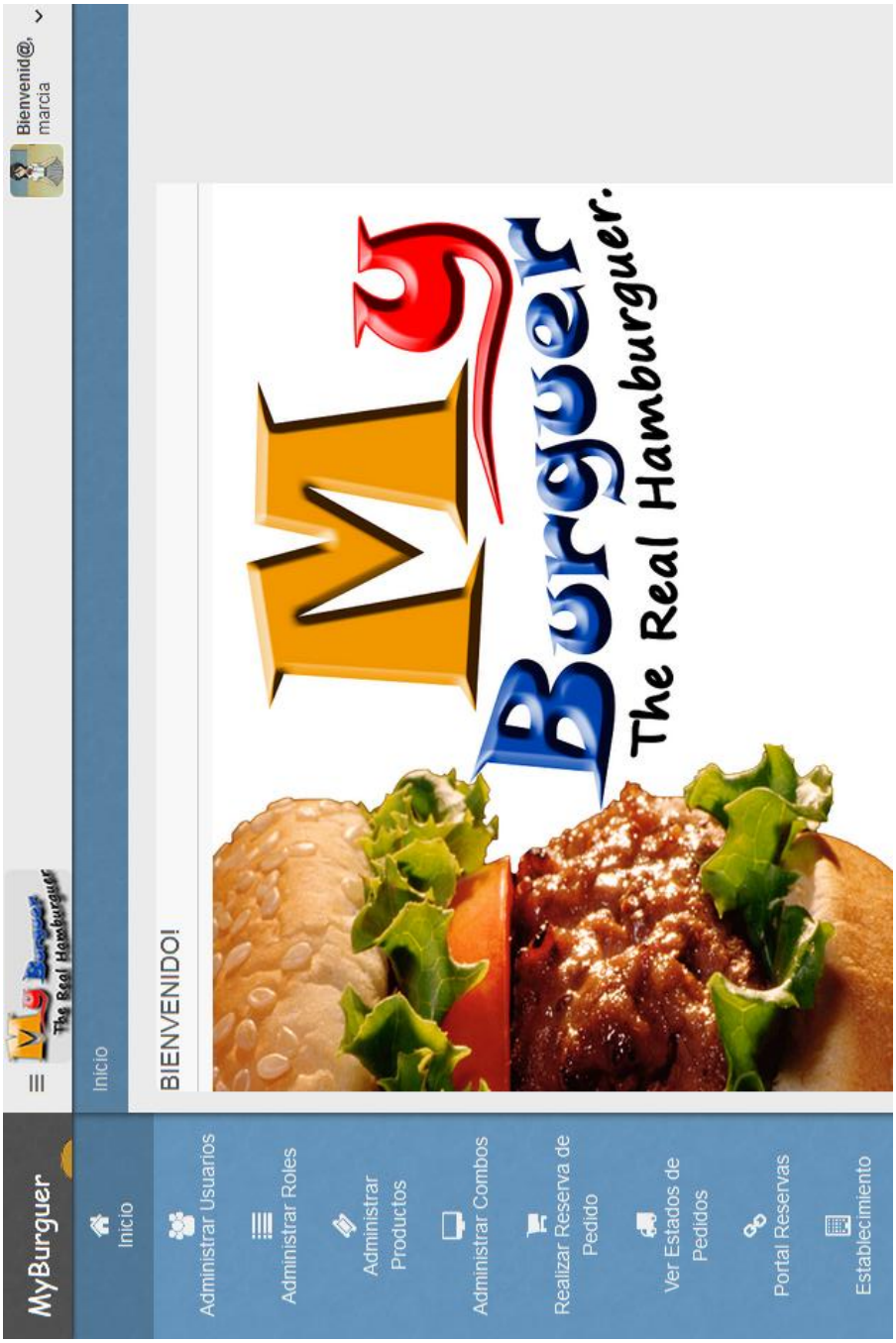


Figura 102. Pantalla Menú principal del sistema

Pantalla Administrar Usuarios

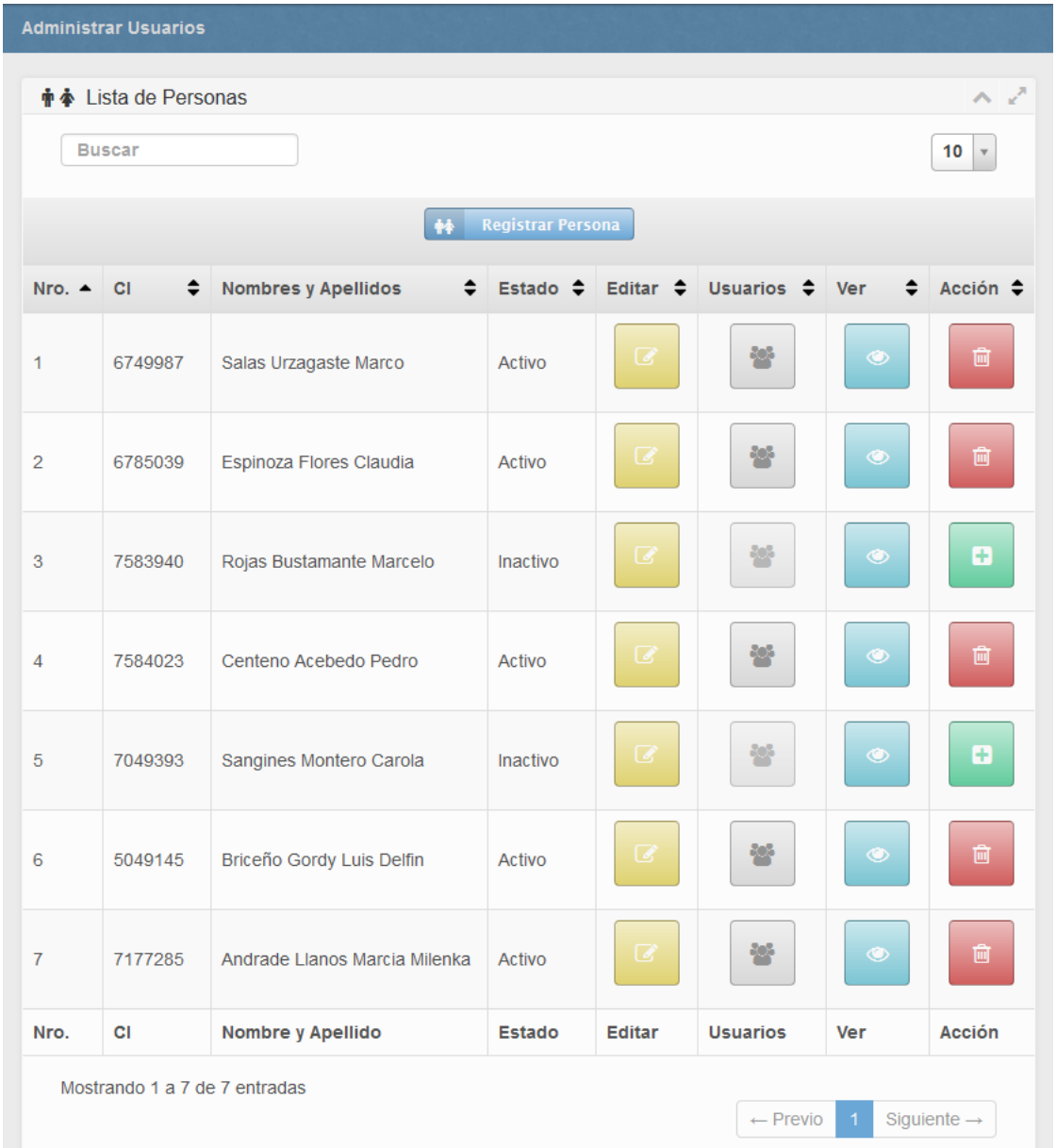


Figura 103. Pantalla Administrar Usuarios

Pantalla Registrar Persona

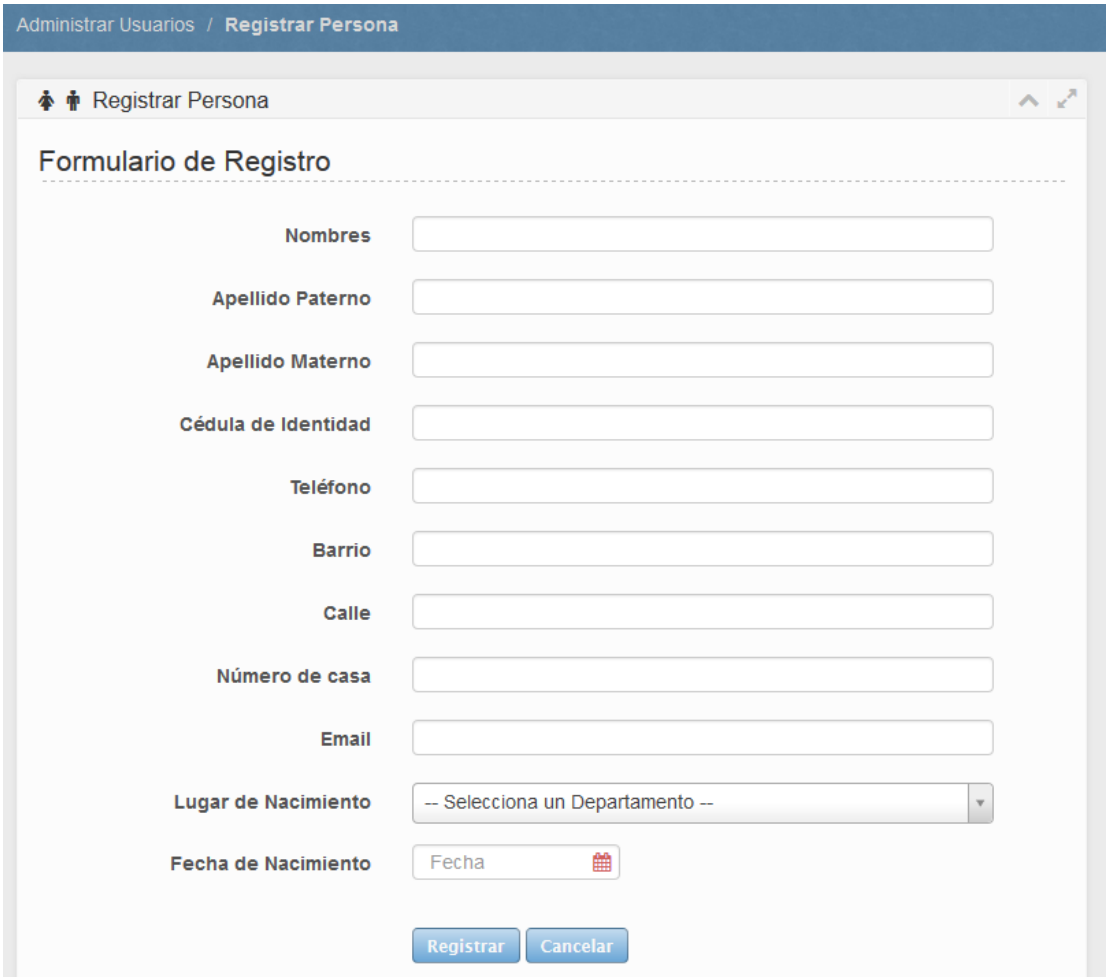


Figura 104. Pantalla Registrar Persona

Pantalla Editar Persona

Figura 105. Pantalla Editar Persona

Pantalla Usuarios de Persona

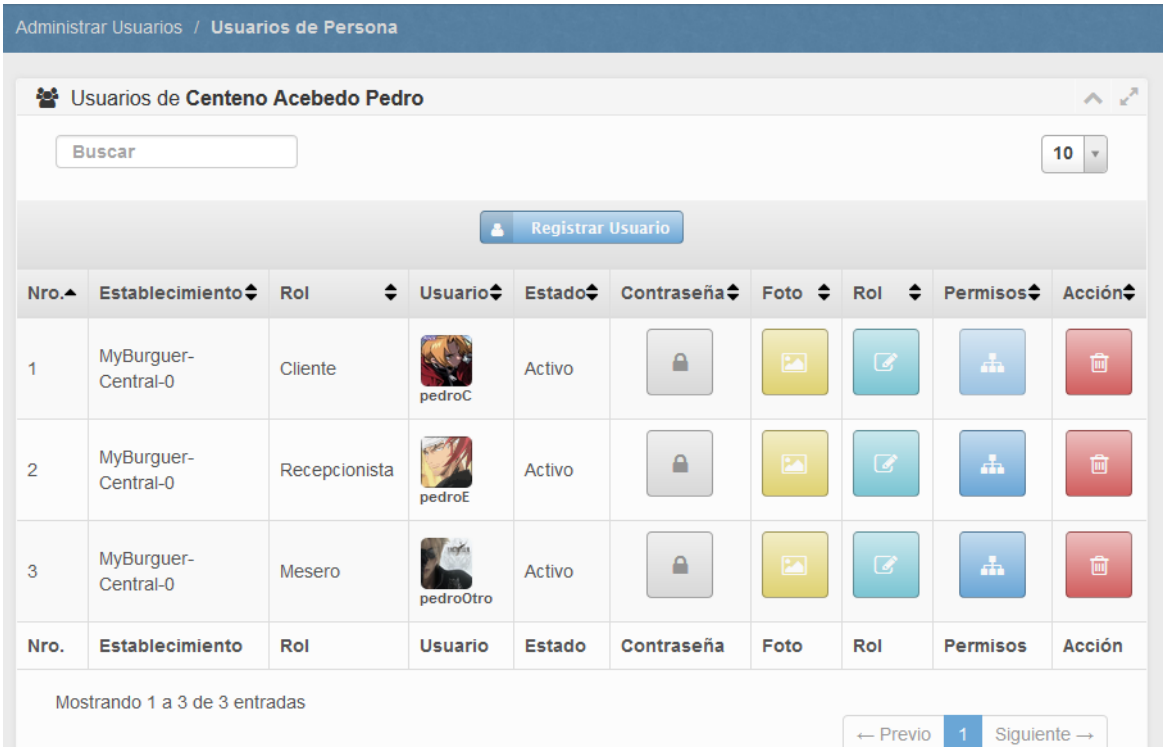


Figura 106. Pantalla Usuarios de Persona

Pantalla Registrar Usuario

Figura 107. Pantalla Registrar Usuario

Pantalla Cambiar contraseña

Figura 108. Pantalla Cambiar Contraseña

Pantalla Editar Fotografía

Figura 109. Pantalla Editar Fotografía

Pantalla Editar Rol

Figura 110. Pantalla Editar Rol

Pantalla Editar Permisos

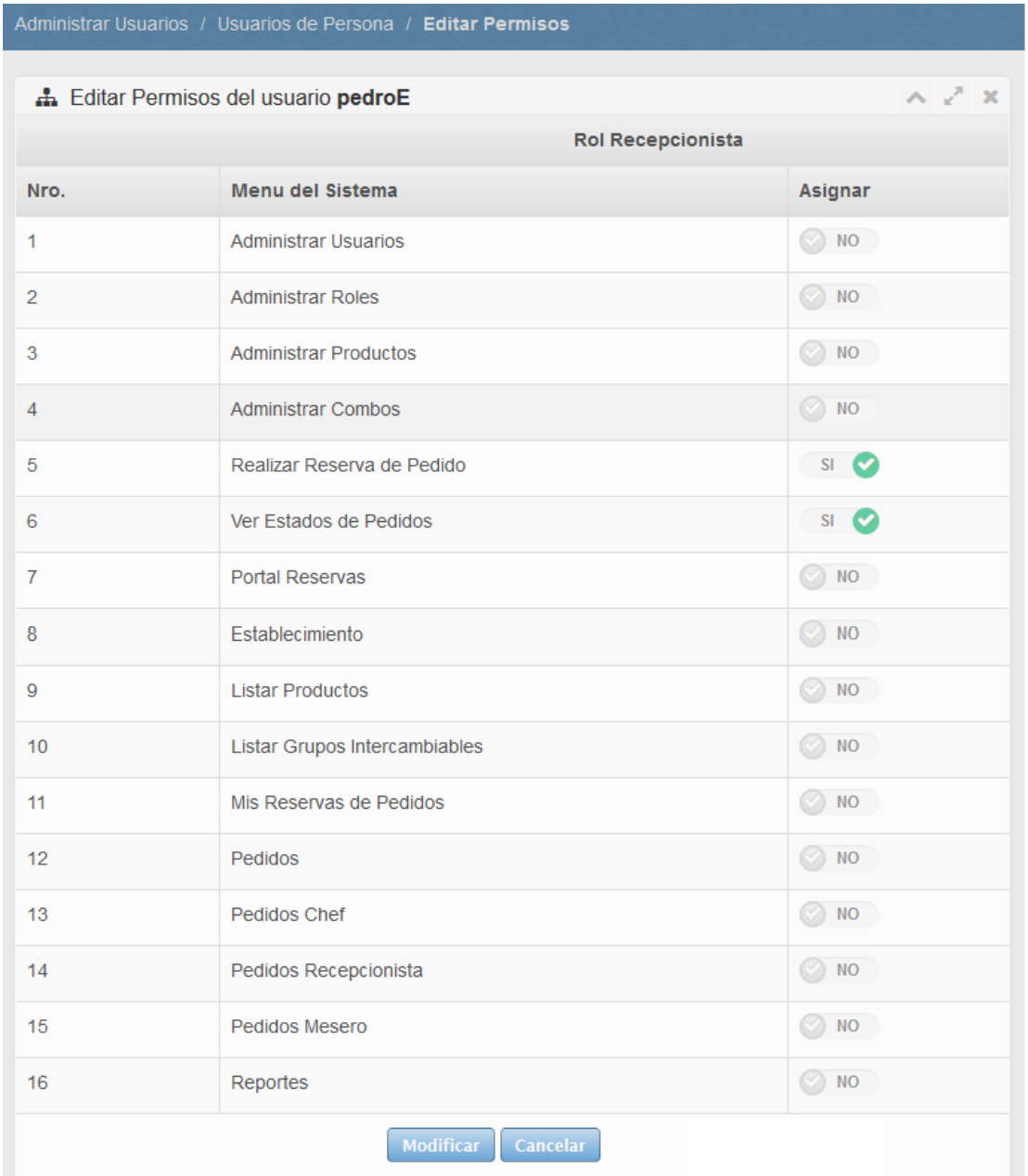


Figura 111. Pantalla Editar Permisos

Pantalla Administrar Roles

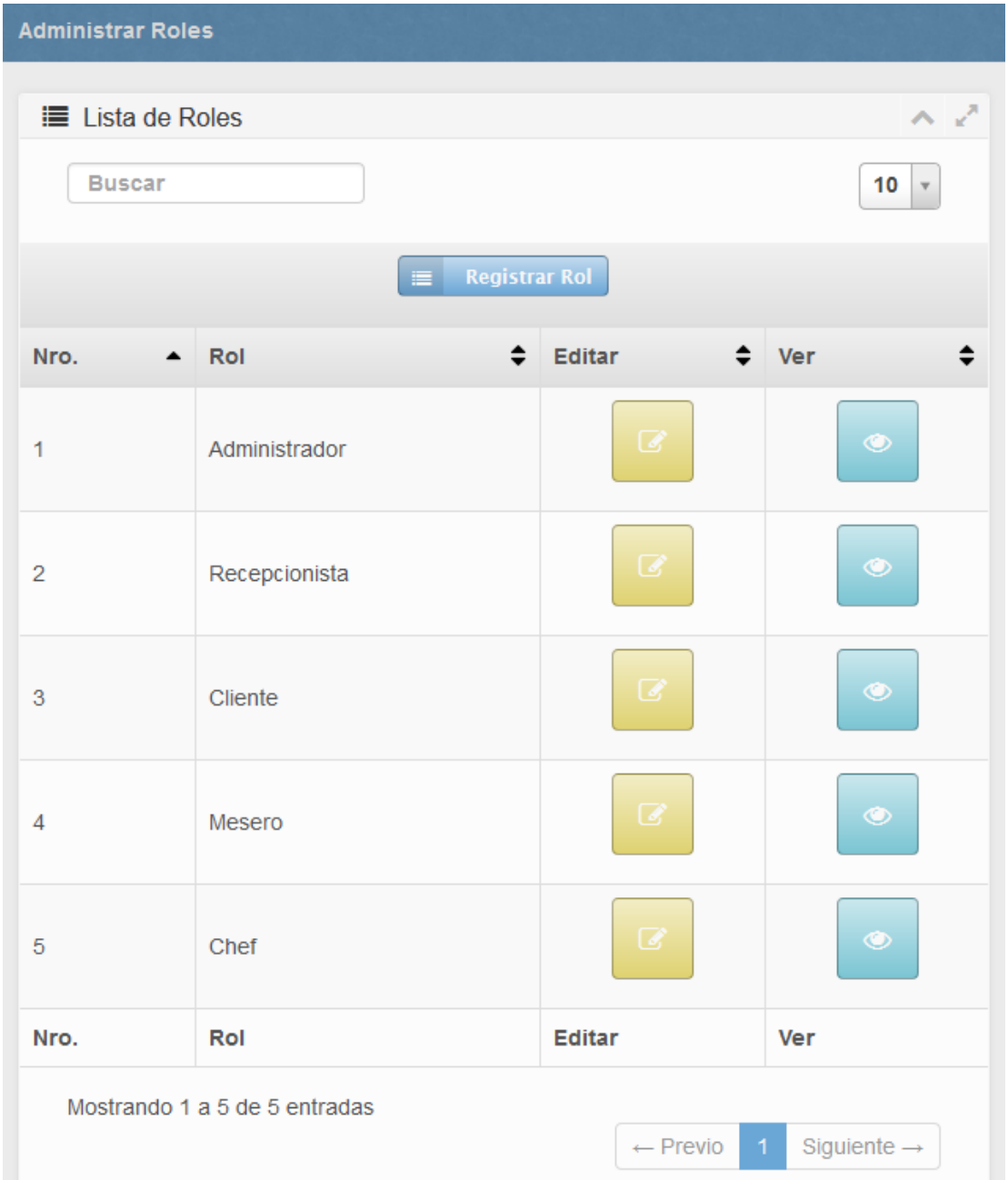


Figura 112. Pantalla Administrar Roles

Pantalla Registrar Rol

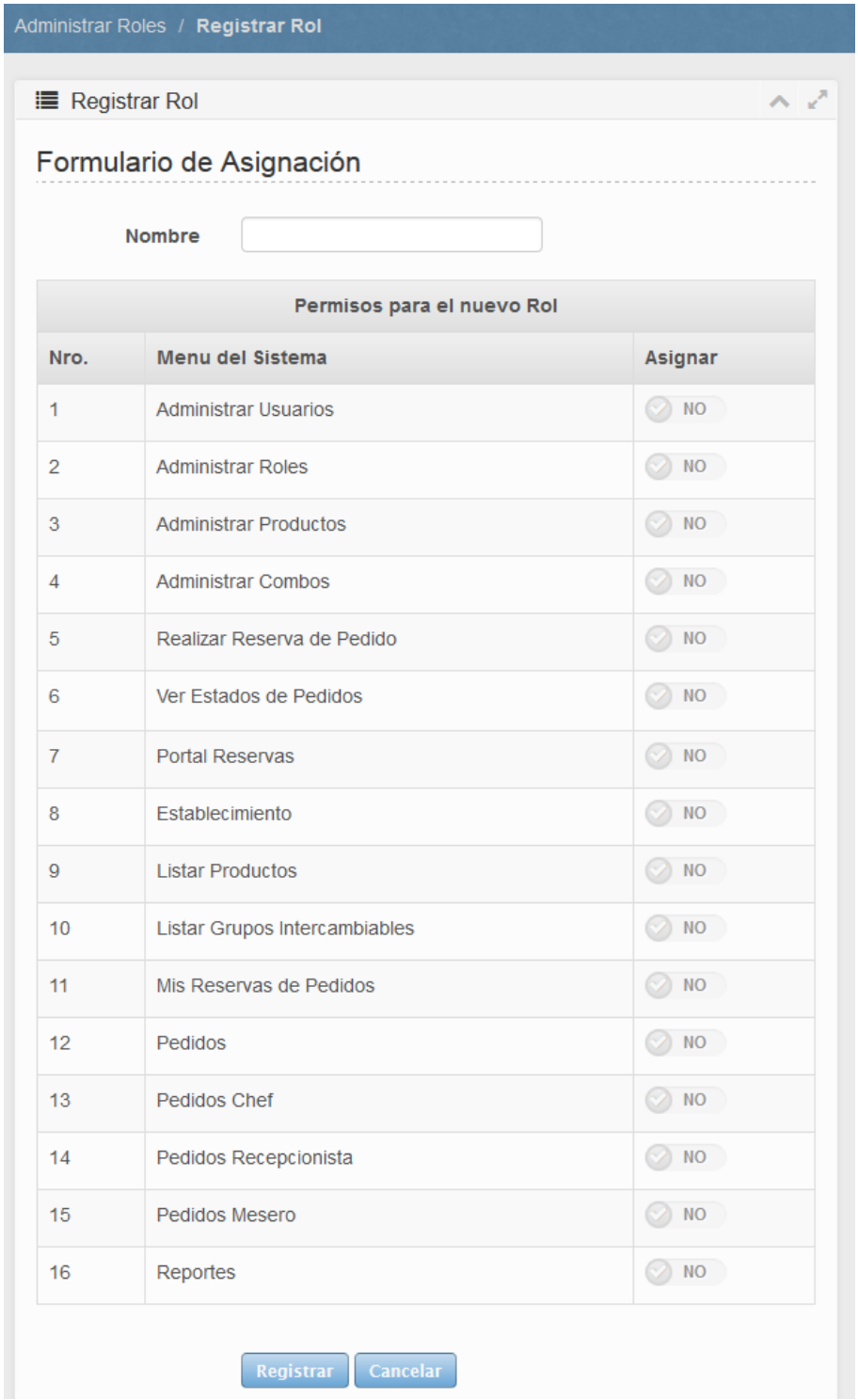


Figura 113. Pantalla Registrar Rol

Pantalla Editar Rol

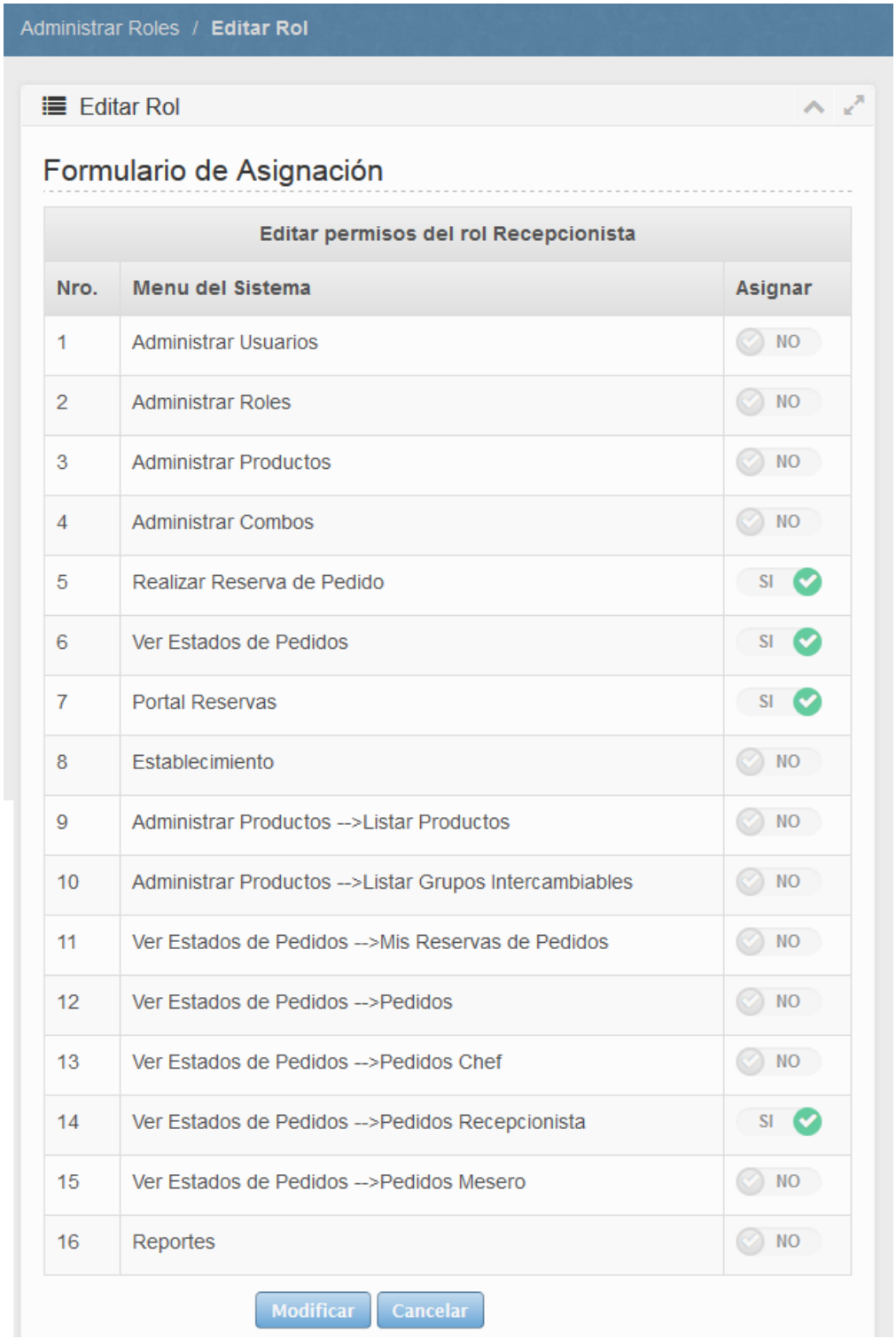


Figura 114. Pantalla Editar Rol

Pantalla Administrar Productos: Listar Productos

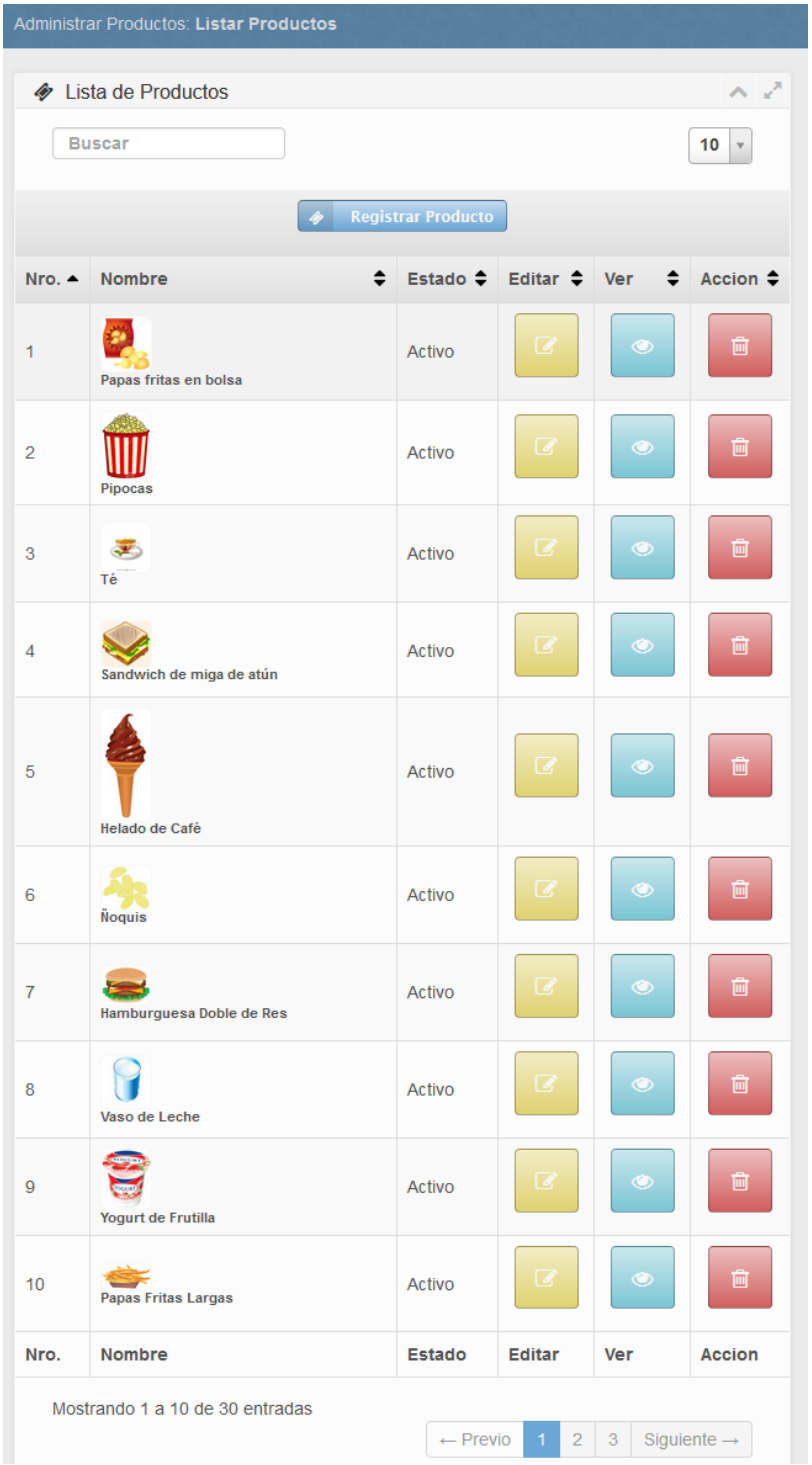


Figura 115. Pantalla Administrar Productos: Listar Productos

Pantalla Registrar Producto

Figura 116. Pantalla Registrar Producto

Pantalla Editar Producto

Figura 117. Pantalla Editar Producto

Listar Grupos Intercambiables

Figura 118. Pantalla Listar Grupos Intercambiables

Pantalla Registrar Grupo Intercambiables

Figura 119. Pantalla Listar Grupos Intercambiables

Pantalla Asignar Intercambio de Productos

Figura 120. Pantalla Asignar Intercambio de Productos

Pantalla Administrar Combos

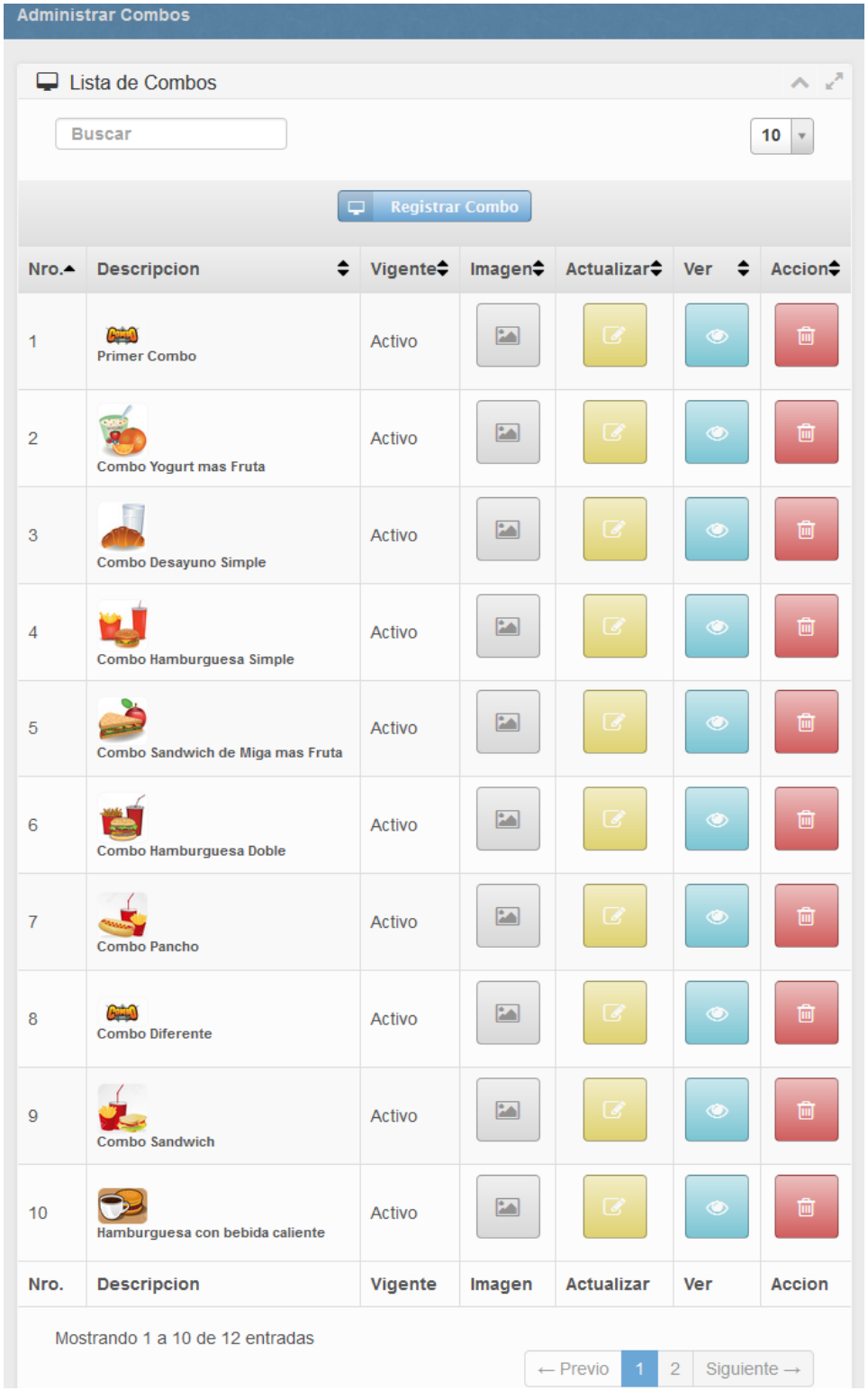


Figura 121. Pantalla Administrar Combos

Pantalla Registrar Combo

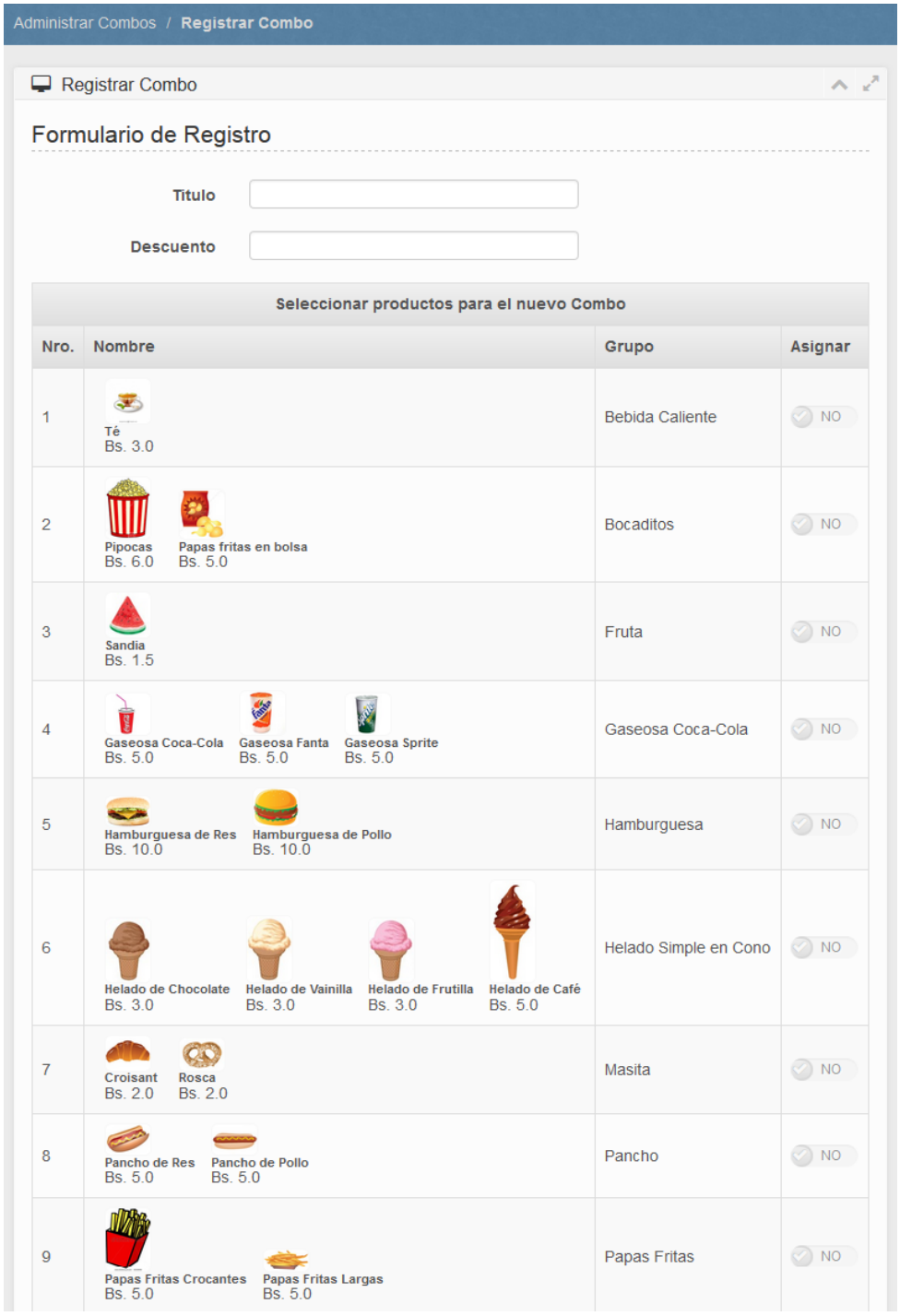


Figura 122. Pantalla Registrar Combo

Pantalla Actualizar Combo

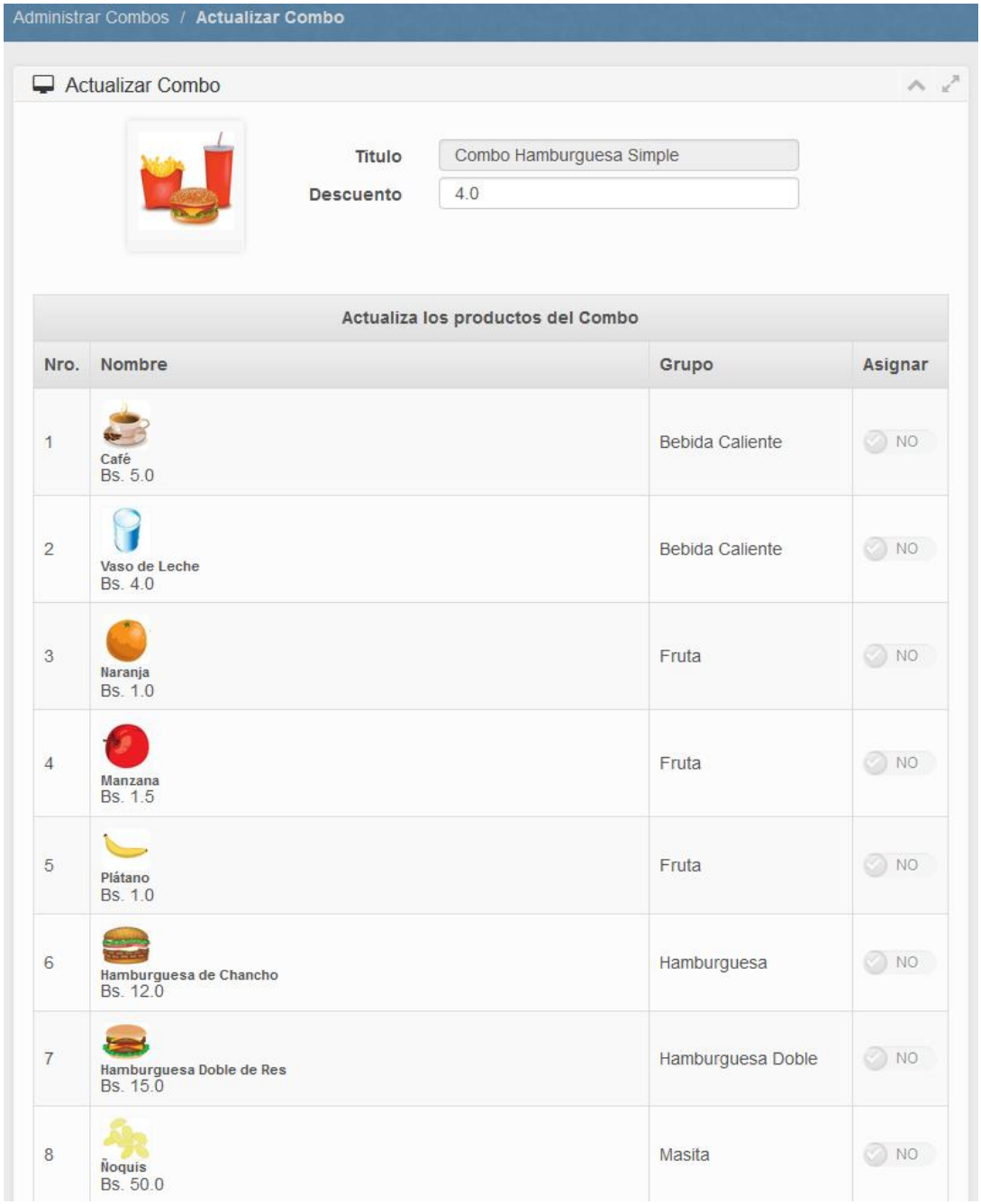


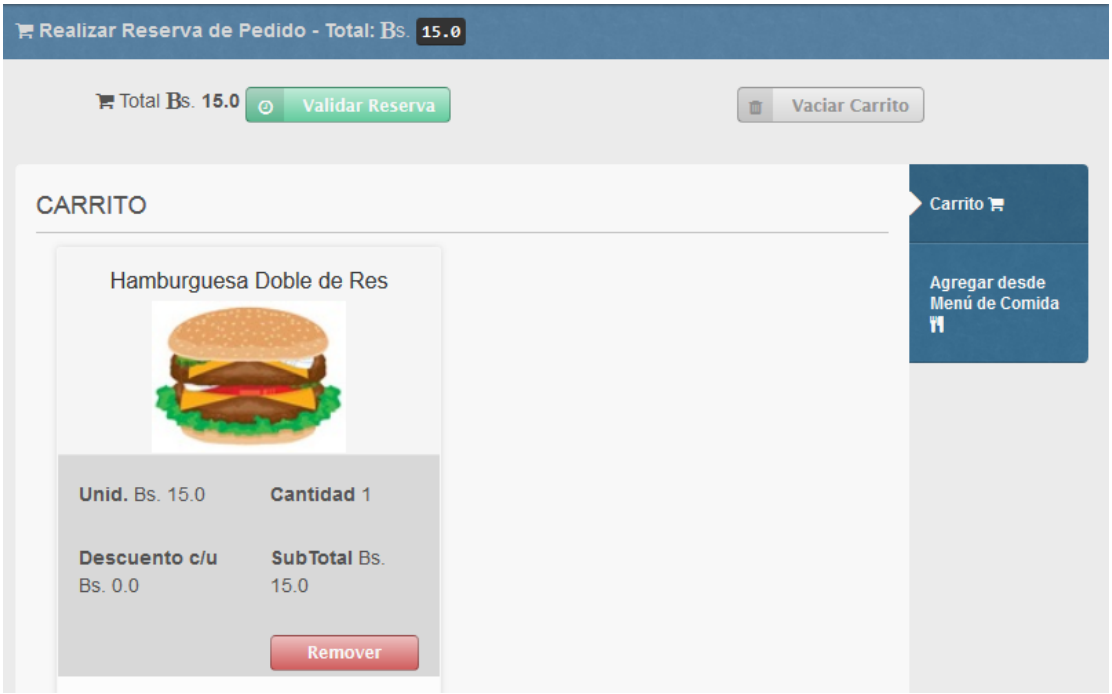
Figura 123. Pantalla Registrar Combo

Ver Combo

Figura 124. Pantalla Ver Combo

Pantalla Realizar Reserva de Pedido

(Pestaña Carrito)



(Pestaña Agregar desde Menú de Comida)

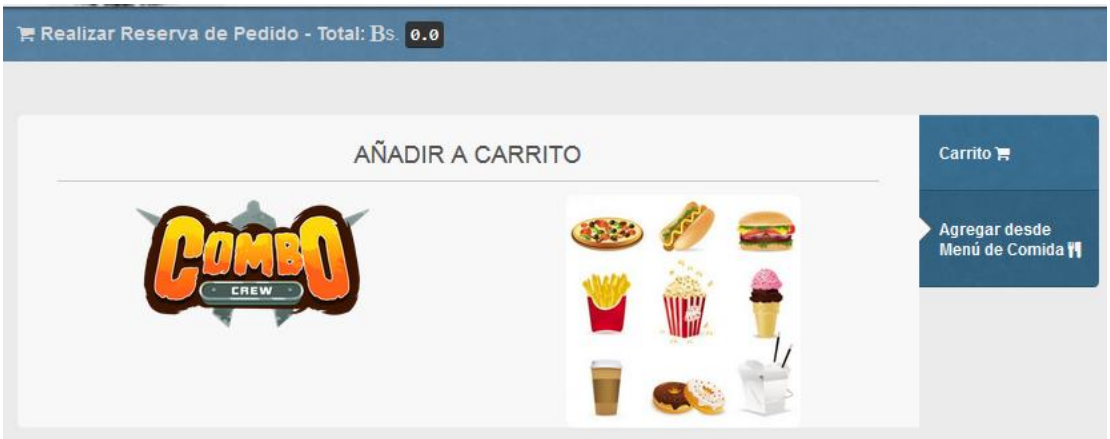


Figura 125. Pantalla Realizar Reserva de Pedido

Pantalla Seleccionar Combos

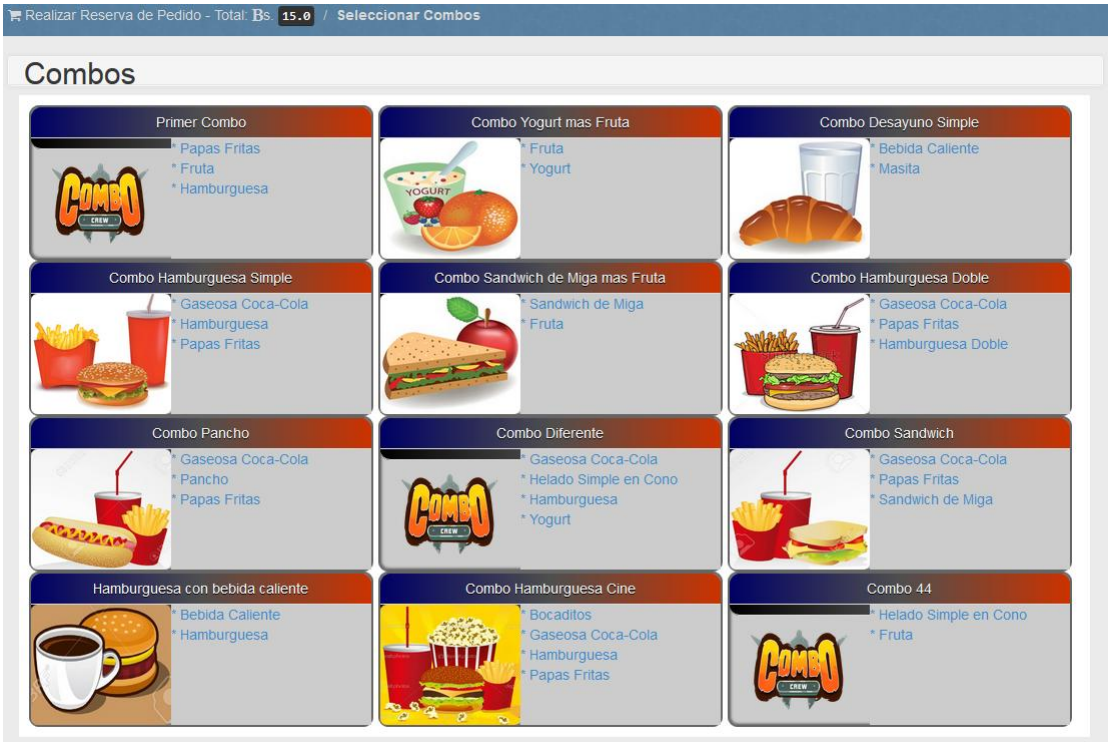


Figura 126. Pantalla Seleccionar Combos

Pantalla Seleccionar Productos de Combo

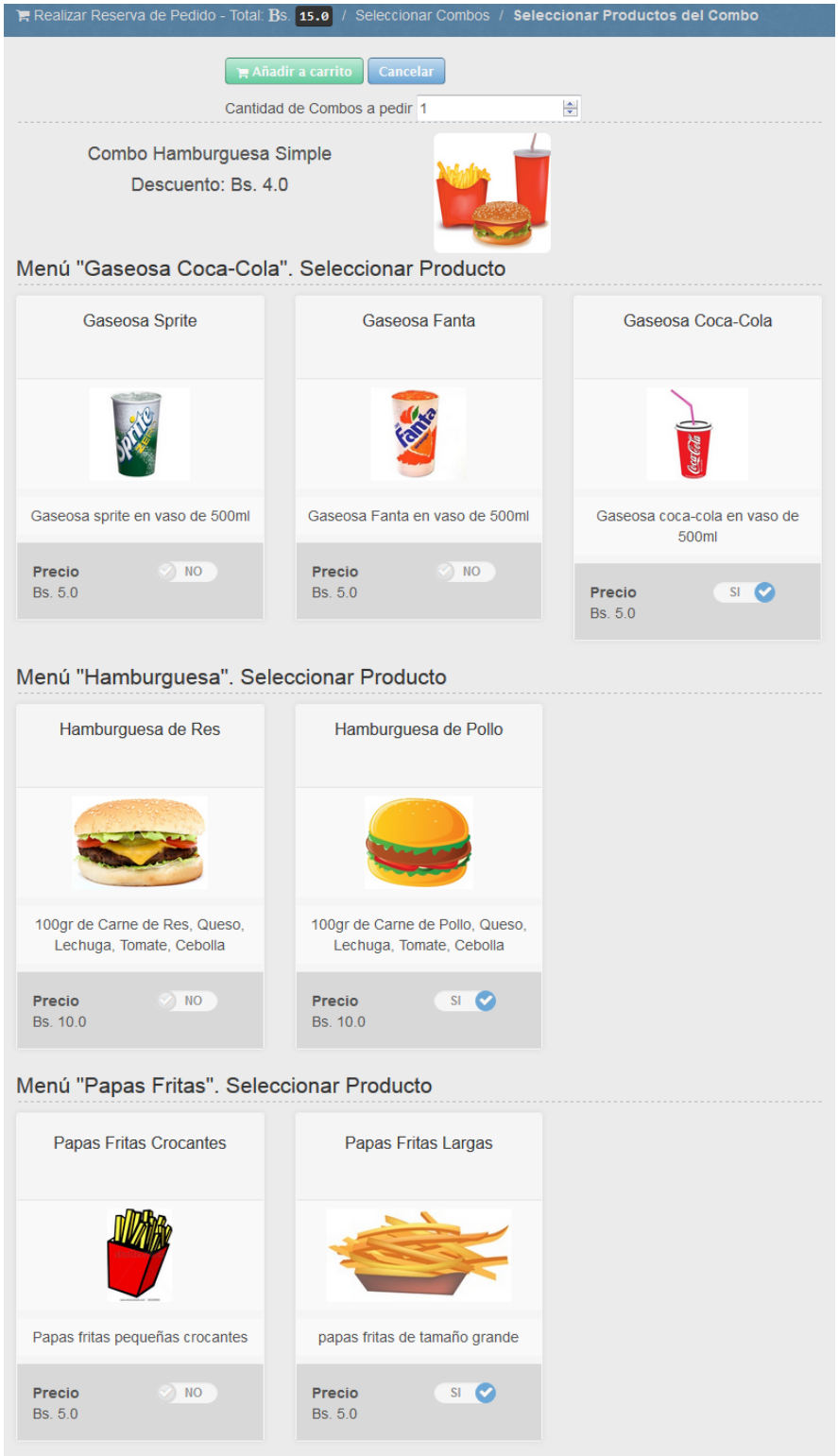


Figura 127. Pantalla Seleccionar Productos de Combo

Pantalla Seleccionar Productos

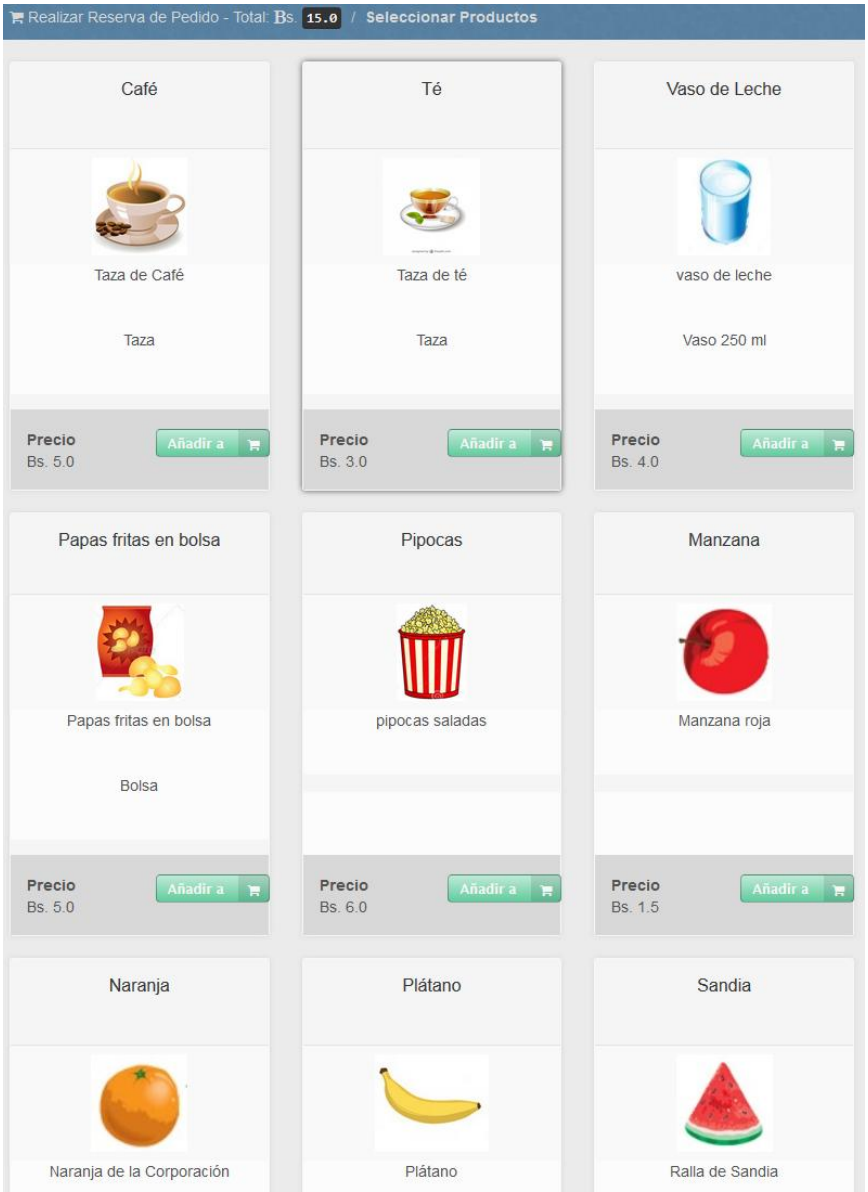


Figura 128. Pantalla Seleccionar Productos

Pantalla Validar Reserva

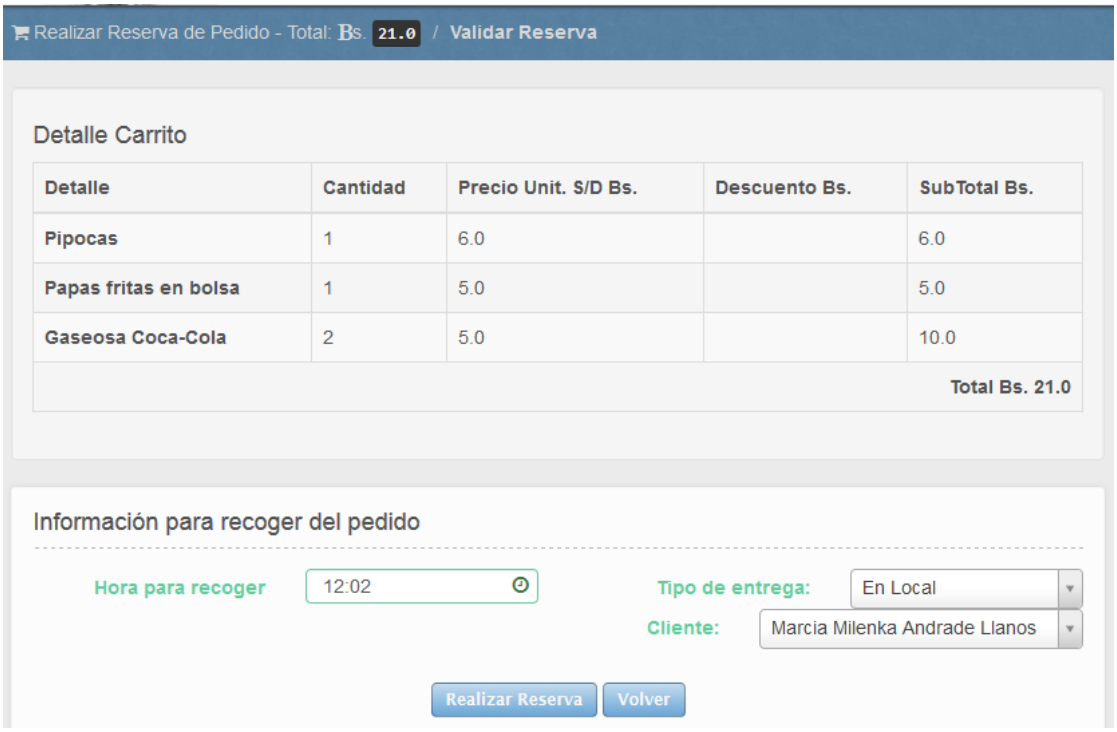


Figura 129. Pantalla Validar Reserva

Pantalla Papeleta Reserva

Figura 130. Pantalla Papeleta Reserva

Pantalla Ver Estados de Pedidos

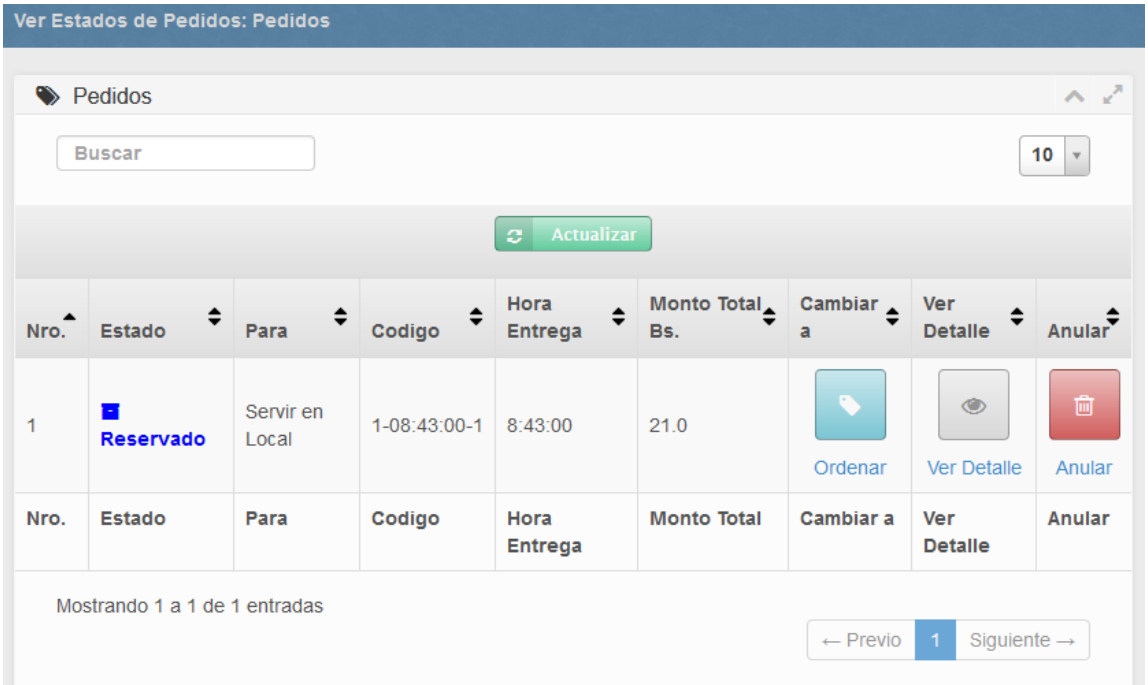


Figura 131. Pantalla Ver Estados de Pedidos

Ver Detalle de Pedido

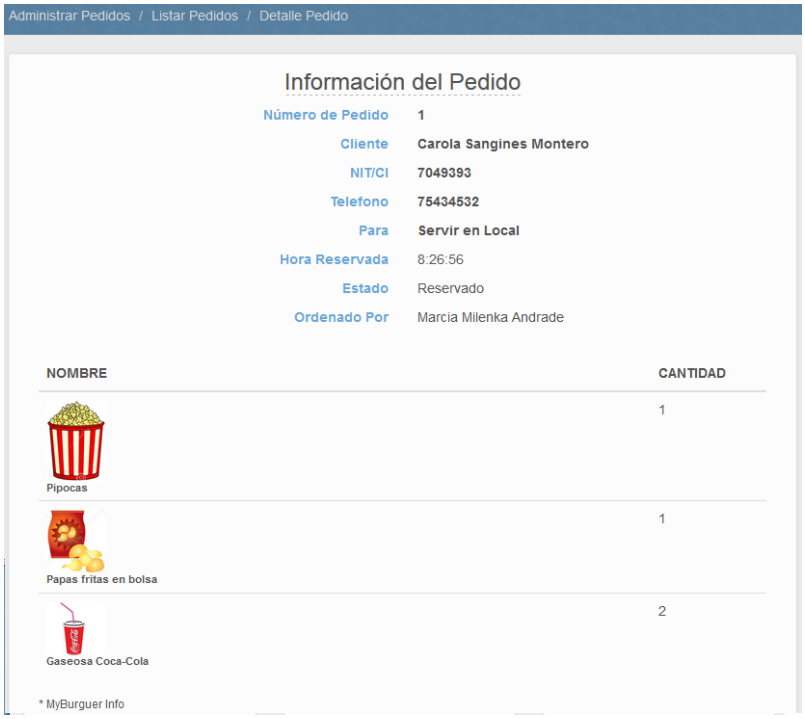


Figura 132. Pantalla Ver Detalle de Pedido

Pantalla Portal Reservas

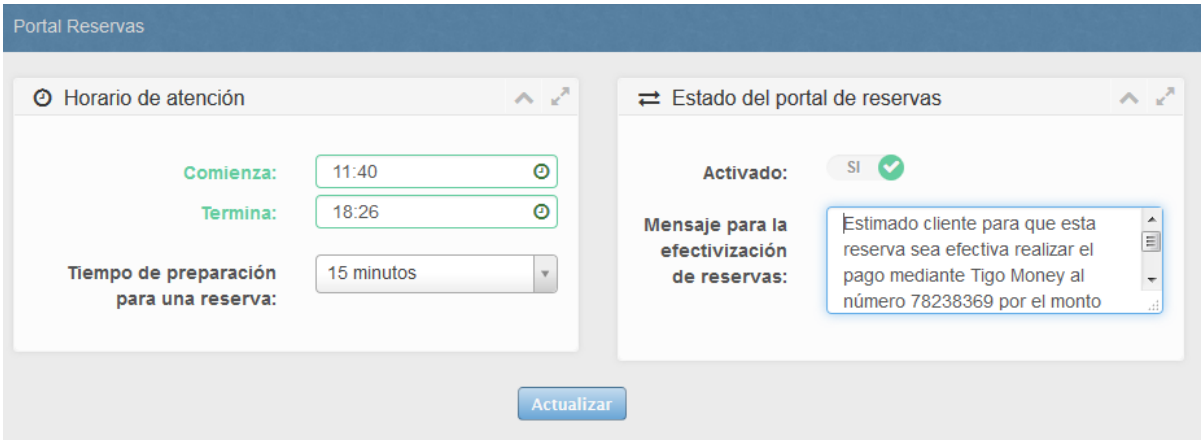


Figura 133. Pantalla Portal Reservas

Pantalla Establecimiento

Figura 134. Pantalla Establecimiento

Pantalla Editar Establecimiento

Figura 135. Pantalla Editar Establecimiento

Pantalla Ver Establecimiento

Figura 136. Pantalla Ver Establecimiento

Pantalla Reportes

Figura 137. Pantalla Reportes Clientes Registrados

Figura 138. Pantalla Reportes Reservas Efectivas por rango de fechas

Reportes / **Historial Reservas**

Historial de Reservas por cliente, por rango de fechas

Fecha actual: 02/03/2016

Fecha Inicio: 15/02/2016 **Fecha Fin:** 02/04/2016 Actualizar

Fecha	Nombres	Telefono	Reservas para llevar	Reservas para local	Monto Reserva
15/02/2016	Carola Sangines Montero	75434532	0	2	26.0
16/02/2016	Carola Sangines Montero	75434532	0	1	20.0
23/02/2016	Carola Sangines Montero	75434532	0	1	3.0
24/02/2016	Carola Sangines Montero	75434532	0	1	16.0
01/03/2016	Carola Sangines Montero	75434532	0	1	21.0
15/02/2016	Pedro Centeno Acebedo	75839202	1	0	20.0
16/02/2016	Pedro Centeno Acebedo	75839202	1	0	15.0
23/02/2016	Pedro Centeno Acebedo	75839202	1	1	99.0
15/02/2016	Marcia Milenka Andrade Llanos	79254690	0	4	87.5

Figura 139. Pantalla Reportes Historial de reservas por cliente, por rango de fechas

MyBurguer
Casa Matriz Barrio Juan XXIII, Calle Av. Font # 758
Telefono: 6634963 Tarija - Bolivia

NIT: 7177285489
N° FACTURA: 38
N° AUTORIZACION: 1904008691195
Actividad Económica: Restaurantes

FACTURA

Lugar y Fecha: Tarija, 20 de Marzo de 2016

Señor(es): **Marcia Milenka Andrade Llanos** **NIT/CI:** **7177285**

DETALLE	CANTIDAD	PRECIO UNIT.	DESCUENTO	SUBTOTAL
pipocas saladas	1	6.0	0.0	6.0
Combo Sandwich	2	15.0	3.0	24.0
* sandwich de miga con atún, lechuga, tomate, queso				
* papas fritas de tamaño grande				
* Gaseosa Fanta en vaso de 500ml				
SON: TREINTA 00/100 BOLIVIANOS.			**TOTAL Bs.:**	**30.0**

Código de Control: 78-71-AF-3C-02
Fecha Límite de Emisión: 16/09/2016

"ESTA FACTURA CONTRIBUYE AL DESARROLLO DEL PAIS"

Pantalla Factura

4.11 Modelo de Datos

4.11.1 Introducción.

Un diagrama de clases [36] se utiliza para modelar la vista de diseños estáticos de un sistema. Un diagrama de clases muestra un conjunto de interfaces, colaboraciones y sus relaciones. Presenta las clases del sistema con sus relaciones estructurales y de herencia. Gráficamente un diagrama es la colección de nodos y arcos".

Previendo que la persistencia de la información del sistema será soportada por una base de datos relacional este modelo describe la representación lógica de los datos persistentes, de acuerdo con el enfoque para el modelo relacional de datos.

Para expresar este modelo se utiliza el diagrama de clases (donde se utiliza un profile UML para el modelado de datos, para conseguir la representación de tablas, etc).

El diagrama de clases del sistema, es un artefacto creado para modelar conceptos de dominios como Clases de software. Normalmente tiene tres comportamientos, el tercero representa los métodos de la clase. UML incluye la notación de los diagramas de clases.

4.11.2 Propósito.

- ✓ Comprender la estructura y la dinámica del sistema deseado para la institución
- ✓ Comprender la interacción de los actores del sistema

4.11.3 Alcance.

- ✓ Si se elaboran bien los sistemas tienden a ser más fáciles de entender

4.11.4 Diagrama de Clases.

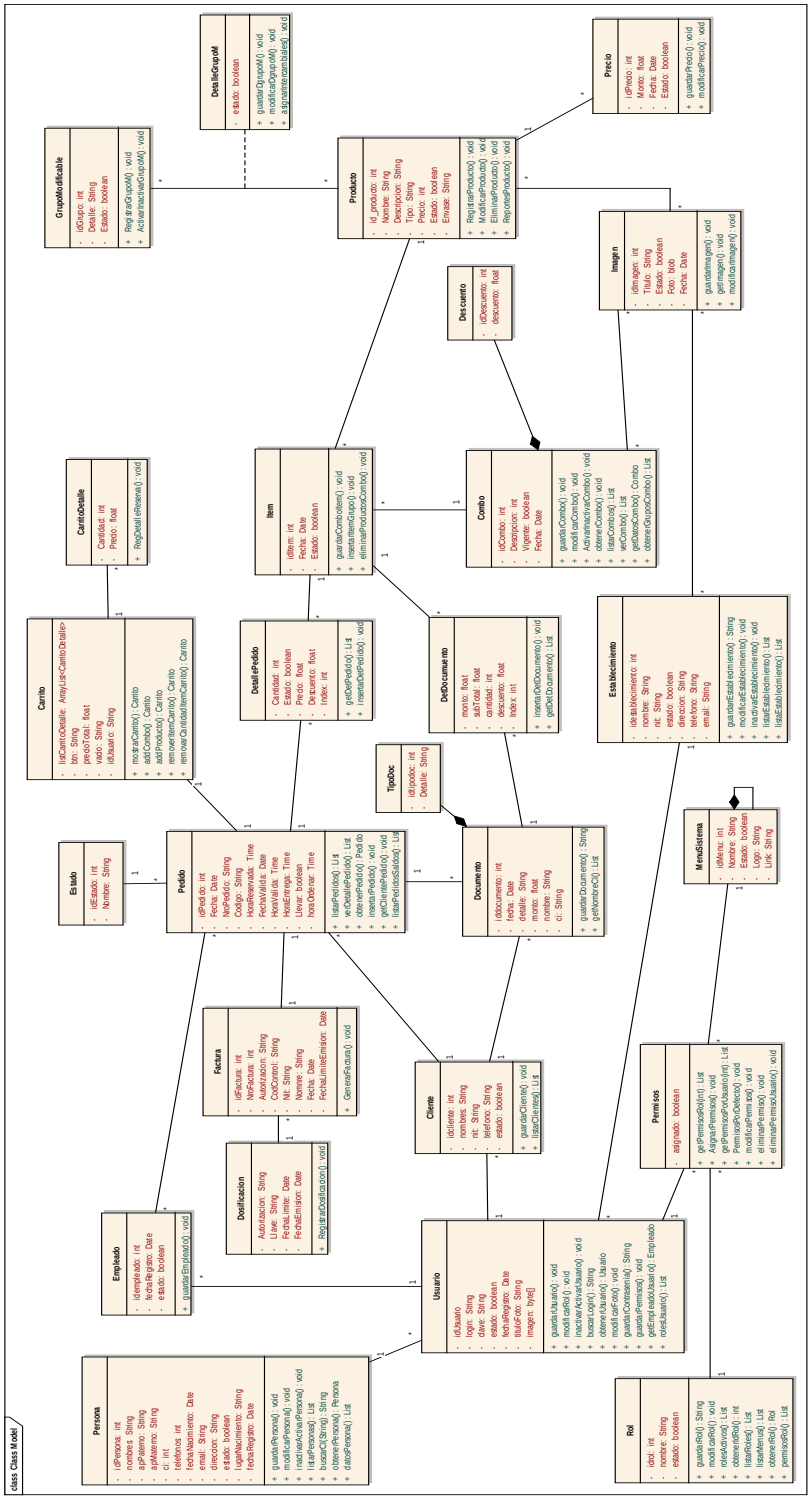


Figura 140. Diagrama de Clases

4.11.5 Diagrama Entidad Relación

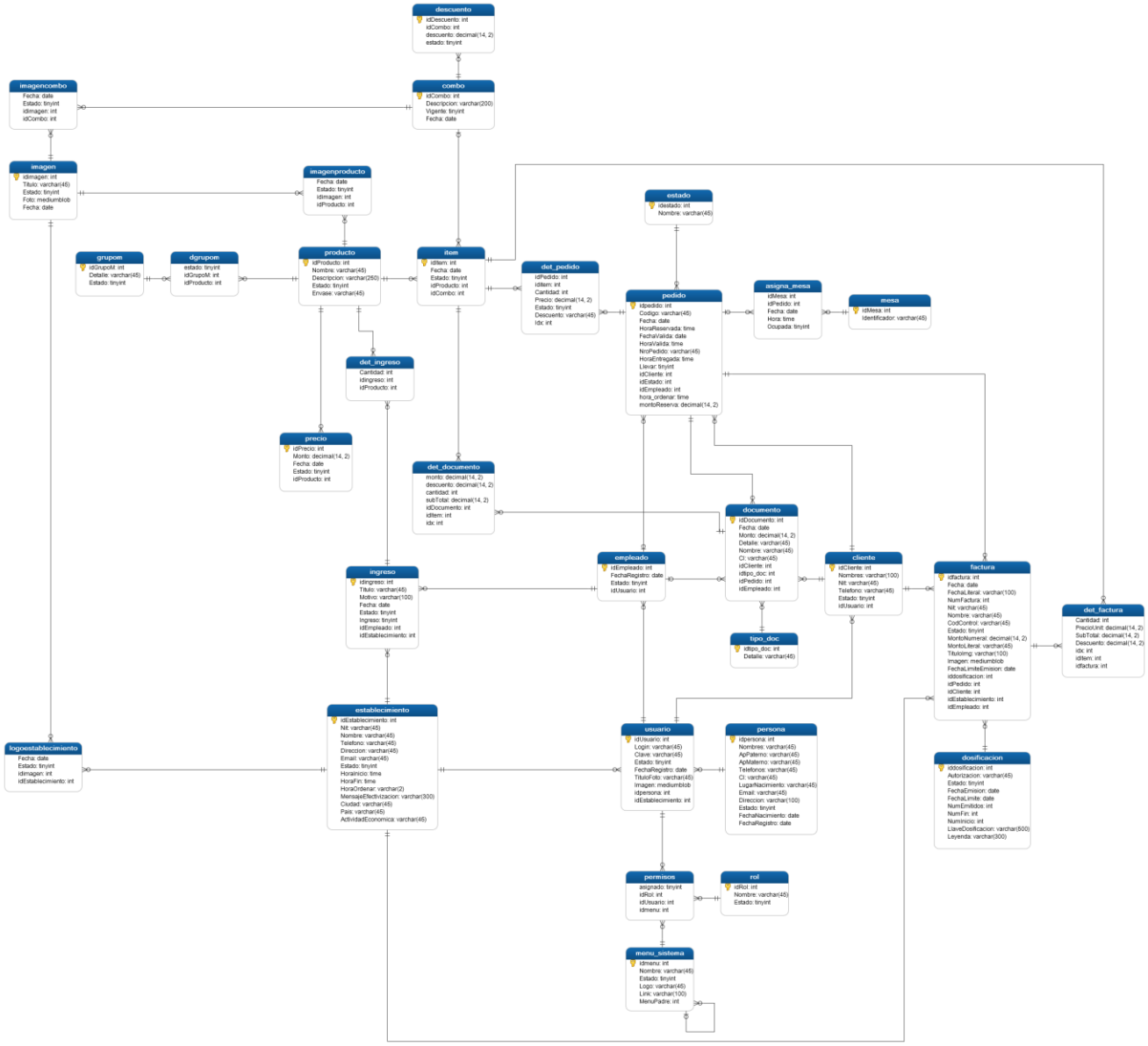


Figura 141. Diagrama Entidad Relación

4.11.6 Script Para la Creación de la Base de Datos.

```
SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='TRADITIONAL,ALLOW_INVALID_DATES';

CREATE SCHEMA IF NOT EXISTS `dbmyburguer` DEFAULT CHARACTER SET utf8 ;
USE `dbmyburguer` ;

-- -----------------------------------------------------
-- Table `dbmyburguer`.`establecimiento`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `dbmyburguer`.`establecimiento` (
  `idEstablecimiento` INT(11) NOT NULL AUTO_INCREMENT,
  `Nit` VARCHAR(45) NOT NULL,
  `Nombre` VARCHAR(45) NOT NULL,
  `Telefono` VARCHAR(45) NULL DEFAULT NULL,
  `Direccion` VARCHAR(45) NULL DEFAULT NULL,
  `Email` VARCHAR(45) NULL DEFAULT NULL,
  `Estado` TINYINT(1) NULL DEFAULT '1',
  `HoraInicio` TIME NULL DEFAULT NULL,
  `HoraFin` TIME NULL DEFAULT NULL,
  `HoraOrdenar` VARCHAR(2) NULL DEFAULT '0',
  `MensajeEfectivizacion` VARCHAR(300) NULL DEFAULT NULL,
  PRIMARY KEY (`idEstablecimiento`),
  UNIQUE INDEX `Nit_UNIQUE` (`Nit` ASC),
  UNIQUE INDEX `Nombre_UNIQUE` (`Nombre` ASC))
ENGINE = InnoDB
AUTO_INCREMENT = 3
DEFAULT CHARACTER SET = utf8;


-- -----------------------------------------------------
-- Table `dbmyburguer`.`persona`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `dbmyburguer`.`persona` (
  `idpersona` INT(11) NOT NULL AUTO_INCREMENT,
  `Nombres` VARCHAR(45) CHARACTER SET 'utf8' NOT NULL,
  `ApPaterno` VARCHAR(45) CHARACTER SET 'utf8' NULL DEFAULT NULL,
  `ApMaterno` VARCHAR(45) CHARACTER SET 'utf8' NULL DEFAULT NULL,
  `Telefonos` VARCHAR(45) CHARACTER SET 'utf8' NULL DEFAULT NULL,
  `CI` VARCHAR(45) CHARACTER SET 'utf8' NULL DEFAULT NULL,
  `LugarNacimiento` VARCHAR(45) CHARACTER SET 'utf8' NULL DEFAULT NULL,
  `Email` VARCHAR(45) CHARACTER SET 'utf8' NULL DEFAULT NULL,
  `Direccion` VARCHAR(100) CHARACTER SET 'utf8' NULL DEFAULT NULL,
  `Estado` TINYINT(1) NULL DEFAULT '1',
  `FechaNacimiento` DATE NULL DEFAULT NULL,
  `FechaRegistro` DATE NULL DEFAULT NULL,
  PRIMARY KEY (`idpersona`))
ENGINE = InnoDB
AUTO_INCREMENT = 9
DEFAULT CHARACTER SET = latin5;


-- -----------------------------------------------------
-- Table `dbmyburguer`.`usuario`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `dbmyburguer`.`usuario` (
  `idUsuario` INT(11) NOT NULL AUTO_INCREMENT,
  `Login` VARCHAR(45) NOT NULL,
```

```
  `Clave` VARCHAR(45) NOT NULL,
  `Estado` TINYINT(1) NULL DEFAULT '1',
  `FechaRegistro` DATE NULL DEFAULT NULL,
  `TituloFoto` VARCHAR(45) NULL DEFAULT NULL,
  `Imagen` MEDIUMBLOB NULL DEFAULT NULL,
  `idpersona` INT(11) NOT NULL,
  `idEstablecimiento` INT(11) NOT NULL,
  PRIMARY KEY (`idUsuario`),
  UNIQUE INDEX `Login_UNIQUE` (`Login` ASC),
  INDEX `fk_usuario_persona1_idx` (`idpersona` ASC),
  INDEX `fk_usuario_Establecimiento1_idx` (`idEstablecimiento` ASC),
  CONSTRAINT `fk_usuario_Establecimiento1`
    FOREIGN KEY (`idEstablecimiento`)
    REFERENCES `dbmyburguer`.`establecimiento` (`idEstablecimiento`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_usuario_persona1`
    FOREIGN KEY (`idpersona`)
    REFERENCES `dbmyburguer`.`persona` (`idpersona`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB
AUTO_INCREMENT = 12
DEFAULT CHARACTER SET = utf8;


-- -----------------------------------------------------
-- Table `dbmyburguer`.`cliente`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `dbmyburguer`.`cliente` (
  `idCliente` INT(11) NOT NULL AUTO_INCREMENT,
  `Nombres` VARCHAR(100) NULL DEFAULT NULL,
  `Nit` VARCHAR(45) NULL DEFAULT NULL,
  `Telefono` VARCHAR(45) NULL DEFAULT NULL,
  `Estado` TINYINT(1) NULL DEFAULT '1',
  `idUsuario` INT(11) NOT NULL,
  PRIMARY KEY (`idCliente`),
  INDEX `fk_Cliente_usuario1_idx` (`idUsuario` ASC),
  CONSTRAINT `fk_Cliente_usuario1`
    FOREIGN KEY (`idUsuario`)
    REFERENCES `dbmyburguer`.`usuario` (`idUsuario`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB
AUTO_INCREMENT = 5
DEFAULT CHARACTER SET = utf8;


-- -----------------------------------------------------
-- Table `dbmyburguer`.`combo`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `dbmyburguer`.`combo` (
  `idCombo` INT(11) NOT NULL AUTO_INCREMENT,
  `Descripcion` VARCHAR(200) NOT NULL,
  `Vigente` TINYINT(1) NULL DEFAULT NULL,
  `Fecha` DATE NULL DEFAULT NULL,
  PRIMARY KEY (`idCombo`),
  UNIQUE INDEX `Descripcion_UNIQUE` (`Descripcion` ASC))
ENGINE = InnoDB
AUTO_INCREMENT = 12
```

```
DEFAULT CHARACTER SET = utf8;


-- -----------------------------------------------------
-- Table `dbmyburguer`.`descuento`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `dbmyburguer`.`descuento` (
  `idDescuento` INT(11) NOT NULL AUTO_INCREMENT,
  `idCombo` INT(11) NOT NULL,
  `descuento` DECIMAL(14,2) NOT NULL,
  `estado` TINYINT(1) NOT NULL DEFAULT '1',
  PRIMARY KEY (`idDescuento`),
  INDEX `fk_Descuento_Combo1_idx` (`idCombo` ASC),
  CONSTRAINT `fk_Descuento_Combo1`
    FOREIGN KEY (`idCombo`)
    REFERENCES `dbmyburguer`.`combo` (`idCombo`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB
AUTO_INCREMENT = 24
DEFAULT CHARACTER SET = utf8;


-- -----------------------------------------------------
-- Table `dbmyburguer`.`empleado`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `dbmyburguer`.`empleado` (
  `idEmpleado` INT(11) NOT NULL AUTO_INCREMENT,
  `FechaRegistro` DATE NULL DEFAULT NULL,
  `Estado` TINYINT(1) NULL DEFAULT '1',
  `idUsuario` INT(11) NOT NULL,
  PRIMARY KEY (`idEmpleado`),
  INDEX `fk_Empleado_usuario1_idx` (`idUsuario` ASC),
  CONSTRAINT `fk_Empleado_usuario1`
    FOREIGN KEY (`idUsuario`)
    REFERENCES `dbmyburguer`.`usuario` (`idUsuario`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB
AUTO_INCREMENT = 13
DEFAULT CHARACTER SET = utf8;


-- -----------------------------------------------------
-- Table `dbmyburguer`.`estado`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `dbmyburguer`.`estado` (
  `idestado` INT(11) NOT NULL AUTO_INCREMENT,
  `Nombre` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`idestado`))
ENGINE = InnoDB
AUTO_INCREMENT = 7
DEFAULT CHARACTER SET = utf8;


-- -----------------------------------------------------
-- Table `dbmyburguer`.`pedido`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `dbmyburguer`.`pedido` (
  `idpedido` INT(11) NOT NULL AUTO_INCREMENT,
```

```
  `Codigo` VARCHAR(45) NOT NULL,
  `Fecha` DATE NOT NULL,
  `HoraReservada` TIME NULL DEFAULT NULL,
  `FechaValida` DATE NULL DEFAULT NULL,
  `HoraValida` TIME NULL DEFAULT NULL,
  `NroPedido` VARCHAR(45) NULL DEFAULT NULL,
  `HoraEntregada` TIME NULL DEFAULT NULL,
  `Llevar` TINYINT(1) NULL DEFAULT NULL,
  `idCliente` INT(11) NOT NULL,
  `idEstado` INT(11) NOT NULL,
  `idEmpleado` INT(11) NULL DEFAULT NULL,
  `hora_ordenar` TIME NULL DEFAULT NULL,
  `montoReserva` DECIMAL(14,2) NULL DEFAULT NULL,
  PRIMARY KEY (`idpedido`),
  INDEX `fk_Pedido_Cliente1_idx` (`idCliente` ASC),
  INDEX `fk_Pedido_Estado1_idx` (`idEstado` ASC),
  INDEX `fk_Pedido_empleado_idx` (`idEmpleado` ASC),
  CONSTRAINT `fk_Pedido_Cliente1`
    FOREIGN KEY (`idCliente`)
    REFERENCES `dbmyburguer`.`cliente` (`idCliente`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_Pedido_empleado1`
    FOREIGN KEY (`idEmpleado`)
    REFERENCES `dbmyburguer`.`empleado` (`idEmpleado`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_Pedido_estado1`
    FOREIGN KEY (`idEstado`)
    REFERENCES `dbmyburguer`.`estado` (`idestado`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB
AUTO_INCREMENT = 72
DEFAULT CHARACTER SET = utf8;


-- -----------------------------------------------------
-- Table `dbmyburguer`.`tipo_doc`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `dbmyburguer`.`tipo_doc` (
  `idtipo_doc` INT(11) NOT NULL AUTO_INCREMENT,
  `Detalle` VARCHAR(45) NULL DEFAULT NULL,
  PRIMARY KEY (`idtipo_doc`))
ENGINE = InnoDB
AUTO_INCREMENT = 2
DEFAULT CHARACTER SET = utf8;


-- -----------------------------------------------------
-- Table `dbmyburguer`.`documento`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `dbmyburguer`.`documento` (
  `idDocumento` INT(11) NOT NULL AUTO_INCREMENT,
  `Fecha` DATE NOT NULL,
  `Monto` DECIMAL(14,2) NOT NULL,
  `Detalle` VARCHAR(45) NOT NULL,
  `Nombre` VARCHAR(45) NOT NULL,
  `CI` VARCHAR(45) NOT NULL,
  `idCliente` INT(11) NOT NULL,
```

```
  `idtipo_doc` INT(11) NOT NULL,
  `idPedido` INT(11) NOT NULL,
  PRIMARY KEY (`idDocumento`),
  INDEX `fk_Documento_Cliente1_idx` (`idCliente` ASC),
  INDEX `fk_Documento_tipo_doc1_idx` (`idtipo_doc` ASC),
  INDEX `fk_Documento_pedido1_idx` (`idPedido` ASC),
  CONSTRAINT `fk_Documento_Cliente1`
    FOREIGN KEY (`idCliente`)
    REFERENCES `dbmyburguer`.`cliente` (`idCliente`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_Documento_pedido1`
    FOREIGN KEY (`idPedido`)
    REFERENCES `dbmyburguer`.`pedido` (`idpedido`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_Documento_tipo_doc1`
    FOREIGN KEY (`idtipo_doc`)
    REFERENCES `dbmyburguer`.`tipo_doc` (`idtipo_doc`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB
AUTO_INCREMENT = 43
DEFAULT CHARACTER SET = utf8;


-- -----------------------------------------------------
-- Table `dbmyburguer`.`producto`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `dbmyburguer`.`producto` (
  `idProducto` INT(11) NOT NULL AUTO_INCREMENT,
  `Nombre` VARCHAR(45) NOT NULL,
  `Descripcion` VARCHAR(250) NULL DEFAULT NULL,
  `Estado` TINYINT(1) NULL DEFAULT '1',
  `Envase` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`idProducto`),
  UNIQUE INDEX `Nombre_UNIQUE` (`Nombre` ASC))
ENGINE = InnoDB
AUTO_INCREMENT = 32
DEFAULT CHARACTER SET = utf8;


-- -----------------------------------------------------
-- Table `dbmyburguer`.`item`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `dbmyburguer`.`item` (
  `idItem` INT(11) NOT NULL AUTO_INCREMENT,
  `Fecha` DATE NULL DEFAULT NULL,
  `Estado` TINYINT(1) NULL DEFAULT NULL,
  `idProducto` INT(11) NOT NULL,
  `idCombo` INT(11) NULL DEFAULT NULL,
  PRIMARY KEY (`idItem`),
  INDEX `fk_item_Producto1_idx` (`idProducto` ASC),
  INDEX `fk_item_Combo1_idx` (`idCombo` ASC),
  CONSTRAINT `fk_item_Combo1`
    FOREIGN KEY (`idCombo`)
    REFERENCES `dbmyburguer`.`combo` (`idCombo`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_item_Producto1`
```

```
    FOREIGN KEY (`idProducto`)
    REFERENCES `dbmyburguer`.`producto` (`idProducto`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB
AUTO_INCREMENT = 195
DEFAULT CHARACTER SET = utf8;


-- -----------------------------------------------------
-- Table `dbmyburguer`.`det_documento`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `dbmyburguer`.`det_documento` (
  `monto` DECIMAL(14,2) NOT NULL,
  `descuento` DECIMAL(14,2) NULL DEFAULT NULL,
  `cantidad` INT(10) NOT NULL,
  `subTotal` DECIMAL(14,2) NOT NULL,
  `idDocumento` INT(11) NOT NULL,
  `idItem` INT(11) NOT NULL,
  `idx` INT(11) NULL DEFAULT NULL,
  INDEX `fk_det_Documento_documento1_idx` (`idDocumento` ASC),
  INDEX `fk_det_Documento_item1_idx` (`idItem` ASC),
  CONSTRAINT `fk_det_Documento_documento1`
    FOREIGN KEY (`idDocumento`)
    REFERENCES `dbmyburguer`.`documento` (`idDocumento`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_det_Documento_item1`
    FOREIGN KEY (`idItem`)
    REFERENCES `dbmyburguer`.`item` (`idItem`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8;


-- -----------------------------------------------------
-- Table `dbmyburguer`.`det_pedido`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `dbmyburguer`.`det_pedido` (
  `idPedido` INT(11) NOT NULL,
  `iditem` INT(11) NOT NULL,
  `Cantidad` INT(10) NOT NULL,
  `Precio` DECIMAL(14,2) NOT NULL,
  `Estado` TINYINT(1) NOT NULL DEFAULT '1',
  `Descuento` VARCHAR(45) NOT NULL,
  `Idx` INT(11) NOT NULL,
  INDEX `fk_det_pedido_pedido1_idx` (`idPedido` ASC),
  INDEX `fk_det_pedido_item1_idx` (`iditem` ASC),
  CONSTRAINT `fk_det_pedido_item1`
    FOREIGN KEY (`iditem`)
    REFERENCES `dbmyburguer`.`item` (`idItem`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_det_pedido_Pedido1`
    FOREIGN KEY (`idPedido`)
    REFERENCES `dbmyburguer`.`pedido` (`idpedido`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB
```

```
DEFAULT CHARACTER SET = utf8;


-- -----------------------------------------------------
-- Table `dbmyburguer`.`grupom`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `dbmyburguer`.`grupom` (
  `idGrupoM` INT(11) NOT NULL AUTO_INCREMENT,
  `Detalle` VARCHAR(45) NOT NULL,
  `Estado` TINYINT(1) NULL DEFAULT '1',
  PRIMARY KEY (`idGrupoM`),
  UNIQUE INDEX `Detalle_UNIQUE` (`Detalle` ASC))
ENGINE = InnoDB
AUTO_INCREMENT = 14
DEFAULT CHARACTER SET = utf8;


-- -----------------------------------------------------
-- Table `dbmyburguer`.`dgrupom`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `dbmyburguer`.`dgrupom` (
  `estado` TINYINT(1) NULL DEFAULT NULL,
  `idGrupoM` INT(11) NOT NULL,
  `idProducto` INT(11) NOT NULL,
  INDEX `fk_DGrupoM_GrupoM1_idx` (`idGrupoM` ASC),
  INDEX `fk_DGrupoM_Producto1_idx` (`idProducto` ASC),
  CONSTRAINT `fk_DGrupoM_GrupoM1`
    FOREIGN KEY (`idGrupoM`)
    REFERENCES `dbmyburguer`.`grupom` (`idGrupoM`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_DGrupoM_Producto1`
    FOREIGN KEY (`idProducto`)
    REFERENCES `dbmyburguer`.`producto` (`idProducto`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8;


-- -----------------------------------------------------
-- Table `dbmyburguer`.`imagen`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `dbmyburguer`.`imagen` (
  `idimagen` INT(11) NOT NULL AUTO_INCREMENT,
  `Titulo` VARCHAR(45) NULL DEFAULT NULL,
  `Estado` TINYINT(1) NULL DEFAULT NULL,
  `Foto` MEDIUMBLOB NULL DEFAULT NULL,
  `Fecha` DATE NULL DEFAULT NULL,
  PRIMARY KEY (`idimagen`))
ENGINE = InnoDB
AUTO_INCREMENT = 61
DEFAULT CHARACTER SET = utf8;


-- -----------------------------------------------------
-- Table `dbmyburguer`.`imagencombo`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `dbmyburguer`.`imagencombo` (
  `Fecha` DATE NULL DEFAULT NULL,
```

```
  `Estado` TINYINT(1) NULL DEFAULT NULL,
  `idimagen` INT(11) NOT NULL,
  `idCombo` INT(11) NOT NULL,
  INDEX `fk_imagenItem_imagen1_idx` (`idimagen` ASC),
  INDEX `fk_imagenItem_Combo1_idx` (`idCombo` ASC),
  CONSTRAINT `fk_imagenItem_Combo1`
    FOREIGN KEY (`idCombo`)
    REFERENCES `dbmyburguer`.`combo` (`idCombo`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_imagenItem_imagen1`
    FOREIGN KEY (`idimagen`)
    REFERENCES `dbmyburguer`.`imagen` (`idimagen`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8;


-- -----------------------------------------------------
-- Table `dbmyburguer`.`imagenproducto`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `dbmyburguer`.`imagenproducto` (
  `Fecha` DATE NULL DEFAULT NULL,
  `Estado` TINYINT(1) NULL DEFAULT NULL,
  `idimagen` INT(11) NOT NULL,
  `idProducto` INT(11) NOT NULL,
  INDEX `fk_imagenProducto_imagen1_idx` (`idimagen` ASC),
  INDEX `fk_imagenProducto_Producto1_idx` (`idProducto` ASC),
  CONSTRAINT `fk_imagenProducto_imagen1`
    FOREIGN KEY (`idimagen`)
    REFERENCES `dbmyburguer`.`imagen` (`idimagen`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_imagenProducto_Producto1`
    FOREIGN KEY (`idProducto`)
    REFERENCES `dbmyburguer`.`producto` (`idProducto`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8;


-- -----------------------------------------------------
-- Table `dbmyburguer`.`logoestablecimiento`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `dbmyburguer`.`logoestablecimiento` (
  `Fecha` DATE NULL DEFAULT NULL,
  `Estado` TINYINT(1) NULL DEFAULT NULL,
  `idimagen` INT(11) NOT NULL,
  `idEstablecimiento` INT(11) NOT NULL,
  INDEX `fk_logoEstablecimiento_imagen1_idx` (`idimagen` ASC),
  INDEX `fk_logoEstablecimiento_Establecimiento1_idx` (`idEstablecimiento`
ASC),
  CONSTRAINT `fk_logoEstablecimiento_Establecimiento1`
    FOREIGN KEY (`idEstablecimiento`)
    REFERENCES `dbmyburguer`.`establecimiento` (`idEstablecimiento`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_logoEstablecimiento_imagen1`
```

```
    FOREIGN KEY (`idimagen`)
    REFERENCES `dbmyburguer`.`imagen` (`idimagen`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8;


-- -----------------------------------------------------
-- Table `dbmyburguer`.`menu_sistema`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `dbmyburguer`.`menu_sistema` (
  `idmenu` INT(11) NOT NULL AUTO_INCREMENT,
  `Nombre` VARCHAR(45) NOT NULL,
  `Estado` TINYINT(1) NULL DEFAULT '1',
  `Logo` VARCHAR(45) NULL DEFAULT NULL,
  `Link` VARCHAR(100) NULL DEFAULT NULL,
  `MenuPadre` INT(11) NULL DEFAULT NULL,
  PRIMARY KEY (`idmenu`),
  INDEX `fk_menu_sistema_menu_sistema1_idx` (`MenuPadre` ASC),
  CONSTRAINT `fk_menu_sistema_menu_sistema1`
    FOREIGN KEY (`MenuPadre`)
    REFERENCES `dbmyburguer`.`menu_sistema` (`idmenu`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB
AUTO_INCREMENT = 19
DEFAULT CHARACTER SET = utf8;


-- -----------------------------------------------------
-- Table `dbmyburguer`.`rol`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `dbmyburguer`.`rol` (
  `idRol` INT(11) NOT NULL AUTO_INCREMENT,
  `Nombre` VARCHAR(45) NOT NULL,
  `Estado` TINYINT(1) NULL DEFAULT '1',
  PRIMARY KEY (`idRol`),
  UNIQUE INDEX `Nombre_UNIQUE` (`Nombre` ASC))
ENGINE = InnoDB
AUTO_INCREMENT = 6
DEFAULT CHARACTER SET = utf8;


-- -----------------------------------------------------
-- Table `dbmyburguer`.`permisos`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `dbmyburguer`.`permisos` (
  `asignado` TINYINT(1) NULL DEFAULT NULL,
  `idRol` INT(11) NOT NULL,
  `idUsuario` INT(11) NOT NULL,
  `idmenu` INT(11) NOT NULL,
  INDEX `fk_permisos_rol1_idx` (`idRol` ASC),
  INDEX `fk_permisos_usuario1_idx` (`idUsuario` ASC),
  INDEX `fk_permisos_menu_sistema1_idx` (`idmenu` ASC),
  CONSTRAINT `fk_permisos_menu_sistema1`
    FOREIGN KEY (`idmenu`)
    REFERENCES `dbmyburguer`.`menu_sistema` (`idmenu`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
```

```
  CONSTRAINT `fk_permisos_rol1`
    FOREIGN KEY (`idRol`)
    REFERENCES `dbmyburguer`.`rol` (`idRol`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_permisos_usuario1`
    FOREIGN KEY (`idUsuario`)
    REFERENCES `dbmyburguer`.`usuario` (`idUsuario`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8;


-- -----------------------------------------------------
-- Table `dbmyburguer`.`precio`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `dbmyburguer`.`precio` (
  `idPrecio` INT(11) NOT NULL AUTO_INCREMENT,
  `Monto` DECIMAL(14,2) NULL DEFAULT NULL,
  `Fecha` DATE NULL DEFAULT NULL,
  `Estado` TINYINT(1) NULL DEFAULT '1',
  `idProducto` INT(11) NOT NULL,
  PRIMARY KEY (`idPrecio`),
  INDEX `fk_Precio_Producto1_idx` (`idProducto` ASC),
  CONSTRAINT `fk_Precio_Producto1`
    FOREIGN KEY (`idProducto`)
    REFERENCES `dbmyburguer`.`producto` (`idProducto`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB
AUTO_INCREMENT = 38
DEFAULT CHARACTER SET = utf8;

USE `dbmyburguer` ;

-- -----------------------------------------------------
-- procedure DesActivarEstablecimiento
-- -----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `DesActivarEstablecimiento`(in
est boolean, in ide int(11))
BEGIN
      update Establecimiento set Estado=est where idestablecimiento = ide;
END$$

DELIMITER ;

-- -----------------------------------------------------
-- procedure actualizarDescuento
-- -----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `actualizarDescuento`(in idc
int(11), in des decimal(14,2))
BEGIN
      update descuento set estado=false
```

```
	where idcombo=idc;

	insert into descuento(idCombo, descuento)
	values (idc, des);
END$$

DELIMITER ;

-- -----------------------------------------------------
-- procedure actualizarPortalReservas
-- -----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `actualizarPortalReservas`(in
est boolean, in horaI time, in horaF time, in horaO varchar(2), in mensaje
varchar(300))
BEGIN
	UPDATE menu_sistema
	SET
	Estado = est
	WHERE idMenu = 5;

	UPDATE establecimiento
	SET HoraInicio = horaI, HoraFin = horaF, HoraOrdenar = horaO,
MensajeEfectivizacion = mensaje
	WHERE idEstablecimiento = 2;

END$$

DELIMITER ;

-- -----------------------------------------------------
-- procedure asignarIntercambiables
-- -----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `asignarIntercambiables`(in idg
int(11), in idp int(11))
BEGIN
	update dgrupom set estado=true where idgrupom=idg and idproducto=idp;
END$$

DELIMITER ;

-- -----------------------------------------------------
-- procedure asignarIntercambiablesFalse
-- -----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `asignarIntercambiablesFalse`(in
idg int(11))
BEGIN
	update dgrupom set estado=false where idgrupom=idg;
END$$

DELIMITER ;
```

```
-- -----------------------------------------------------
-- procedure asignarPermisos
-- -----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `asignarPermisos`(in idm
int(11), in idr int(11), in idu int(11), in asg boolean)
BEGIN
	insert into permisos (idmenu, idRol, idUsuario, asignado) values(idm,
idr, idu, asg);
END$$

DELIMITER ;

-- -----------------------------------------------------
-- procedure borratemporal
-- -----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `borratemporal`()
BEGIN
	drop temporary table tablatemporal;
END$$

DELIMITER ;

-- -----------------------------------------------------
-- procedure buscarCombo
-- -----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `buscarCombo`(in des
varchar(200))
BEGIN
	select Descripcion from combo where Descripcion=des;
END$$

DELIMITER ;

-- -----------------------------------------------------
-- procedure buscarGrupoM
-- -----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `buscarGrupoM`(in det
varchar(45))
BEGIN
	select detalle from grupom where detalle=det;
END$$

DELIMITER ;

-- -----------------------------------------------------
-- procedure buscarNitEstablecimiento
-- -----------------------------------------------------
```

```
DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `buscarNitEstablecimiento`(in ni
varchar(45))
BEGIN
	SELECT Nit FROM Establecimiento WHERE Nit = ni;
END$$

DELIMITER ;

-- -----------------------------------------------------
-- procedure buscarNombreEstablecimiento
-- -----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `buscarNombreEstablecimiento`(in
nom varchar(45))
BEGIN
	SELECT Nombre FROM Establecimiento WHERE Nombre = nom;
END$$

DELIMITER ;

-- -----------------------------------------------------
-- procedure buscarPersonaCI
-- -----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `buscarPersonaCI`(in c
varchar(15))
BEGIN
	SELECT ci FROM persona WHERE ci = c;
END$$

DELIMITER ;

-- -----------------------------------------------------
-- procedure buscarProducto
-- -----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `buscarProducto`(in nom
varchar(45))
BEGIN
	select Nombre from producto where Nombre=nom;
END$$

DELIMITER ;

-- -----------------------------------------------------
-- procedure buscarRol
-- -----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `buscarRol`(in nom varchar(45))
BEGIN
```

```
	select * from rol where Nombre = nom;
END$$

DELIMITER ;

-- -----------------------------------------------------
-- procedure buscarUsuarioLogin
-- -----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `buscarUsuarioLogin`(in log
varchar(45))
BEGIN
	SELECT Login FROM usuario WHERE Login = log;
END$$

DELIMITER ;

-- -----------------------------------------------------
-- procedure cambiarContraseña
-- -----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `cambiarContraseña`(in cl
varchar(45), in idu int(11))
BEGIN
	update usuario set Clave=cl WHERE idUsuario=idu;
END$$

DELIMITER ;

-- -----------------------------------------------------
-- procedure cambiarEstadoPedido
-- -----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `cambiarEstadoPedido`(in idp
int(11), in ide int(11))
BEGIN
	UPDATE pedido
	SET
	idEstado = ide
	WHERE idpedido = idp;
END$$

DELIMITER ;

-- -----------------------------------------------------
-- procedure cantidadProductosRangoFechaAnio
-- -----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE
`cantidadProductosRangoFechaAnio`(in FechaInicio varchar(45), in FechaFinal
varchar(45), in nombreEstado varchar(45))
BEGIN
```

```
select year(p.fecha)as 'Año',pr.idproducto,pr.Nombre,sum(dp.Cantidad) as
'Cantidad'
from pedido p
left join det_pedido dp on dp.idPedido=p.idpedido and estado=1
left join item i on i.idItem=dp.iditem
left join producto pr on pr.idProducto=i.idProducto
where p.idestado=(select idestado from estado where Nombre=nombreEstado)
and (convert(p.Fecha,date) between convert(FechaInicio,date) AND
convert(FechaFinal,date))
and not isnull(pr.idProducto)
group by year(p.fecha),pr.idProducto;
END$$

DELIMITER ;

-- -----------------------------------------------------
-- procedure cantidadProductosRangoFechaMes
-- -----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE
`cantidadProductosRangoFechaMes`(in FechaInicio varchar(45), in FechaFinal
varchar(45), in nombreEstado varchar(45))
BEGIN
select year(p.fecha)as 'Año',monthname(p.fecha) as 'mes'
,pr.idproducto,pr.Nombre,sum(dp.Cantidad) as 'Cantidad'
from pedido p
left join det_pedido dp on dp.idPedido=p.idpedido and estado=1
left join item i on i.idItem=dp.iditem
left join producto pr on pr.idProducto=i.idProducto
where p.idestado=(select idestado from estado where Nombre=nombreEstado)
and (convert(p.Fecha,date) between convert(FechaInicio,date) AND
convert(FechaFinal,date))
and not isnull(pr.idProducto)
group by monthname(p.fecha),pr.idProducto;

END$$

DELIMITER ;

-- -----------------------------------------------------
-- procedure cantidadProductosRangoFechaSemana
-- -----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE
`cantidadProductosRangoFechaSemana`(in FechaInicio varchar(45), in
FechaFinal varchar(45), in nombreEstado varchar(45))
BEGIN
select year(p.fecha)as 'Año',monthname(p.fecha) as 'mes' ,week(p.fecha) as
'Semana',pr.idproducto,pr.Nombre,sum(dp.Cantidad) as 'Cantidad'
from pedido p
left join det_pedido dp on dp.idPedido=p.idpedido and estado=1
left join item i on i.idItem=dp.iditem
left join producto pr on pr.idProducto=i.idProducto
where p.idestado=(select idestado from estado where Nombre=nombreEstado)
and (convert(p.Fecha,date) between convert(FechaInicio,date) AND
convert(FechaFinal,date))
```

```
and not isnull(pr.idProducto)
group by week(p.fecha),pr.idProducto;
END$$

DELIMITER ;

-- -----------------------------------------------------
-- procedure createmporal
-- -----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `createmporal`()
BEGIN
      create temporary table tablatemporal
       (`idproducto` INT NOT NULL,
  `descripcion` VARCHAR(200) NULL,
  `precio` DECIMAL(14,4) NULL,
  PRIMARY KEY (`idproducto`));
END$$

DELIMITER ;

-- -----------------------------------------------------
-- procedure eliminarCombo
-- -----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `eliminarCombo`(in vig boolean,
in idc int(11))
BEGIN
      update combo set Vigente=vig WHERE idCombo=idc;

      SET @fecha = Date(NOW());
      select max(Fecha) into @fecha from item where idCombo=idc;
      update item set Estado=vig WHERE idCombo=idc and Fecha=@fecha;
END$$

DELIMITER ;

-- -----------------------------------------------------
-- procedure eliminarGrupoMod
-- -----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `eliminarGrupoMod`(in est
boolean,in idg int(11))
BEGIN
      update grupom set Estado=est WHERE idgrupom=idg;
END$$

DELIMITER ;

-- -----------------------------------------------------
-- procedure eliminarPermiso
-- -----------------------------------------------------

DELIMITER $$
```

```
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `eliminarPermiso`(in idr
int(11))
BEGIN
      DELETE FROM permisos WHERE idRol = idr and idUsuario =-1;
END$$

DELIMITER ;

-- -----------------------------------------------------
-- procedure eliminarPermisoUsuario
-- -----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `eliminarPermisoUsuario`(in idu
int(11))
BEGIN
      DELETE FROM permisos where idUsuario=idu;
END$$

DELIMITER ;

-- -----------------------------------------------------
-- procedure eliminarPersona
-- -----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `eliminarPersona`(in est
boolean, in idp int(11))
BEGIN
      update persona set Estado=est WHERE idPersona=idp;
END$$

DELIMITER ;

-- -----------------------------------------------------
-- procedure eliminarProducto
-- -----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `eliminarProducto`(in est
boolean, in idp int(11))
BEGIN
      update producto set Estado=est WHERE idProducto=idp;
      update item set Estado=est WHERE idProducto=idp;
END$$

DELIMITER ;

-- -----------------------------------------------------
-- procedure eliminarProductosCombo
-- -----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `eliminarProductosCombo`(in idc
int(11))
```

```
BEGIN
	SET @fecha = Date(NOW());
	select distinct Fecha into @fecha from item where idCombo=idc and
Estado=true;
	update item set Estado=false where idCombo=idc and Fecha=@fecha;
END$$

DELIMITER ;

-- -----------------------------------------------------
-- procedure eliminarUsuario
-- -----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `eliminarUsuario`(in est
boolean, in idu int(11))
BEGIN
	update usuario set Estado=est WHERE idUsuario=idu;
END$$

DELIMITER ;

-- -----------------------------------------------------
-- procedure estadoEntregarPedido
-- -----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `estadoEntregarPedido`(in idp
int(11))
BEGIN
	UPDATE pedido
	SET	idEstado = 5, HoraEntregada = curtime()
	WHERE idpedido = idp;

END$$

DELIMITER ;

-- -----------------------------------------------------
-- procedure getClientePedido
-- -----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `getClientePedido`(in idp
int(11))
BEGIN
	select c.nombres as cliente, c.nit
	from pedido p
	left join cliente c on c.idCliente=p.idCliente
	where p.idpedido = idp;
END$$

DELIMITER ;

-- -----------------------------------------------------
-- procedure getClienteUsuario
-- -----------------------------------------------------
```

```
DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `getClienteUsuario`(in idu
int(11))
BEGIN
	select idCliente
	from cliente
	where idUsuario=idu and estado=true;
END$$

DELIMITER ;

-- -----------------------------------------------------
-- procedure getCombo
-- -----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `getCombo`(in idc int(11))
BEGIN
	SELECT c.idCombo, c.Descripcion, c.Vigente, c.Fecha, img.Titulo,
img.Foto, d.descuento
	from combo c
	left join imagencombo imc on c.idCombo = imc.idCombo and imc.estado =
true
	left join imagen img on imc.idimagen = img.idimagen
	left join descuento d on c.idcombo=d.idcombo and d.estado=true
	where c.idCombo=idc;
END$$

DELIMITER ;

-- -----------------------------------------------------
-- procedure getDatosPersonaUsuario
-- -----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `getDatosPersonaUsuario`(in
idusuario int(11))
BEGIN
	select p.idPersona, p.Nombres, p.ApPaterno, p.ApMaterno, p.Telefonos,
p.CI, p.LugarNacimiento, p.Email, p.Direccion, p.Estado, p.FechaRegistro,
p.FechaNacimiento from persona p inner join usuario u on u.idPersona =
p.idPersona where u.idUsuario = idusuario;
END$$

DELIMITER ;

-- -----------------------------------------------------
-- procedure getDescuentoCombo
-- -----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `getDescuentoCombo`(in idc
int(11))
BEGIN
	SELECT c.idCombo, d.descuento
```

```
	from combo c
	left join descuento d on c.idcombo=d.idcombo and d.estado=true
	where c.idCombo=idc;
END$$

DELIMITER ;

-- -----------------------------------------------------
-- procedure getDetDocumento
-- -----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `getDetDocumento`(in idd
int(11))
BEGIN
	select
	dpd.iditem,
	dpd.idx,
	ifnull(concat('<b>',cmb.Descripcion,'</b>'),concat('<b>',pr.Descripcio
n,'</b>')) as 'Combo',
	-- '' as 'Detalle',
	max(dpd.Cantidad) as 'Cantidad' ,
	sum(dpd.monto) as 'monto',
	max(dpd.Descuento) as 'descuento',
	max(dpd.subTotal) as 'subTotal'
	from documento p
	left join det_documento dpd on dpd.idDocumento=p.idDocumento
	left join item i on i.idItem=dpd.iditem
	left join combo cmb on cmb.idCombo=i.idCombo
	left join producto pr on pr.idProducto=i.idProducto
	where p.idDocumento=idd
	group by dpd.Idx
	union
	select
		dpd.iditem,
		dpd.idx,
		-- '',
		concat(' * ',pr.Descripcion),
		0,
		0,
		0,
		0
	from documento p
	left join det_documento dpd on dpd.idDocumento=p.idDocumento
	left join item i on i.idItem=dpd.iditem
	left join combo cmb on cmb.idCombo=i.idCombo
	left join producto pr on pr.idProducto=i.idProducto
	where p.idDocumento=idd and not isnull(cmb.Descripcion)
	order by idx,Combo desc;
END$$

DELIMITER ;

-- -----------------------------------------------------
-- procedure getDetPedido
-- -----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
```

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `getDetPedido`(in idp int(11))
BEGIN
	select i.iditem, p.Nombre, c.Descripcion, dp.Cantidad, dp.Descuento,
dp.Precio, dp.idx
	from det_pedido dp
	left join item i on i.idItem = dp.iditem
	left join producto p on i.idProducto = p.idProducto
	left join combo c on c.idCombo = i.idCombo
	where dp.idPedido = idp and dp.Estado = true;
END$$

DELIMITER ;

-- -----------------------------------------------------
-- procedure getDetallePedidoCompleto
-- -----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `getDetallePedidoCompleto`(in
idp int(11))
BEGIN
select
	dpd.iditem,
	dpd.idx,
	ifnull(concat('<b>',cmb.Descripcion,'</b>'),concat('<b>',pr.Descripcio
n,'</b>')) as 'Combo',
	-- '' as 'Detalle',
	max(dpd.Cantidad) as 'Cantidad' ,
	sum(dpd.Precio) as 'precio',
	max(dpd.Descuento) as 'descuento',
	(sum(Precio)*max(Cantidad))-(max(Descuento)*max(Cantidad)) as
'precioSubTotal'
from pedido p
left join det_pedido dpd on dpd.idPedido=p.idpedido
left join item i on i.idItem=dpd.iditem
left join combo cmb on cmb.idCombo=i.idCombo
left join producto pr on pr.idProducto=i.idProducto
where p.idpedido=idp
group by dpd.Idx
union
select
	dpd.iditem,
	dpd.idx,
	-- '',
	concat(' * ',pr.Descripcion),
	0,
	dpd.precio,
	0,
	0
from pedido p
left join det_pedido dpd on dpd.idPedido=p.idpedido
left join item i on i.idItem=dpd.iditem
left join combo cmb on cmb.idCombo=i.idCombo
left join producto pr on pr.idProducto=i.idProducto
where p.idpedido=idp and not isnull(cmb.Descripcion)
order by idx,Combo desc;
END$$

DELIMITER ;
```

```
-- -----------------------------------------------------
-- procedure getDgrupoM
-- -----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `getDgrupoM`(in idp int(11))
BEGIN
	select estado, idGrupom, idProducto from dgrupom where idProducto =
idp;
END$$

DELIMITER ;

-- -----------------------------------------------------
-- procedure getEmpleadoUsuario
-- -----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `getEmpleadoUsuario`(in idu
int(11))
BEGIN
	select idEmpleado
	from empleado
	where idUsuario=idu and estado=true;
END$$

DELIMITER ;

-- -----------------------------------------------------
-- procedure getEstablecimiento
-- -----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `getEstablecimiento`(in ide
int(11))
BEGIN
	Select e.idEstablecimiento, e.Nit, e.Nombre, e.Telefono, e.Direccion,
e.Email, e.Estado, im.titulo, im.foto
	from establecimiento e
	left join logoEstablecimiento le on e.idEstablecimiento =
le.idEstablecimiento
	left join imagen im on le.idimagen = im.idimagen
	where le.estado=true and e.idEstablecimiento=ide;
END$$

DELIMITER ;

-- -----------------------------------------------------
-- procedure getEstablecimientos
-- -----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `getEstablecimientos`()
BEGIN
```

```
	Select e.idEstablecimiento, e.Nit, e.Nombre, e.Telefono, e.Email,
e.Estado, im.titulo, im.foto
	from establecimiento e
	left join logoEstablecimiento le on e.idEstablecimiento =
le.idEstablecimiento
	left join imagen im on le.idimagen = im.idimagen
	where le.estado=true;
END$$

DELIMITER ;

-- -----------------------------------------------------
-- procedure getEstadoCombo
-- -----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `getEstadoCombo`(in idc int(11))
BEGIN
	select idCombo, Vigente
	from combo
	where idCombo = idc;
END$$

DELIMITER ;

-- -----------------------------------------------------
-- procedure getEstadoEstablecimiento
-- -----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `getEstadoEstablecimiento`(in
ide int(11))
BEGIN
	select estado
	from establecimiento
	where idEstablecimiento=ide;
END$$

DELIMITER ;

-- -----------------------------------------------------
-- procedure getEstadoGrupoM
-- -----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `getEstadoGrupoM`(in idg
int(11))
BEGIN
	select idGrupoM, Detalle, Estado
	from grupoM
	where idGrupoM = idg;
END$$

DELIMITER ;

-- -----------------------------------------------------
-- procedure getEstadoPortalPedidos
```

```
-- -----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `getEstadoPortalPedidos`()
BEGIN
	select idMenu, estado
	from menu_sistema
	where idMenu=5;
END$$

DELIMITER ;

-- -----------------------------------------------------
-- procedure getEstadoProducto
-- -----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `getEstadoProducto`(in idp
int(11))
BEGIN
	select idProducto, Estado
	from producto
	where idProducto = idp;
END$$

DELIMITER ;

-- -----------------------------------------------------
-- procedure getGrupoMdeCombo
-- -----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `getGrupoMdeCombo`(in idc
int(11))
BEGIN
	Select max(gm.Detalle) as 'Detalle', dg.idGrupoM
	from item it
	left join Producto p on it.idProducto=p.idProducto
	left join dgrupom dg on p.idProducto=dg.idProducto
	left join grupom gm on dg.idGrupoM=gm.idGrupoM
	where it.idCombo=idc and it.Estado=true and dg.estado=true
	group by Detalle;
END$$

DELIMITER ;

-- -----------------------------------------------------
-- procedure getGruposProdCombo
-- -----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `getGruposProdCombo`()
BEGIN
	select distinct gm.Detalle as descripcion, i.idCombo
		from item i
		left join producto p on p.idProducto=i.idProducto
```

```
            left join dgrupom dg on p.idProducto=dg.idProducto
            left join grupom gm on dg.idGrupoM=gm.idGrupoM
      where i.idCombo is not null and i.Estado=true;
END$$

DELIMITER ;

-- ----------------------------------------------------
-- procedure getHorarios
-- ----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `getHorarios`()
BEGIN
      select idEstablecimiento, horaInicio, horaFin, horaOrdenar,
mensajeEfectivizacion
      from establecimiento
      where idestablecimiento = 2;
END$$

DELIMITER ;

-- ----------------------------------------------------
-- procedure getIdProducto
-- ----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `getIdProducto`(in nom
varchar(45))
BEGIN
      select idProducto from producto where Nombre = nom;
END$$

DELIMITER ;

-- ----------------------------------------------------
-- procedure getIdRol
-- ----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `getIdRol`(in nom varchar(45))
BEGIN
      SELECT idRol from rol where nombre=nom;
END$$

DELIMITER ;

-- ----------------------------------------------------
-- procedure getIdUsuario
-- ----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `getIdUsuario`(in log
varchar(45))
BEGIN
      select idUsuario from usuario where Login=log;
```

```
END$$

DELIMITER ;

-- ----------------------------------------------------
-- procedure getImagenCombo
-- ----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `getImagenCombo`(in idc int(11))
BEGIN
      SELECT c.idCombo, c.Descripcion, img.Titulo, img.Foto, d.descuento
      from combo c
      left join imagencombo imc on c.idCombo = imc.idCombo and imc.estado =
true
      left join imagen img on imc.idimagen = img.idimagen
      left join descuento d on c.idcombo=d.idcombo and d.estado=true
      where c.idCombo=idc;
END$$

DELIMITER ;

-- ----------------------------------------------------
-- procedure getMenuHijosPermisos2
-- ----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `getMenuHijosPermisos2`(in
pidusuario int(11))
BEGIN
    Set @P=3;
      select distinct ifnull(idrol,3) into @P from permisos where
idusuario=pidusuario;
      if @P=3 then
            begin
                  Select ms.* from menu_sistema ms inner join permisos pr
on pr.idUsuario = -1 and pr.idrol=@P and pr.asignado=true where
ms.idmenu=pr.idmenu and not MenuPadre is null;
            end;
      else
            begin
                  Select ms.* from menu_sistema ms inner join permisos pr
on pr.idUsuario = pidusuario and pr.asignado=true where ms.idmenu=pr.idmenu
and not MenuPadre is null;
            end;
      end if;
END$$

DELIMITER ;

-- ----------------------------------------------------
-- procedure getMenuPermisos2
-- ----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `getMenuPermisos2`(in pidusuario
int(11))
```

```
begin
    Set @P=3;
       select distinct ifnull(idrol,3) into @P from permisos where
idusuario=pidusuario;
       Set @usuario= if(@P=3,-1,pidusuario);

-- Consulta para obtener el menu padre para un Usuario específico
             select ms.*
             from rol rl
             left join permisos pr on pr.idUsuario=@usuario and
pr.idRol=rl.idRol
             left join menu_sistema ms on ms.idmenu=pr.idmenu and
isnull(ms.MenuPadre) and ms.Estado=true
             where rl.idRol=@P and not isnull(ms.idmenu);

END$$

DELIMITER ;

-- -----------------------------------------------------
-- procedure getNombreNitDoc
-- -----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `getNombreNitDoc`(in iddoc
int(11))
BEGIN
       select nombre, ci, fecha, monto
       from documento
       where iddocumento = iddoc;
END$$

DELIMITER ;

-- -----------------------------------------------------
-- procedure getPedido
-- -----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `getPedido`(in idp int(11))
BEGIN
       select p.idPedido,p.Codigo, p.montoReserva,
p.Fecha,p.HoraReservada,p.FechaValida, p.HoraValida,
p.NroPedido,p.HoraEntregada, if(p.Llevar,'Si','Mesa') as Llevar,
c.idcliente, c.nombres as cliente, c.nit, c.telefono, e.idestado, e.nombre
as estado, concat(per.nombres,' ',per.appaterno) as empleado
       from pedido p
       left join cliente c on c.idCliente=p.idCliente
       left join estado e on e.idestado=p.idEstado
       left join empleado em on p.idempleado=em.idempleado
       left join usuario u on em.idusuario=u.idusuario
       left join persona per on per.idpersona=u.idpersona
       where p.idpedido = idp;
END$$

DELIMITER ;

-- -----------------------------------------------------
```

```
-- procedure getPedidoMontoReserva
-- ----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `getPedidoMontoReserva`(in idp
int(11))
BEGIN
	select p.idPedido, p.NroPedido, p.montoReserva
	from pedido p
	where p.idpedido=idp;
END$$

DELIMITER ;

-- ----------------------------------------------------
-- procedure getPermisosDeRol
-- ----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `getPermisosDeRol`(in idr
int(11))
BEGIN
	select idmenu, idRol, idUsuario, asignado from permisos where
idRol=idr and idUsuario = -1 and asignado=true;
END$$

DELIMITER ;

-- ----------------------------------------------------
-- procedure getPermisosPorUsuario2
-- ----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `getPermisosPorUsuario2`(in idu
int(11))
BEGIN
	set @pp=3;
	select distinct ifnull(idRol,3) into @pp from permisos where idUsuario
= idu;
	set @idUs=idu;
	if @pp=3 then
		begin
			set @idUs=-1;
		end;
	end if;
	Select m.idmenu, m.nombre,
			@pp as idRol,
			@idUs as idusuario,
			ifnull(p.asignado,false) as asignado
		from menu_sistema m
		left join permisos p on p.idmenu=m.idmenu and p.idUsuario=@idUs
and idRol=@pp
		order by m.idmenu;
END$$

DELIMITER ;
```

```
-- -----------------------------------------------------
-- procedure getPermisosRol2
-- -----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `getPermisosRol2`(in idr
int(11))
BEGIN
	set @idRs=idr;
	Select distinct m.idmenu,
concat(if(isnull(m.MenuPadre),'',concat((select mm.nombre from menu_sistema
mm where mm.idmenu=m.MenuPadre),' -->')),m.nombre) as nombre,
                   @idRs as idRol,
                   -1 as idUsuario,
                   ifnull(p.asignado,false) as asignado
             from menu_sistema m
             left join permisos p on p.idmenu=m.idmenu and p.idRol=@idRs and
p.idUsuario=-1
             order by m.idmenu;
END$$

DELIMITER ;

-- -----------------------------------------------------
-- procedure getPersona
-- -----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `getPersona`(in id int(11))
BEGIN
	Select idPersona, Nombres, ApPaterno, ApMaterno, Telefonos, CI,
LugarNacimiento, Email, Direccion, Estado, FechaRegistro, FechaNacimiento
from persona where idPersona = id;
END$$

DELIMITER ;

-- -----------------------------------------------------
-- procedure getPrecioCombo
-- -----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `getPrecioCombo`(in idc int(11))
BEGIN
	select sum(p.monto) as monto
             from item i
             left join precio p on p.iditem=i.idItem
	where i.estado is true and i.idCombo = idc;
END$$

DELIMITER ;

-- -----------------------------------------------------
-- procedure getPrecioProducto
-- -----------------------------------------------------

DELIMITER $$
```

```
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `getPrecioProducto`(in idp
int(11))
BEGIN
	select distinct pr.Monto
	from producto p
	left join precio pr on p.idProducto = pr.idProducto and pr.estado =
true
	where p.idProducto = idp;
END$$

DELIMITER ;

-- -----------------------------------------------------
-- procedure getProducto
-- -----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `getProducto`(in idp int(11))
BEGIN
	select distinct gm.Detalle, gm.idGrupoM, it.iditem, p.idProducto,
p.Nombre, p.Descripcion, p.Estado, p.Envase, pr.Monto, im.titulo, im.foto
	from producto p
	left join item it on it.idProducto = p.idProducto and it.idCombo is
null
	left join precio pr on p.idProducto = pr.idProducto and pr.estado =
true
	left join imagenproducto imp on p.idProducto = imp.idProducto and
imp.estado = true
	left join imagen im on imp.idimagen = im.idimagen
	left join dgrupom dgm on dgm.idProducto = p.idProducto
	left join grupom gm on gm.idGrupoM = dgm.idGrupoM
	where p.idProducto = idp;
END$$

DELIMITER ;

-- -----------------------------------------------------
-- procedure getProductoCombo
-- -----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `getProductoCombo`(in idp
int(11), in idc int(11))
BEGIN
	select distinct it.iditem, p.idProducto, p.Nombre, p.Descripcion,
p.Estado, p.Envase, pr.Monto
	from producto p
	left join item it on it.idProducto = p.idProducto
	left join precio pr on p.idProducto = pr.idProducto and pr.estado =
true
	where it.idProducto = idp and it.idcombo = idc and it.estado = true;
END$$

DELIMITER ;

-- -----------------------------------------------------
-- procedure getProductosCombo
```

```
-- -----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `getProductosCombo`(in idc
int(11))
BEGIN
	Select p.Nombre, p.Descripcion, pr.Monto, gm.Detalle, dg.estado as
asignado, it.Estado, it.Fecha, im.Foto, im.Titulo
	from item it
	left join Producto p on it.idProducto=p.idProducto
	left join dgrupom dg on p.idProducto=dg.idProducto
	left join grupom gm on dg.idGrupoM=gm.idGrupoM
	left join imagenproducto imp on p.idProducto=imp.idProducto and
imp.Estado=true
	left join imagen im on imp.idimagen=im.idimagen
	left join precio pr on p.idProducto=pr.idProducto and pr.Estado=true
	where it.idCombo=idc and it.Estado=true;
END$$

DELIMITER ;

-- -----------------------------------------------------
-- procedure getProductosFalseCombo
-- -----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `getProductosFalseCombo`(in idc
int(11))
BEGIN
	select distinct PR.IDPRODUCTO, Pr.Nombre, pr.descripcion, pr.envase,
pre.Monto,gm.idGrupoM, gm.Detalle, im.Foto, im.Titulo
	from producto PR
		left join item it on PR.idproducto=it.idproducto
	  left join precio pre on pre.idProducto = PR.idproducto and
pre.estado=true
		left join dgrupom dg on pr.idProducto=dg.idProducto
		left join grupom gm on dg.idGrupoM=gm.idGrupoM
		left join imagenproducto imp on pr.idProducto=imp.idProducto
and imp.Estado=true
		left join imagen im on imp.idimagen=im.idimagen
	where it.idcombo=idc and pr.Estado=true and dg.estado=false
	order by 3 desc, 6 asc;
END$$

DELIMITER ;

-- -----------------------------------------------------
-- procedure getProductosGrupo
-- -----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `getProductosGrupo`(in idg
int(11))
BEGIN
	select idProducto
	from dgrupom
	where idgrupom = idg;
```

```
END$$

DELIMITER ;

-- -----------------------------------------------------
-- procedure getProductosTrueCombo
-- -----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `getProductosTrueCombo`(in idc
int(11))
BEGIN
	select distinct PR.IDPRODUCTO, Pr.Nombre, pr.descripcion, pr.envase,
pre.Monto,gm.idGrupoM, gm.Detalle, im.Foto, im.Titulo
	from producto PR
		left join item it on PR.idproducto=it.idproducto
	 left join precio pre on pre.idProducto = PR.idproducto and
pre.estado=true
		left join dgrupom dg on pr.idProducto=dg.idProducto
		left join grupom gm on dg.idGrupoM=gm.idGrupoM
		left join imagenproducto imp on pr.idProducto=imp.idProducto
and imp.Estado=true
		left join imagen im on imp.idimagen=im.idimagen
	where it.idcombo=idc and pr.Estado=true and dg.estado=true
	order by 3 desc, 6 asc;
END$$

DELIMITER ;

-- -----------------------------------------------------
-- procedure getRol
-- -----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `getRol`(in idr int(11))
BEGIN
	SELECT * from rol where idRol = idr;
END$$

DELIMITER ;

-- -----------------------------------------------------
-- procedure getUsuario
-- -----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `getUsuario`(in id int(11))
BEGIN
	Select idUsuario, Login, Clave, idPersona,
		Estado, FechaRegistro, TituloFoto, Imagen,
idEstablecimiento
	from usuario
	where idUsuario = id;
END$$

DELIMITER ;
```

```
-- ----------------------------------------------------
-- procedure getUsuariosPersona
-- ----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `getUsuariosPersona`(in idp
int(11))
BEGIN
	select idUsuario, Estado from usuario where idPersona=idp;
END$$

DELIMITER ;

-- ----------------------------------------------------
-- procedure getUsuariosRolPersona2
-- ----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `getUsuariosRolPersona2`(in idp
int(11))
BEGIN
	select distinct ifnull(r.Nombre,"Cliente") as Nombre,
			u.idUsuario, u.Login, u.Clave, u.idPersona,
			u.Estado, u.FechaRegistro, u.TituloFoto, u.Imagen,
e.Nombre as Establecimiento
	from usuario u
		left join permisos p ON p.idUsuario=u.idUsuario
		left join Rol r on r.idRol = p.idRol
		left join Establecimiento e on u.idEstablecimiento =
e.idEstablecimiento
	where u.idpersona=idp
	order by idUsuario;
END$$

DELIMITER ;

-- ----------------------------------------------------
-- procedure inactivarCliente
-- ----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `inactivarCliente`(in idu
int(11))
BEGIN
	update cliente set Estado=false WHERE idUsuario=idu;
END$$

DELIMITER ;

-- ----------------------------------------------------
-- procedure inactivarEmpleado
-- ----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `inactivarEmpleado`(in idu
int(11))
```

```
BEGIN
      update empleado set Estado=false WHERE idUsuario=idu;
END$$

DELIMITER ;

-- -----------------------------------------------------
-- procedure ingresoTotalRango
-- -----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `ingresoTotalRango`(in FInicio
Date, in FFinal Date)
BEGIN
select sum(Monto) as 'Ingresos' from documento where Fecha Between FInicio
and FFinal;

END$$

DELIMITER ;

-- -----------------------------------------------------
-- function insertarCarrito
-- -----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` FUNCTION `insertarCarrito`(fecha
date,horaReservada time,fechaValida date,horaValida time,idCliente
int(11),idestado int(11)) RETURNS int(11)
BEGIN
      Set @idc=0;
      SET @fecha = Date(NOW());

      INSERT INTO `dbmyburguer`.`carrito`
      (`Fecha`,`HoraReservada`,`FechaValida`,`HoraValida`,`idCliente`,`idest
ado`)
      VALUES
      (fecha,horaReservada,fechaValida,horaValida,idCliente,idestado);

      Select max(idreserva) into @idc from carrito;

RETURN @idc;
END$$

DELIMITER ;

-- -----------------------------------------------------
-- procedure insertarCliente
-- -----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `insertarCliente`(in nom
varchar(100), in ni varchar(45), in tel varchar(45), in est boolean, in idu
int(11))
BEGIN
      insert into cliente (Nombres, Nit, Telefono, Estado, idUsuario) values
(nom , ni, tel, est, idu);
```

```
END$$

DELIMITER ;

-- ----------------------------------------------------
-- function insertarCombo
-- ----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` FUNCTION `insertarCombo`(vDescripcion
Varchar(200), vFoto mediumblob, vtitulo varchar(45)) RETURNS int(11)
BEGIN
	Set @idc=0;
	SET @fecha = Date(NOW());

	INSERT INTO combo (Descripcion, Vigente, Fecha)
		VALUES (vDescripcion, true, @fecha);

	Select max(idCombo) into @idc from combo;

	INSERT INTO imagen (Titulo, Estado, Foto, Fecha)
		VALUES (vtitulo, true, vFoto, @fecha);

	SET @idImagen=0;
	Select max(idimagen) into @idImagen from imagen;

	INSERT INTO imagencombo (Fecha, Estado, idcombo, idimagen)
	VALUES(@fecha,true,@idc,@idImagen);

RETURN @idc;
END$$

DELIMITER ;

-- ----------------------------------------------------
-- procedure insertarComboItem
-- ----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `insertarComboItem`(in idp
int(11), in idc int(11))
BEGIN
SET @fecha = Date(NOW());
	INSERT INTO item (Fecha, Estado, idProducto, idCombo)
	VALUES (@fecha, true, idp, idc);

END$$

DELIMITER ;

-- ----------------------------------------------------
-- procedure insertarDescuento
-- ----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `insertarDescuento`(in idc
int(11), in des decimal(14,2))
```

```
BEGIN
	insert into descuento(idCombo, descuento)
	values (idc, des);
END$$

DELIMITER ;

-- -----------------------------------------------------
-- procedure insertarDetDocumento
-- -----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `insertarDetDocumento`(in monto
decimal(14,2),in descuento decimal(14,2),in cantidad int(10),in subTotal
decimal(14,2),in iddoc int(11),in idi int(11), in ix int(11))
BEGIN
	INSERT INTO
det_documento(monto,descuento,cantidad,subTotal,idDocumento,idItem,idx)
	VALUES(monto,descuento,cantidad,subTotal,iddoc,idi,ix);
END$$

DELIMITER ;

-- -----------------------------------------------------
-- procedure insertarDet_Pedido
-- -----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `insertarDet_Pedido`(in idp
int(11), in cant int(2), in idi int(11), in precio decimal(14,2), in des
decimal(14,2), in ix int(11))
BEGIN
	INSERT INTO
det_pedido(idpedido,Cantidad,Precio,Estado,iditem,descuento,idx)
	VALUES(idp,cant,precio,true,idi,des,ix);
END$$

DELIMITER ;

-- -----------------------------------------------------
-- procedure insertarDgrupoM
-- -----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `insertarDgrupoM`(in est
boolean, in idgm int(11), in idp int(11))
BEGIN
	insert into dgrupom (estado, idGrupoM, idProducto) values (est, idgm,
idp);
END$$

DELIMITER ;

-- -----------------------------------------------------
-- function insertarDocumento
-- -----------------------------------------------------
```

```
DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` FUNCTION `insertarDocumento`(monto
decimal(14,2),det varchar(45),nombre varchar(45),ci varchar(45),idcli
int(11),idtd int(11),idp int(11)) RETURNS int(11)
BEGIN
	Set @iddoc=0;
	INSERT INTO
documento(Fecha,Monto,Detalle,Nombre,CI,idCliente,idtipo_doc,idPedido)
	VALUES(curdate(),monto,det,nombre,ci,idcli,idtd,idp);

	Select max(idDocumento) into @iddoc from documento;
RETURN @iddoc;
END$$

DELIMITER ;

-- -----------------------------------------------------
-- procedure insertarEmpleado
-- -----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `insertarEmpleado`(in fechaR
date, in est boolean, in idu int(11))
BEGIN
	insert into empleado (FechaRegistro, Estado, idUsuario) values
(fechaR, est, idu);
END$$

DELIMITER ;

-- -----------------------------------------------------
-- procedure insertarEstablecimiento
-- -----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `insertarEstablecimiento`(in ni
varchar(10), nom varchar(45), dir varchar(45), tel varchar(45), emai
varchar(45), in vFoto mediumblob, in vtitulo varchar(45))
BEGIN
	SET @ide = 0;
	SET @fecha = Date(NOW());

	insert into Establecimiento (Nit, Nombre, Direccion, Telefono, Email,
Estado)
		values (ni, nom, dir, tel, emai, true);

	Select max(idEstablecimiento) into @ide from Establecimiento;

	INSERT INTO imagen (Titulo, Estado, Foto, Fecha)
		VALUES (vtitulo, true, vFoto, @fecha);

	SET @idImagen=0;
	Select max(idimagen) into @idImagen from imagen;

	INSERT INTO logoEstablecimiento (Fecha, Estado, idEstablecimiento,
idimagen)
		VALUES(@fecha,true,@ide,@idImagen);
```

```
END$$

DELIMITER ;

-- ----------------------------------------------------
-- procedure insertarGrupoM
-- ----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `insertarGrupoM`(in det
varchar(45), est boolean)
BEGIN
	insert into grupom (detalle, estado) values(det, est);
END$$

DELIMITER ;

-- ----------------------------------------------------
-- procedure insertarItemXGrupoCombo
-- ----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `insertarItemXGrupoCombo`(in idg
int(11), in idc int(11))
BEGIN
SET @fecha = Date(NOW());
	Insert into Item (Fecha, Estado, idProducto, idCombo)
		select @fecha,true, p.idProducto, idc
		from producto p
			left join dgrupom dg on p.idProducto=dg.idProducto and
dg.estado=true
		where dg.idGrupoM=idg;
END$$

-- ----------------------------------------------------
-- function insertarPedido
-- ----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` FUNCTION `insertarPedido`(hv time, lle
boolean, idcli int(11), ides int(11), idem int(11), montoRe float, tprepara
time) RETURNS int(11)
BEGIN
	set @idp=0;
	set @nropedido =0;
	select count(nropedido)+1 into @nropedido from pedido where
Fecha=curdate();
	set @cod=concat(idcli,'-',hv,'-',@nropedido);

	INSERT INTO pedido(Codigo, Fecha, HoraReservada, FechaValida,
HoraValida,NroPedido,
HoraEntregada,Llevar,idcliente,idEmpleado,idestado,hora_ordenar,montoReserva
)
	VALUES(@cod,curdate(),curtime(),curdate(),hv,@nropedido,null,lle,idcli
,idem,ides,tprepara,montoRe);
```

```
      Select max(idpedido) into @idp from pedido;
RETURN @idp;
END$$

DELIMITER ;

-- ----------------------------------------------------
-- procedure insertarPersona
-- ----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `insertarPersona`(in nombre
varchar(25), paterno varchar(25), materno varchar(25), telefono
varchar(100), ci varchar(15), lugar varchar(45), email varchar(45),
direccion varchar(100), estado boolean, nac date, reg date)
BEGIN
      insert into persona (Nombres, ApPaterno, ApMaterno, Telefonos, CI,
LugarNacimiento, Email, Direccion, Estado, FechaNacimiento, FechaRegistro)
values(nombre, paterno, materno, telefono, ci, lugar, email, direccion,
estado, nac, reg);
END$$

DELIMITER ;

-- ----------------------------------------------------
-- procedure insertarProducto
-- ----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `insertarProducto`(in vnombre
varchar(45), in vDescripcion Varchar(250), in vEnvase Varchar(45), in vMonto
decimal(14,2), in vFoto mediumblob, in vtitulo varchar(45))
BEGIN
  SET @idp = 0;
  SET @fecha = Date(NOW());

  INSERT INTO producto (Nombre, Descripcion, Estado, Envase)
            VALUES (vNombre, vDescripcion, true, vEnvase);

  Select max(idProducto) into @idp from producto;

  INSERT INTO item (Fecha, Estado, idProducto, idCombo)
            VALUES (@fecha, true, @idp, null);

  INSERT INTO imagen (Titulo, Estado, Foto, Fecha)
            VALUES (vtitulo, true, vFoto, @fecha);

  SET @idImagen=0;
  Select max(idimagen) into @idImagen from imagen;

  INSERT INTO imagenproducto (Fecha, Estado, idproducto, idimagen)
      VALUES(@fecha, true, @idp, @idImagen);

  INSERT INTO precio (Monto, Fecha, Estado, idProducto)
            VALUES(vMonto, @fecha, true, @idp);
END$$

DELIMITER ;
```

```
-- -----------------------------------------------------
-- procedure insertarRol
-- -----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `insertarRol`(in nom
varchar(45), in est boolean)
BEGIN
	insert into rol (Nombre, Estado) values (nom, est);
END$$

DELIMITER ;

-- -----------------------------------------------------
-- procedure insertarUsuario
-- -----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `insertarUsuario`(in log
varchar(45), cl varchar(45), in idp int(11), in est boolean, in reg
varchar(45), in tfo varchar(45), in img mediumblob, in ide int(11))
BEGIN
	insert into usuario (Login, Clave, idPersona, Estado, FechaRegistro,
TituloFoto, Imagen, idEstablecimiento) values (log, cl, idp, est, reg, tfo,
img, ide);
END$$

DELIMITER ;

-- -----------------------------------------------------
-- procedure insertatemporal
-- -----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `insertatemporal`(in idp int, in
descrip varchar(200), in prec decimal(14,4))
BEGIN
	insert into tablatemporal(idproducto,descripcion,precio) values
(idp,descrip,prec);
END$$

DELIMITER ;

-- -----------------------------------------------------
-- procedure listaProductosComboFalse
-- -----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `listaProductosComboFalse`(in
idc int(11))
BEGIN
	select distinct PR.IDPRODUCTO, Pr.Nombre,
if(isnull(it.idProducto),false,true) as Asignado,
		pre.Monto,gm.idGrupoM, max(gm.Detalle) as 'Detalle', im.Foto,
im.Titulo, dg.estado
```

```
       from producto PR
             left join item it on PR.idproducto=it.idproducto and
it.idcombo=idc
        left join precio pre on pre.idProducto = PR.idproducto and
pre.estado=true
             left join dgrupom dg on pr.idProducto=dg.idProducto
             left join grupom gm on dg.idGrupoM=gm.idGrupoM
             left join imagenproducto imp on pr.idProducto=imp.idProducto
and imp.Estado=true
             left join imagen im on imp.idimagen=im.idimagen
       where pr.Estado=true and dg.estado=false
       Group by Detalle
       order by 3, 6 asc;
END$$

DELIMITER ;

-- -----------------------------------------------------
-- procedure listaProductosComboTrue
-- -----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `listaProductosComboTrue`(in idc
int(11))
begin
       select distinct PR.IDPRODUCTO, Pr.Nombre,
if(isnull(it.idProducto),false,true) as Asignado,
             pre.Monto,gm.idGrupoM,gm.Detalle, im.Foto, im.Titulo, dg.estado
       from producto PR
             left join item it on PR.idproducto=it.idproducto and
it.idcombo=idc and it.Estado=true
        left join precio pre on pre.idProducto = PR.idproducto and
pre.estado=true
             left join dgrupom dg on pr.idProducto=dg.idProducto
             left join grupom gm on dg.idGrupoM=gm.idGrupoM
             left join imagenproducto imp on pr.idProducto=imp.idProducto
and imp.Estado=true
             left join imagen im on imp.idimagen=im.idimagen
       where pr.Estado=true and dg.estado=true
       order by 6;
end$$

DELIMITER ;

-- -----------------------------------------------------
-- procedure listarClientes
-- -----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `listarClientes`()
BEGIN
       select idCliente, nombres, nit, telefono, estado, idusuario
       from cliente
       where estado=true;
END$$

DELIMITER ;
```

```
-- -----------------------------------------------------
-- procedure listarCombos
-- -----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `listarCombos`()
BEGIN
	select c.idCombo, c.Descripcion, c.Vigente, c.Fecha, im.Foto,
im.Titulo
	from combo c
	left join imagencombo imc on c.idCombo = imc.idCombo and imc.Estado =
true
	left join imagen im on imc.idimagen = im.idimagen;
END$$

DELIMITER ;

-- -----------------------------------------------------
-- procedure listarCombosDisponibles
-- -----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `listarCombosDisponibles`()
BEGIN
	select c.idCombo, c.Descripcion, c.Vigente, c.Fecha, im.Foto,
im.Titulo
	from combo c
	left join imagencombo imc on c.idCombo = imc.idCombo and imc.Estado =
true
	left join imagen im on imc.idimagen = im.idimagen
	where c.vigente=true;
END$$

DELIMITER ;

-- -----------------------------------------------------
-- procedure listarGrupoMdeProductosTrue
-- -----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `listarGrupoMdeProductosTrue`()
BEGIN
	select distinct g.idGrupoM, g.Detalle
		from grupom g
		left join dgrupom dg on g.idGrupoM=dg.idGrupoM
		where dg.estado=true
	order by detalle;
END$$

DELIMITER ;

-- -----------------------------------------------------
-- procedure listarGrupoMod
-- -----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
```

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `listarGrupoMod`()
BEGIN
	select idGrupoM, Detalle, Estado
	from grupom
	order by detalle;
END$$

DELIMITER ;

-- -----------------------------------------------------
-- procedure listarGruposDisponibles
-- -----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `listarGruposDisponibles`()
BEGIN
	select idGrupoM, Detalle, Estado
	from grupom
	where estado = true
	order by detalle;
END$$

DELIMITER ;

-- -----------------------------------------------------
-- procedure listarMenuSistema
-- -----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `listarMenuSistema`()
BEGIN
	select * from menu_sistema;
END$$

DELIMITER ;

-- -----------------------------------------------------
-- procedure listarPedidosCaAdm
-- -----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `listarPedidosCaAdm`()
BEGIN
	select distinct p.idPedido, p.montoReserva,p.Codigo,
p.Fecha,p.HoraReservada,p.FechaValida, p.HoraValida,
p.NroPedido,p.HoraEntregada, p.Llevar, c.idcliente, c.nombres as cliente,
c.telefono, e.idestado, e.nombre as estado, concat(per.nombres,'
',per.appaterno) as empleado
	from pedido p
	left join cliente c on c.idCliente=p.idCliente
	left join estado e on e.idestado=p.idEstado
	left join empleado em on p.idempleado=em.idempleado
	left join usuario u on em.idusuario=u.idusuario
	left join permisos pe on pe.idUsuario=u.idUsuario
	left join Rol r on r.idRol = pe.idRol
	left join persona per on per.idpersona=u.idpersona
	where p.FechaValida = curdate() and e.idestado not like 6
```

```
      order by 14 asc, 7 asc, 8 asc;
END$$

DELIMITER ;

-- ----------------------------------------------------
-- procedure listarPedidosChMe
-- ----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `listarPedidosChMe`(in ides
int(11))
BEGIN
      select p.idPedido,p.Codigo, p.Fecha,p.HoraReservada,p.FechaValida,
p.HoraValida, p.NroPedido,p.HoraEntregada, p.Llevar, c.idcliente, c.nombres
as cliente, e.idestado, e.nombre as estado, concat(per.nombres,'
',per.appaterno) as empleado
      from pedido p
      left join cliente c on c.idCliente=p.idCliente
      left join estado e on e.idestado=p.idEstado
      left join empleado em on p.idempleado=em.idempleado
      left join usuario u on em.idusuario=u.idusuario
      left join persona per on per.idpersona=u.idpersona
      where p.FechaValida = curdate() and p.idestado=ides
      order by 7 asc, 12 asc;
END$$

DELIMITER ;

-- ----------------------------------------------------
-- procedure listarPedidosCliente
-- ----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `listarPedidosCliente`(in idcli
int(11))
BEGIN
      select p.idPedido, p.montoReserva,p.Codigo,
p.Fecha,p.HoraReservada,p.FechaValida, p.HoraValida, p.Llevar, e.idestado,
e.nombre as estado
      from pedido p
      left join cliente c on c.idCliente=p.idCliente
      left join estado e on e.idestado=p.idEstado
      left join usuario u on c.idusuario=u.idusuario
      where p.FechaValida = curdate() and p.idestado=1 and c.idCliente=idcli
      order by 7 asc, 9 asc;
END$$


-- ----------------------------------------------------
-- procedure listarPersonas
-- ----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `listarPersonas`()
BEGIN
      Select * from persona where not idPersona = 0 order by idpersona desc;
```

```
END$$

DELIMITER ;

-- ----------------------------------------------------
-- procedure listarProductos
-- ----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `listarProductos`()
BEGIN
	select p.idProducto, p.Nombre, p.Descripcion, p.Estado, p.Envase,
im.titulo, im.foto
	from producto p
	left join imagenproducto imp on p.idProducto = imp.idProducto
	left join imagen im on imp.idimagen = im.idimagen
	where imp.estado=true
	order by p.idProducto desc;
END$$

DELIMITER ;

-- ----------------------------------------------------
-- procedure listarProductosDisponibles
-- ----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `listarProductosDisponibles`()
BEGIN
	select p.idProducto, p.Nombre, p.Descripcion, p.Envase, pr.Monto,
gm.Detalle, im.titulo, im.foto
	from producto p
	left join precio pr on p.idProducto=pr.idProducto and pr.Estado=true
	left join dgrupom dg on pr.idProducto=dg.idProducto
	left join grupom gm on dg.idGrupoM=gm.idGrupoM
	left join imagenproducto imp on p.idProducto = imp.idProducto
	left join imagen im on imp.idimagen = im.idimagen
	where p.estado=true and imp.estado=true
	order by 6,2;
END$$

DELIMITER ;

-- ----------------------------------------------------
-- procedure listarProductosGeneralFalse
-- ----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `listarProductosGeneralFalse`()
BEGIN
	Select p.idProducto, p.Nombre, pr.Monto, gm.Detalle, im.Foto,
im.Titulo, dg.estado
		from producto p
		left join dgrupom dg on p.idProducto=dg.idProducto
		left join grupom gm on dg.idGrupoM=gm.idGrupoM
		left join imagenproducto imp on p.idProducto=imp.idProducto and
imp.Estado=true
```

```
            left join imagen im on imp.idimagen=im.idimagen
            left join precio pr on p.idProducto=pr.idProducto and
pr.Estado=true
              where p.Estado=true and dg.estado=false
      order by Detalle;
END$$

DELIMITER ;

-- -----------------------------------------------------
-- procedure listarProductosGeneralTrue
-- -----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `listarProductosGeneralTrue`()
begin
      Select p.idProducto, p.Nombre, pr.Monto, gm.Detalle, im.Foto,
im.Titulo, dg.estado
            from producto p
            left join dgrupom dg on p.idProducto=dg.idProducto
            left join grupom gm on dg.idGrupoM=gm.idGrupoM
            left join imagenproducto imp on p.idProducto=imp.idProducto and
imp.Estado=true
            left join imagen im on imp.idimagen=im.idimagen
            left join precio pr on p.idProducto=pr.idProducto and
pr.Estado=true
              where p.Estado=true and dg.estado=true
      order by Detalle;
END$$

DELIMITER ;

-- -----------------------------------------------------
-- procedure listarProductosXgrupo
-- -----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `listarProductosXgrupo`(in idg
int(11))
BEGIN
      select p.idProducto,p.Nombre,dg.Estado, im.Foto, im.Titulo, pr.monto
      from producto p
            left join dgrupom dg on dg.idProducto=p.idProducto
            left join imagenproducto imp on p.idProducto=imp.idProducto and
imp.Estado=true
            left join imagen im on imp.idimagen=im.idimagen
            left join precio pr on p.idProducto=pr.idProducto and
pr.Estado=true
      where dg.idGrupoM=idg;
END$$

DELIMITER ;

-- -----------------------------------------------------
-- procedure listarRoles
-- -----------------------------------------------------

DELIMITER $$
```

```
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `listarRoles`()
BEGIN
	Select idRol, Nombre, Estado from rol;
END$$

DELIMITER ;

-- -----------------------------------------------------
-- procedure listarRolesActivos
-- -----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `listarRolesActivos`()
BEGIN
	Select * from rol where Estado=true;
END$$

DELIMITER ;

-- -----------------------------------------------------
-- procedure listarTipoDoc
-- -----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `listarTipoDoc`()
BEGIN
	select idtipo_doc, detalle from tipo_doc;
END$$


-- -----------------------------------------------------
-- procedure logueo
-- -----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `logueo`(in log varchar(45),in
cl varchar(45))
begin
	SELECT idUsuario, Login, Clave, idPersona, Estado, FechaRegistro,
TituloFoto, Imagen, idEstablecimiento FROM usuario where Login = log and
Clave = cl and Estado = true;
END$$

DELIMITER ;

-- -----------------------------------------------------
-- procedure modificarDgrupoM
-- -----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `modificarDgrupoM`(in idp
int(11), in idgm int(11))
BEGIN
	update dgrupom set idGrupoM = idgm where idProducto = idp;
END$$
```

```
DELIMITER ;

-- -----------------------------------------------------
-- procedure modificarEstablecimiento
-- -----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `modificarEstablecimiento`(in
ide int(11), nom varchar(45), dir varchar(45), tel varchar(45), emai
varchar(45), in vFoto mediumblob, in vtitulo varchar(45))
BEGIN
	update Establecimiento set Nombre=nom, Direccion=dir, Telefono=tel,
Email=emai
	where idestablecimiento=ide;

	update logoEstablecimiento set Estado = false
	where idestablecimiento = ide;

	SET @fecha = Date(NOW());

	INSERT INTO imagen (Titulo, Estado, Foto, Fecha)
		VALUES (vtitulo, true, vFoto, @fecha);

	SET @idImagen=0;
	Select max(idimagen) into @idImagen from imagen;

	INSERT INTO logoEstablecimiento (Fecha, Estado, idestablecimiento,
idimagen)
	VALUES(@fecha,true,ide,@idImagen);
END$$

DELIMITER ;

-- -----------------------------------------------------
-- procedure modificarImagenCombo
-- -----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `modificarImagenCombo`(in idc
int(11), in tit varchar(45), img mediumblob)
BEGIN
	SET @fecha = Date(NOW());

	update imagencombo set Estado = false
	where idCombo = idc;

	INSERT INTO imagen (Titulo, Estado, Foto, Fecha)
		VALUES (tit, true, img, @fecha);

	SET @idImagen=0;
	Select max(idimagen) into @idImagen from imagen;

	INSERT INTO imagencombo (Fecha, Estado, idCombo, idimagen)
	VALUES(@fecha,true,idc,@idImagen);
END$$

DELIMITER ;
```

```
-- -----------------------------------------------------
-- procedure modificarPersona
-- -----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `modificarPersona`(in nom
varchar(25), pat varchar(25), mat varchar(25), telefono varchar(100), ema
varchar(45), dir varchar(100), in idp int(11))
BEGIN
	update persona set Nombres=nom, ApPaterno=pat, ApMaterno=mat,
Telefonos=telefono, Email=ema, Direccion=dir WHERE idPersona=idp;
END$$

DELIMITER ;

-- -----------------------------------------------------
-- procedure modificarProducto
-- -----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `modificarProducto`(in idp
int(11), in dsc varchar(250), in env varchar(45), in mon decimal(14,2), in
tit varchar(45), img mediumblob)
BEGIN
	update producto set Descripcion = dsc, Envase = env
	where idProducto = idp;

	update precio set Estado = false
	where idProducto = idp;

	SET @fecha = Date(NOW());

	INSERT INTO precio (Monto, Fecha, Estado, idProducto)
		VALUES(mon, @fecha, true, idp);

	update imagenproducto set Estado = false
	where idProducto = idp;

	INSERT INTO imagen (Titulo, Estado, Foto, Fecha)
		VALUES (tit, true, img, @fecha);

	SET @idImagen=0;
	Select max(idimagen) into @idImagen from imagen;

	INSERT INTO imagenproducto (Fecha, Estado, idproducto, idimagen)
	VALUES(@fecha,true,idp,@idImagen);
END$$

DELIMITER ;

-- -----------------------------------------------------
-- procedure modificarUsuario
-- -----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
```

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `modificarUsuario`(in fo
varchar(45), img mediumblob, in idu int(11))
BEGIN
	update usuario set TituloFoto=fo, Imagen=img WHERE idUsuario=idu;
END$$

DELIMITER ;

-- -----------------------------------------------------
-- procedure obtenerUsuario
-- -----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `obtenerUsuario`(in idu int(11))
BEGIN
	select distinct ifnull(r.Nombre,"Cliente") as Nombre,
			u.idUsuario, u.Login, u.Clave, u.idPersona,
			u.Estado, u.FechaRegistro, u.TituloFoto, u.Imagen,
e.Nombre as Establecimiento
	from usuario u
		left join permisos p ON p.idUsuario=u.idUsuario
		left join Rol r on r.idRol = p.idRol
		left join Establecimiento e on u.idEstablecimiento =
e.idEstablecimiento
	where u.idUsuario=idu
	order by idUsuario;
END$$

DELIMITER ;

-- -----------------------------------------------------
-- procedure verDetallePedido
-- -----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `verDetallePedido`(in idp
int(11))
BEGIN
	select detpe.idpedido, detpe.idItem, p.nombre, detpe.Cantidad,
detpe.precio, detpe.estado, im.Titulo, im.Foto
		from det_pedido detpe
			left join item i on i.idItem=detpe.iditem
			left join producto p on i.idProducto=p.idProducto
			left join imagenproducto imp on p.idProducto =
imp.idProducto
			left join imagen im on imp.idimagen = im.idimagen
	where imp.estado=true and detpe.idpedido=idp;
END$$

DELIMITER ;

-- -----------------------------------------------------
-- procedure verificarMenuPermiso
-- -----------------------------------------------------

DELIMITER $$
USE `dbmyburguer`$$
```

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `verificarMenuPermiso`(in idm
int(11), in idu int(11))
BEGIN
	Set @P=3;
	select distinct ifnull(idrol,3) into @P from permisos where
idusuario=idu;
	if @P=3 then
		begin
			select p.*
			from permisos p
			left join menu_sistema ms on p.idmenu = ms.idmenu
			where p.idmenu=idm and p.idUsuario=0 and p.asignado=true
and ms.Estado = true;
		end;
	else
		begin
			select p.*
			from permisos p
			left join menu_sistema ms on p.idmenu = ms.idmenu
			where p.idmenu=idm and p.idUsuario=idu and
p.asignado=true and ms.Estado = true;
		end;
	end if;
END$$

DELIMITER ;

SET SQL_MODE=@OLD_SQL_MODE;
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```

4.11.7 Diccionario de Datos

Tabla: persona

NOMBRE	TIPO	TAMAÑO	PK	FK	DESCRIPCION
idPersona	Entero	11	si		ID de persona
Nombres	Texto	45			Nombres de la persona.
ApPaterno	Texto	45			Apellido paterno de la persona
ApMaterno	Texto	45			Apellido materno de la persona
FechaNacimiento	Date				Fecha de nacimiento de la persona
Telefonos	Texto	45			Teléfonos de la persona
Email	Texto	45			Correo electrónico de la persona
Estado	Boolean o	1			Estado de la persona en el sistema
CI	Texto	45			Cedula de Identidad de la persona
Direccion	Texto	100			Dirección de residencia de la persona
LugarNacimiento	Texto	45			Lugar de Nacimiento de la persona
FechaRegistro	Date				Fecha de Registro de la persona en el sistema

Tabla 55. persona

Tabla: establecimiento

NOMBRE	TIPO	TAMAÑO	PK	FK	DESCRIPCION
idEstablecimiento	Entero	11	si		ID de Establecimiento
Nombre	Texto	45			Nombre del establecimiento.
Telefono	Texto	45			Teléfono del establecimiento
Email	Texto	45			Correo electrónico del establecimiento
Estado	Boolean o	1			Estado del establecimiento en el sistema
Nit	Texto	45			Nit del establecimiento
Direccion	Texto	100			Dirección del establecimiento

Tabla 56. establecimiento

Tabla: usuario

NOMBRE	TIPO	TAMAÑO	PK	FK	DESCRIPCION
idUsuario	Entero	11	si		ID de usuario
Login	Texto	45			Login o usuario de la persona para acceder al sistema
Clave	Texto	45			Contraseña del usuario para acceder al sistema
Estado	Boolean o	1			Estado del usuario en el sistema
TituloFoto	Texto	45			Nombre de la fotografía del usuario
Imagen	Blob				Imagen almacenada en byte[] del usuario
idPersona	Entero	11		Si	ID de Persona
idEstablecimiento	Entero	11		Si	ID de Establecimiento

FechaRegistro	Date				Fecha de Registro del usuario en el sistema

Tabla 57. usuario

Tabla: rol

NOMBRE	TIPO	TAMAÑO	PK	FK	DESCRIPCION
idRol	Entero	11	si		ID del rol
Nombre	Texto	45			Nombre del rol
Estado	Boolean o	1			Estado del rol en el sistema

Tabla 58. rol

Tabla: menú_sistema

NOMBRE	TIPO	TAMAÑO	PK	FK	DESCRIPCION
idmenu	Entero	11	si		ID del menú del sistema
Nombre	Texto	45			Nombre del menú del sistema
Estado	Boolean o	1			Estado del usuario en el sistema
Logo	Texto	45			Nombre del icono del menú
Link	Texto	45			Link del menú
MenuPadre	Entero	11		Si	ID padre del menú del sistema

Tabla 59. menu_sistema

Tabla: permisos

NOMBRE	TIPO	TAMAÑO	PK	FK	DESCRIPCION
idmenu	Entero	11		si	ID del menú del sistema
idUsuario	Entero	11		si	ID del usuario
asignado	Boolean o	1			Permiso asignado o denegado al usuario
idRol	Entero	11		si	ID del rol

Tabla 60. permisos

Tabla: producto

NOMBRE	TIPO	TAMAÑO	PK	FK	DESCRIPCION
idProducto	Entero	11	si		ID del producto
Nombre	Texto	45			Nombre del producto
Estado	Boolean o	1			Estado del producto en el sistema
Descripcion	Texto	250			Descripción de las características del producto.
Envase	Texto	45			Envase

Tabla 61. producto

Tabla: grupom

NOMBRE	TIPO	TAMAÑO	PK	FK	DESCRIPCION
idGrupoM	Entero	11	si		ID del grupo intercambiable
Estado	Boolean o	1			Estado del grupo en el sistema
Detalle	Texto	45			Detalle del grupo intercambiable.

Tabla 62. grupom

Tabla: dgrupom

NOMBRE	TIPO	TAMAÑO	PK	FK	DESCRIPCION
idGrupoM	Entero	11		si	ID del detalle del grupo intercambiable.
Estado	Boolean o	1			Estado del detalle del grupo en el sistema
idProducto	Entero	11		si	ID del detalle del grupo intercambiable.

Tabla 63. grupom

Tabla: precio

NOMBRE	TIPO	TAMAÑO	PK	FK	DESCRIPCION
idPrecio	Entero	11	si		ID del precio
Monto	Decimal	14,2			Valor del precio del producto
Estado	Booleano	1			Estado de la imagen en el sistema
idProducto	Entero	11			ID del Producto
Fecha	Date				Fecha de registro del precio

Tabla 64. precio

Tabla: imagen

NOMBRE	TIPO	TAMAÑO	PK	FK	DESCRIPCION
idimagen	Entero	11	si		ID de la imagen
Titulo	Texto	45			Titulo o nombre de la imagen
Estado	Boolean o	1			Estado de la imagen en el sistema
Foto	Blob				Imagen almacenada en byte[]
Fecha	Date				Fecha de registro de la imagen

Tabla 65. imagen

Tabla: imagenProducto

NOMBRE	TIPO	TAMAÑO	PK	FK	DESCRIPCION
idimagen	Entero	11		si	ID de la imagen del producto
Estado	Boolean o	1			Estado de la imagen del producto en el sistema
idProducto	Entero	11		si	ID del producto.
Fecha	Date				Fecha de registro de la imagen del producto

Tabla 66. imagenProducto

Tabla: logoEstablecimiento

NOMBRE	TIPO	TAMAÑO	PK	FK	DESCRIPCION
idimagen	Entero	11		si	ID de la imagen del establecimiento
Estado	Boolean o	1			Estado de la imagen del establecimiento en el sistema
idEstablecimient o	Entero	11		si	ID del establecimiento.
Fecha	Date				Fecha de registro de la imagen del establecimiento

Tabla 67. logoEstablecimiento

Tabla: combo

NOMBRE	TIPO	TAMAÑO	PK	FK	DESCRIPCION
idCombo	Entero	11	si		ID del combo
Fecha	Date				Fecha de registro del combo en el sistema
Vigente	Boolean o	1			Estado del combo en el sistema
Descripcion	Texto	200			Descripción o nombre del combo

Tabla 68. combo

Tabla: imagenCombo

NOMBRE	TIPO	TAMAÑO	PK	FK	DESCRIPCION
idimagen	Entero	11		si	ID de la imagen del combo
Estado	Boolean o	1			Estado de la imagen del combo en el sistema
idCombo	Entero	11		si	ID del combo.
Fecha	Date				Fecha de registro de la imagen del combo

Tabla 69. logoCombo

Tabla: descuento

NOMBRE	TIPO	TAMAÑO	PK	FK	DESCRIPCION
idDescuento	Entero	11	si		ID del descuento
descuento	Decimal	14,2			Valor del precio del descuento
idCombo	Entero	11			ID del Combo

Tabla 70. descuento

Tabla: item

NOMBRE	TIPO	TAMAÑO	PK	FK	DESCRIPCION
idItem	Entero	11	si		ID del ítem
Estado	Boolean o	1			Estado del ítem en el sistema
idCombo	Entero	11		si	ID del combo.
Fecha	Date				Fecha de registro del ítem
idProducto	Entero	11		si	ID del producto

Tabla 71. item

Tabla: empleado

NOMBRE	TIPO	TAMAÑO	PK	FK	DESCRIPCION
idEmpleado	Entero	11		si	ID del empleado
FechaRegistro	Date				Fecha de Registro del empleado en el sistema
Estado	Boolean o	1			Estado del empleado en el sistema
idUsuario	Entero	11		si	ID del usuario

Tabla 72. empleado

Tabla: cliente

NOMBRE	TIPO	TAMAÑO	PK	FK	DESCRIPCION
idCliente	Entero	11	si		ID del Cliente
Nombres	Texto	100			Nombre del cliente.
Telefono	Texto	45			Teléfono del cliente
Estado	Boolean o	1			Estado del cliente en el sistema
Nit	Texto	45			Nit del cliente
idUsuario	Entero	11		si	ID del usuario

Tabla 73. cliente

Tabla: estado

NOMBRE	TIPO	TAMAÑO	PK	FK	DESCRIPCION
idestado	Entero	11	si		ID del Estado del pedido
Nombre	Texto	45			Nombre del estado.

Tabla 74. estado

Tabla: pedido

NOMBRE	TIPO	TAMAÑO	PK	FK	DESCRIPCION
idPedido	Entero	11	si		ID del pedido
Codigo	Texto	45			Código del pedido.
HoraReservada	Time				Hora de reserva del pedido
FechaValida	Date				Fecha de entrega del pedido
HoraValida	Time				Hora de entrega del pedido
NroPedido	Texto	45			Nro de pedido que inicia en 1 cada día
Llevar	Booleano	1			Valor que indica si el pedido es para llevar o consumir en local.
idCliente	Entero	11			ID del cliente
idEstado	Entero	11			ID del estado del pedido

idEmpleado	Entero	11			ID del empleado
Fecha	Date				Fecha de Registro del pedido en el sistema
HoraEntregada	Time				Hora entregada del pedido
horaOrdenar	Time				Tiempo de anticipación para cambiar el estado del pedido de reservado a ordenado.
montoReserva	Float				Monto total de la reserva

Tabla 75. pedido

Tabla: det_pedido

NOMBRE	TIPO	TAMAÑO	PK	FK	DESCRIPCION
idPedido	Entero	11		si	ID del pedido
Estado	Booleano	1			Estado del ítem en el pedido registrado en el sistema
idItem	Entero	11		si	ID del ítem.
Cantidad	Entero				Cantidad de ítems.
Precio	Decimal	14,2			Valor del precio del ítem.
Descuento	Decimal	14,2			Valor del descuento del combo

Tabla 76. det_pedido

Tabla: tipo_doc

NOMBRE	TIPO	TAMAÑO	PK	FK	DESCRIPCION
Idtipo_doc	Entero	11	si		ID del tipo de documento
Detalle	Texto	45			Detalle o nombre del tipo de documento.

Tabla 77. tipo_doc

Tabla: documento

NOMBRE	TIPO	TAMAÑO	PK	FK	DESCRIPCION
idDocumento	Entero	11	si		ID del documento.
Codigo	Texto	45			Código del pedido.
Detalle	Texto	100			Detalle o glosa del documento.
Monto	Decimal	14,2			Valor del precio total del pedido entregado.
Nombre	Texto	100			Nombre para el cual será emitido el documento.
CI	Texto	45			Cedula de identidad o nit para el cual será emitido el documento.
idCliente	Entero	11			ID del cliente
idPedido	Entero	11			ID del pedido
idtipo_doc	Entero	11			ID del tipo de documento
Fecha	Date				Fecha de registro del documento

Tabla 78. documento

Tabla: det_documento

NOMBRE	TIPO	TAMAÑO	PK	FK	DESCRIPCION
idDocumento	Entero	11		si	ID del documento
subTotal	Decimal	14,2			Subtotal del ítem.
idItem	Entero	11		si	ID del ítem.
cantidad	Entero				Cantidad de ítems.
monto	Decimal	14,2			Valor del precio del ítem.
descuento	Decimal	14,2			Valor del descuento del combo

Tabla 79. det_documento

4.12 Modelo de Despliegue

4.12.1 Diagrama de Desplazamiento

Introducción

El diagrama de despliegue es el que representa o muestra la parte física de la arquitectura del sistema que se está modelando.

Propósito

Modelar la arquitectura del sistema.

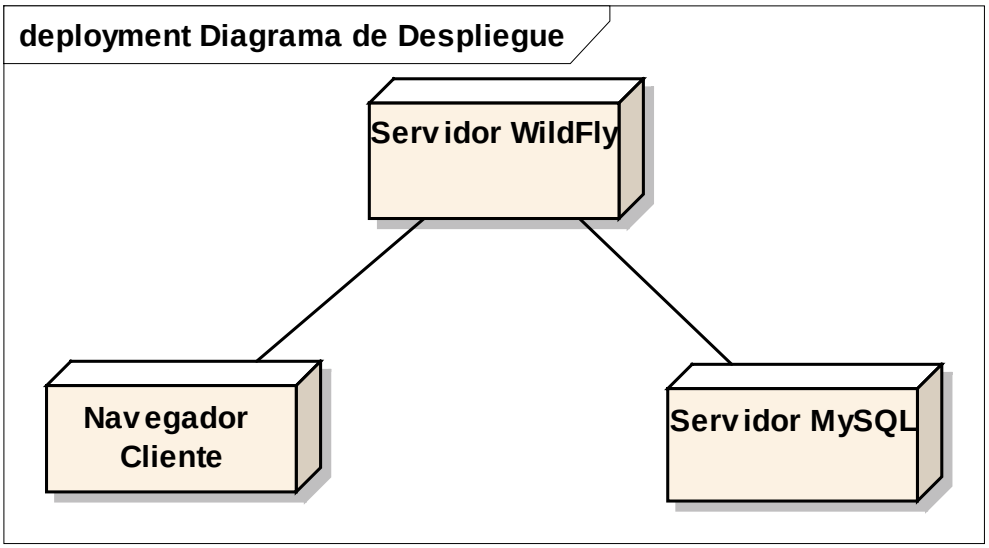


Figura 142. Diagrama de Despliegue

4.13 Casos de prueba del sistema

La prueba del sistema es un elemento crítico para la garantía de calidad del sistema y representa una revisión final de las especificaciones del diseño y codificación.

Definición:

La prueba es un proceso de ejecución de un programa con la intensión de descubrir error.

Un buen caso de prueba es aquel que tiene una alta probabilidad de mostrar un error no descubierto hasta entonces. Todos los productos del sistema probados de dos formas (1) conociendo la función específica para la que fue diseñado el producto, se puede llevar a cabo pruebas que demuestren que cada función es completamente operativa denominada Prueba de caja negra y (2) conociendo el funcionamiento del producto se puede realizar pruebas que todas las piezas encajen, o sea que la especificación interna se ajuste a las especificaciones y que todos los componentes internos se han comprobado de forma adecuada, esta prueba se denomina a Prueba de caja Blanca.

Las pruebas de caja negra permiten descubrir de forma inmediata una clase de errores (por ejemplo, procedimiento incorrecto de todos los datos de caracteres) que de otro modo. La partición equivalente se dirige a la definición de casos de prueba que descubran clases de errores, reduciendo el número total de casos de prueba que hay que desarrollar.

4.13.1 Pruebas de caja negra:

La prueba de caja negra permite derivar conjuntos de condiciones de entrada que ejerciten completamente todos los requerimientos funcionales de un programa. Los errores que se pretenden detectar mediante las pruebas de caja negra son:

- ✓ Funciones incorrecta o ausente

- ✓ Errores de interfaz.
- ✓ Errores en la estructura de datos
- ✓ Error de Rendimiento y error de inicialización o terminación.

Ingresar al sistema

Condiciones de entrada	Clases válidas	Entradas inválidas
Usuario	1. Cualquier caracter	2.
Longitud de Usuario	3. 7 < valor < 45	4. Valor < 7 5. Valor > 10
Contraseña	6. Cualquier caracter	7.
Longitud de Contraseña	8. 1 < valor < 45	9. En blanco 10. Valor > 45

Casos de prueba: Clase válida

Caso	Login	Contraseña	Clases
CP1	trico*154	761sarcor.	1,3,6,8

Casos de prueba: Clases inválidas

Caso	Login	Contraseña	Clases
CP1			9

Registrar Persona

Condiciones de entrada	Clases válidas	Entradas inválidas
Nombres	1. Letras	2. Cualquier carácter
Longitud de Nombres	3. 1 < valor < 45	4. En blanco 5. Valor > 45
Apellido Paterno	6. Letras	7. Cualquier carácter
Longitud de Paterno	8. 1 < valor < 45	9. En blanco 10. Valor > 45
Apellido Materno	11. Letras	12. En blanco 13. Valor > 45
Longitud de Materno	14. 1 < valor < 45	15. En blanco

		16. Valor > 45
Cedula de Identidad	17. Numero	18. Cualquier caracter
Longitud de CI	19. 8 <= valor < 13	20. Valor < 8 21. Valor > 13
Telefono	22. Numero	23. Cualquier caracter
Longitud Telefono	24. 6 < valor < 9	25. En blanco 26. Valor < 7 27. Valor > 8
Barrio	28. Alfanumérico	29.
Longitud Barrio	30. 0 < valor < 45	31. En blanco 32. Valor > 45
Calle	33. Alfanumérico	34.
Longitud Calle	35. 0 < valor < 45	36. En blanco 37. Valor > 45
Número de Casa	38. Número	39. Cualquier caracter
Longitud N° Casa	40. 0 < valor < 6	41. En blanco 42. Valor > 5
eMail	43. Alfanumérico . _ - @	44. Simbolos
Longitud eMail	45. 0 < valor < 45	46. En blanco 47. Valor > 45

Casos de prueba: Clases válidas

Caso	Nombres	Ap. Paterno	Ap. materno	CI	Telefono	Barrio	Calle	N°	Email	Clases
CP1	Marcia Milenka	Andrade	Llanos	7177285	79254690	Avaroa	Litoral	258	marcia.andrade.llanos@gmail.com	1, 3, 6, 8, 11, 14, 17, 19, 22, 24, 28, 30, 33, 35, 38, 40, 43, 45

Casos de prueba: Clases inválidas

Caso	Nombres	Ap. Paterno	Ap. materno	CI	Telefono	Barrio	Calle	N°	Email	Clases
CP1	123			fasd7177285	79fsd254690fb*			25<gr8-		2

Modificar Persona

Condiciones de entrada	Clases válidas	Entradas inválidas
Nombres	1. Letras	2. Cualquier carácter
Longitud de Nombres	3. 1 < valor < 45	4. En blanco 5. Valor > 45
Apellido Paterno	6. Letras	7. Cualquier carácter
Longitud de Paterno	8. 1 < valor < 45	9. En blanco 10. Valor > 45
Apellido Materno	11. Letras	12. En blanco 13. Valor > 45
Longitud de Materno	14. 1 < valor < 45	15. En blanco 16. Valor > 45
Telefono	17. Numero	18. Cualquier caracter
Longitud Telefono	19. 6 < valor < 9	20. En blanco 21. Valor < 7 22. Valor > 8
Dirección	23. Alfanumérico # ,	24.
Longitud Dirección	25. 0 < valor < 100	26. En blanco 27. Valor > 45
eMail	28. Alfanumérico . _ - @	29. Simbolos
Longitud eMail	30. 0 < valor < 45	31. En blanco 32. Valor > 45

Casos de prueba: Clases válidas

Caso	Nombres	Ap. Paterno	Ap. materno	Telefono	Barrio	Email	Clases
CP1	Marcia Milenka	Andrade	Llanos	79254690	Avaroa	marcia.andrade.llanos@gmail.com	1, 3, 6, 8, 11, 14, 17, 19, 22, 24, 28, 30, 33, 35, 38, 40, 43, 45

Casos de prueba: Clases inválidas

Caso	Nombres	Ap. Paterno	Ap. materno	Telefono	Barrio	Email	Clases
CP1	123			79fsd254690fb*			2

Registrar Usuario

Condiciones de entrada	Clases válidas	Entradas inválidas
Usuario	1. Cualquier caracter	2.
Longitud de Usuario	3. 7 < valor < 45	4. Valor < 7 5. Valor > 10
Contraseña	6. Cualquier carácter, tiene que ser igual a Confirmar Contraseña	7.
Longitud de Contraseña	8. 1 < valor < 45	9. En blanco 10. Valor > 45
Confirmar Contraseña	11. Cualquier caracter, tiene que ser igual a Contraseña	12.
Longitud de Confirmar Contraseña	13. 1 < valor < 45	14. En blanco 15. Valor > 45

Casos de prueba: Clase válida

Caso	Login	Clave	Confirmar Contraseña	Clases
CP1	trico*154	761sarcor.	761sarcor.	1,3,6,8,11,13

Casos de prueba: Clases inválidas

Caso	Login	Clave	Confirmar Contraseña	Clases
CP1				9

Cambiar Contraseña

Condiciones de entrada	Clases válidas	Entradas inválidas
Contraseña	1. Cualquier carácter 2. Igual a Confirmar Contraseña	3. 4. Diferente a Confirmar Contraseña
Longitud de Contraseña	5. 1 < valor < 45	6. En blanco 7. Valor > 45
Confirmar Contraseña	8. Cualquier carácter 9. Igual a Contraseña	10. 11. Diferente a Contraseña

Longitud de Confirmar Contraseña	12. 1 < valor < 45	13. En blanco 14. Valor > 45

Casos de prueba: Clase válida

Caso	Login	Clave	Confirmar Contraseña	Clases
CP1	trico*154	761sarcor.	761sarcor.	1,3,6,8

Casos de prueba: Clases inválidas

Caso	Login	Clave	Confirmar Contraseña	Clases
CP1				9
CP2				

Registrar Rol

Condiciones de entrada	Clases válidas	Entradas inválidas
Nombre	1. Letras	2. Cualquier carácter
Longitud de Nombre	3. 1 < valor < 45	4. En blanco 5. Valor > 45

Casos de prueba: Clases válidas

Caso	Nombre	Clases
CP1	Rol	1, 3

Casos de prueba: Clases inválidas

Caso	Nombre	Clases
CP1	123rol	2
CP2		4

Registrar Producto

Condiciones de entrada	Clases válidas	Entradas inválidas
Nombre	1. Letras	2. Cualquier carácter
Longitud de Nombres	3. 1 < valor < 45	4. En blanco 5. Valor > 45
Descripción	6. Cualquier caracter	7.
Longitud de Descripcion	8. 0 =< valor =< 250	9. Valor > 250
Envase	10. Letras	11. Cualquier caracter 12. Valor > 45
Longitud de Envase	13. 1 < valor < 45	14. En blanco 15. Valor > 45
Precio	16. Numero decimal	17. Cualquier caracter
Longitud Precio	18. 14,2	19. En blanco 20. Valor > 14,2

Casos de prueba: Clases válidas

Caso	Nombre	Descripcion	Envase	Precio	Clases
CP1	Manzana	Manzana roja 1° calidad	Ninguno	2.50	1, 3, 6, 8, 10, 13, 16, 18

Casos de prueba: Clases inválidas

Caso	Nombre	Descripcion	Envase	Precio	Clases
CP1	6776#$%”		87652:_$;#	79fsd254690fb*	2,11,17
CP2					4,14,19

Editar Producto

Condiciones de entrada	Clases válidas	Entradas inválidas
Descripción	1. Cualquier caracter	2.
Longitud de Descripcion	3. 0 =< valor =< 250	4. Valor > 250

Envase	5. Letras	6. Cualquier caracter 7. Valor > 45
Longitud de Envase	8. 1 < valor < 45	9. En blanco 10. Valor > 45
Precio	11. Numero decimal	12. Cualquier caracter
Longitud Precio	13. 14,2	14. En blanco 15. Valor > 14,2

Casos de prueba: Clases válidas

Caso	Descripcion	Envase	Precio	Clases
CP1	Manzana roja 1° calidad	Ninguno	2.50	1, 3, 5, 8, 11, 13

Casos de prueba: Clases inválidas

Caso	Descripcion	Envase	Precio	Clases
CP1		87652:_$;#	79fsd254690fb*	6,12
CP2				9,15

Registrar Grupo Intercambiable

Condiciones de entrada	Clases válidas	Entradas inválidas
Detalle	1. Letras	2. Cualquier carácter
Longitud de Detalle	3. 1 < valor < 45	4. En blanco 5. Valor > 45

Casos de prueba: Clases válidas

Caso	Detalle	Clases
CP1	Grupo	1, 3

Casos de prueba: Clases inválidas

Caso	Detalle	Clases
CP1	$&”$#Grupo	2
CP2		4

Registrar Combo

Condiciones de entrada	Clases válidas	Entradas inválidas
Titulo	1. Cualquier caracter	2. En blanco
Longitud de Titulo	3. 0 =< valor =< 200	4. Valor > 200
Descuento	5. Numero decimal	6. Cualquier caracter
Longitud Descuento	7. 14,2	8. En blanco 9. Valor > 14,2

Casos de prueba: Clases válidas

Caso	Titulo	Descuento	Clases
CP1	Combo 2342 =&$	2.50	1, 3, 5, 7

Casos de prueba: Clases inválidas

Caso	Titulo	Descuento	Clases
CP1		79fsd254690fb*	2,6
CP2			2,8

Actualizar Combo

Condiciones de entrada	Clases válidas	Entradas inválidas
Titulo	1. Cualquier caracter	2. En blanco
Longitud de Titulo	3. 0 =< valor =< 200	4. Valor > 200
Descuento	5. Numero decimal	6. Cualquier caracter
Longitud Descuento	7. 14,2	8. En blanco 9. Valor > 14,2

Casos de prueba: Clases válidas

Caso	Titulo	Descuento	Clases
CP1	Combo 2342 =&$	2.50	1, 3, 5, 7

Casos de prueba: Clases inválidas

Caso	Titulo	Descuento	Clases
CP1		79fsd254690fb*	2,6
CP2			2,8

Conclusión de las Pruebas

Una vez realizados los casos de prueba pudimos comprobar que el sistema tenía algunos errores como ser los datos de entrada, gracias a las pruebas pudimos eliminar estos errores y así garantizar de calidad del sistema.

Glosario

Introducción

Este documento recoge todos y cada uno de los términos manejados a lo largo de todo el proyecto de desarrollo de un sistema de gestión. Se trata de un diccionario informal de datos y definiciones de la nomenclatura que se maneja, de tal modo que se crea un estándar para todo el proyecto.

Propósito

El propósito de este glosario es definir con exactitud y sin ambigüedad la tecnología manejada en el proyecto de desarrollo de un sistema de gestión. También sirve como guía de consulta para la clarificación de los puntos conflictivos o poco esclarecedores del proyecto.

Alcance

El alcance del presente documento se extiende a todos los subsistemas definidos para la empresa.

Términos

Glosario de términos

Acceso

Ingreso al sistema por medio de un nombre de una cuenta y contraseña

Desconectar

Cierra la sesión iniciada por el usuario.

Iniciar sesión

Habilitar al usuario para trabajar en Sistema, de acuerdo a los roles que le corresponden.

Proceso

Un proceso es un orden específico de actividades de trabajo, que se realizan en el tiempo, en lugares específicos con un principio, un fin y entradas y salidas claramente definidas. Es decir, una estructura cohesionada y coordinada adecuadamente para la acción

Proyecto

Un proyecto es un esfuerzo temporal comprometido para crear un producto o servicio único. Temporal significa que tiene un inicio y fin, y único que el servicio o producto es diferente e identificable de otros similares.

Seguridad

Es una característica de cualquier sistema (informático o no) que nos indica que ese sistema está libre de todo peligro, daño o riesgo, y que es, en cierta manera, infalible.

Usabilidad

La usabilidad es el rango en el cual un producto puede ser usado por unos usuarios específicos para alcanzar ciertas metas especificadas con efectividad, eficiencia y satisfacción en un contexto de uso especificado.

4.14 Medios de Verificación del Componente I

Anexo 1

Carta de respaldo de la gerencia "My Burguer" respecto a la conformidad y aceptación del sistema.

myBurguer

AVENIDA FONT, B/ JUAN XXIII – CELULAR 78238369

Tarija, 25 de Enero de 2016

Señores:

Docentes de Taller III

Presente.-

REF. CONFORMIDAD DE CAPACITACIÓN.

De mi consideración

Mediante la presente Certifico en mi condición de Gerente de "MyBurguer", que se llevó a cabo el Plan de Capacitación expuesto por la Univ. Marcia Milenka Andrade Llanos referente al proyecto **"Mejoramiento en el manejo de la información y tiempo de atención al cliente mediante la gestión de reservas de pedidos en línea"**, el mismo que fue desarrollado cumpliendo con los objetivos que fueron propuestos en el plan.

Sin otro particular me despido cordialmente deseándoles éxito en su trabajo

Luis Delfín Briceño Gordy
CI: 5049145 Tja.
Gerente "MyBurguer"

4.15 Componente II

4.15.1 Marco Teórico

4.15.2 Antecedentes

De acuerdo a la vida actual del mundo empresarial el termino capacitación y sistemas de información están cambiando la forma de trabajo de las empresas, los sistemas de información ayudan a acelerar procesos por lo tanto; las organizaciones que los implantan logran ventajas competitivas al adoptarlos en sus funciones. La capacitación se refiere a los métodos que se usan para proporcionar a las personas dentro de la empresa las habilidades que necesitan para realizar su trabajo, esta abarca desde pequeños cursos sobre terminología hasta cursos que le permitan al usuario entender el funcionamiento del sistema nuevo, ya sea teórico o a base de prácticas o mejor aún, combinando los dos.

Este es un proceso que lleva a la mejora continua y con esto a implantar nuevas formas de trabajo, como en este caso un sistema que será automatizado viene a agilizar los procesos y llevar a la empresa que lo adopte a generar un valor agregado y contribuir a la mejora continua por medio de la implantación de sistemas y capacitación a los usuarios.

El desarrollo de los cursos de capacitación es de vital importancia para dar respuestas a las necesidades que se perciben dentro de la empresa y así actualizar el conocimiento de las personas involucradas en la capacitación de tal forma que se tenga un personal productivo y eficiente que responda a los nuevos métodos de trabajo.

La capacitación en los distintos niveles constituye un beneficio en las mejores inversiones en cuanto a los recursos humanos y será un beneficio que conduzca a dar mayor facilidad para el usuario en tener automatizado la mayor parte de los registros; crea una mejor organización al tener un sistema que ayudara el trabajo que se desempeña, dándose respuestas positivas y rápidas para cualquier petición de la empresa.

Puede afirmarse que la capacitación forma parte del poder aprender nuevos conocimientos acerca de algún tema, en relación constituye y se considera como una parte importante del desarrollo del individuo, porque permite su inserción en la sociedad de manera más participativa mediante el conocimiento de herramientas o medios para informarse.

La capacitación beneficiara al personal de la empresa de forma eficiente en la toma de decisiones contribuyendo positivamente en el mejoramiento de las aptitudes comunitarias, permitir el logro de las metas individuales progresando en distintos campos tanto como en lo laboral como en la personal. Hoy en día para toda empresa los recursos humanos son una parte esencial para su buen funcionamiento ya que el personal de una organización es uno de los recursos más importantes, que por medio de ellos se puede alcanzar los objetivos establecidos como también se puede llegar a un mejor desarrollo de la empresa para alcanzar un alto nivel de competitividad.

La capacitación es una herramienta fundamental en los recursos humanos que ofrece la posibilidad de mejorar la eficiencia del trabajo dentro de la empresa, permitiendo a su vez que la misma se adapte a las nuevas circunstancias que se presentan tanto dentro como fuera de la empresa. Proporcionando a los empleados la oportunidad de adquirir mayores conocimientos, de esta manera también resulta siendo una herramienta motivadora.

La capacitación implica la aprobación de los nuevos conocimientos, carece de profundidad desde las perspectivas del análisis de contenido y el pensamiento crítico, porque el foco de la atención se ubica en la adquisición de habilidades y conocimientos prácticamente mecánicos.

Todo aprendizaje requiere de estrategias que definen como un proceso de toma de decisiones conscientes e intencionales en los cuales los capacitados eligen, recuperar de una manera coordinada, los conocimientos que necesita para

completar una demanda u objetivo, dependiendo de las características de la situación que forman parte del proceso de aprendizaje.

El auto aprendizaje es la forma de aprender principalmente por uno mismo, buscando información, haciendo prácticas o experimentos. En cuanto al aprendizaje colaborativo se puede decir que es un conjunto de métodos de instrucción o entrenamiento para uso en grupos, así como de estrategias para propiciar el desarrollo de habilidades mixtas, aprendizaje y desarrollo personal y social. Cada miembro del grupo es responsabilidad de su propio aprendizaje, así como el de los restantes miembros del grupo.

Con la sociabilización se pretende generar flujos de comunicación y participación de la sociedad en el Proyecto.

De esta forma el Proyecto se fortalece y se garantiza un tiempo de vida prolongado.

4.15.3 Introducción

Para la capacitación del personal se realizó una explicación detallada del funcionamiento del sistema, esta se realiza junto al manual que tiene el sistema, el manual de usuario, con lo que se resolvió las dudas en cuanto al funcionamiento del sistema.

Con la sociabilización se llega a más clientes gracias a las publicaciones de la presentación del sistema.

4.15.4 Objetivo General

- Dar a conocer el manejo del sistema, e incrementar conocimientos y habilidades de las personas acerca de los conocimientos en la actualización y conocimiento en nuestra sociedad en general.

El plan de capacitación que se presento es una estrategia efectiva basada en el programa que se tendrá para poder realizar el proceso de aprendizaje.

4.15.5 Objetivos Específicos

- Lograr que en la capacitación se aprenda a trabajar con el sistema desarrollado y de esta manera mejorar el proceso enseñanza aprendizaje.
- Lograr que se perfeccionen el personal en el desempeño de sus puestos tanto actuales como futuros.
- Mantener al personal permanentemente actualizados frente a los cambios científicos y tecnológicos que se generen proporcionándoles información sobre la aplicación de nueva tecnología.
- Lograr cambios en su comportamiento con el propósito de mejorar las relaciones interpersonales entre todos los miembros de la empresa.
- Hacer conocer a la Población sobre la existencia del proyecto.
- Fomentar la utilización y participación de la sociedad en el Sitio Web diseñado.

4.15.6 Misión

El programa de capacitación tiene la misión de apoyar y dar a conocer el funcionamiento del sistema que se desarrolló para la empresa “MyBurguer” y la utilización efectiva en conocimiento como elemento clave del desarrollo.

4.15.7 Metas

La meta del programa es integrar efectivamente el uso y el manejo del sistema dentro de la empresa para que los encargados sean capaces de poder adaptarse al manejo de toda la información procesada en el sistema.

4.15.8 Alcances

- ✓ El listado de personas participantes de la capacitación.
- ✓ Certificados otorgados al personal participante de dicha capacitación.
- ✓ Enseñarles el manejo del sistema.

4.15.9 Justificación de la capacitación

La capacitación se justifica plenamente por la falta de conocimiento y experiencia en el manejo de algún sistema informático, para el proceso de información que se manipula para la obtención de datos que se necesitan saber, también para dar a conocer los beneficios que nos brindan los sistemas de gestión computarizados.

4.15.10 Medios a Utilizar

Materiales de capacitación:

- ✓ Materiales impresos: guías del curso, manuales de usuario
- ✓ Diapositivas de presentación hechas en PowerPoint, televisor de la empresa cliente.
- ✓ Otros materiales: marcadores, cuaderno, lápiz

4.15.11 Estructura del curso de capacitación

Fecha a Realizar	Módulos de Aprendizaje del Curso	Horas C/Módulo	Nº de Sesiones	Dirigido a
11/01/16	Módulo 1. Introducción a los Conceptos de Sistemas Web	20 min.	1	Gerente
	Módulo 2. Presentación del Sistema multiplataforma para el manejo de información en atención al cliente y gestión de reservas en línea	50 min.	1	
	Totales	**1 Hora, 10 min**	**2 Sesiones**	

Fecha a Realizar	Módulos de Aprendizaje del Curso	Horas C/Modulo	Nº de Sesiones	Dirigido A
12/01/16	Módulo 1. Introducción a los Conceptos de Sistemas Web	20 min.	1	Recepcionista
	Módulo 2. Presentación del Sistema multiplataforma para el manejo de información en atención al cliente y gestión de reservas en línea.	40 min.	1	
	Totales	**1 Hora**	**2 Sesiones**	

Fecha a Realizar	Módulos de Aprendizaje del Curso	Horas C/Modulo	Nº de Sesiones	Dirigido A
13/01/16	Módulo 1. Introducción a los Conceptos de Sistemas Web.	20 min.	1	Personal de My Burguer
	Módulo 2. Presentación del Sistema multiplataforma para el manejo de información en atención al cliente y gestión de reservas en línea	40 min.	1	
	Totales	**1 Hora**	**2 Sesiones**	

4.15.11.1 Contenido del curso de Capacitación

Nº	Actividad	Responsable	Hora inicio	Hora fin	Duración	Medios a utilizar
1	Presentación y Bienvenida al Curso de Capacitación.	Gerente de "MyBurguer"	06:00 pm	06:10 pm	10 min.	
2	Explicación de la Importancia de la Capacitación.	Jefe del Proyecto	06:10 pm	06:15 pm	5 min.	Diapositivas Televisor Computador
3	Introducción a los Conceptos de Sistemas Web	Jefe del Proyecto	06:15 pm	06:40 pm	25 min.	Diapositivas Televisor Computador
4	Explicación de la Importancia del Sistema Desarrollado	Jefe del Proyecto	06:50 pm	07:00 pm	10 min.	Diapositivas Televisor Computador

5	Presentación del Sistema	Jefe del Proyecto	07:00 pm	07:40 pm	40 min.	Sistema Desarrollado. Televisor Computador
6	Consultas	Gerente, Personal de la empresa	07:40 pm	08:00 pm	20 min.	Sistema Desarrollado. Televisor Computador
7	Entrega de Encuestas Post Capacitación	Jefe del Proyecto	08:00 pm	08:02 pm	2 min.	Encuestas
9	Llenado de las encuestas	Gerente, Personal de la empresa	08:02 pm	08:12 pm	10 min.	Encuestas
10	Recogida de las escuestas	Jefe del Proyecto	08:12 pm	08:15 pm	3 min.	Encuestas
11	Consultas Finales	Gerente, Personal de la empresa	08:15 pm	08:25 pm	10 min.	Sistema Desarrollado. Televisor Computador Diapositivas
12	Clausura del Curso de Capacitación	Gerente de “MyBurguer” y Jefe del Proyecto	08:25 pm	08:30 pm	5 min.	

4.16 Medios de verificación del componente II

Anexo 1: Carta de conformidad y respaldo con el desarrollo de la capacitación por parte de la Gerencia de "My Burguer".

myBurguer

AVENIDA FONT, B/ JUAN XXIII – CELULAR 78238369

Tarija, 25 de Enero de 2016

Señores:

Docentes de Taller III

Presente.-

REF. CERTIFICADO DE CONFORMIDAD Y ACEPTACIÓN.

De mi consideración

En mi condición de Gerente de "MyBurguer", certifico y acepto el trabajo realizadopor la Univ. Marcia Milenka Andrade Llanos, y que proporcionándole toda la información de las áreas contempladas en el proyecto titulado **"Mejoramiento en el manejo de la información y tiempo de atención al cliente mediante la gestión de reservas de pedidos en línea"**, fue desarrollado en su mayor parte cumpliendo los objetivos que fueron propuestos en el inicio del mismo, denotando un gran interés y profesionalismo de su parte.

Es todo cuanto puedo certificar en honor a la verdad, pudiendo la interesada hacer uso de la presente, en lo que estimaré conveniente.

Sin otro particular, me despido.

.....

Luis Delfín Briceño Gordy
CI: 5049145 Tja.
Gerente "MyBurguer"

Anexo 2: Fotografías de la capacitación

Anexo 3: Certificados de Capacitación

UNIVERSIDAD AUTÓNOMA "JUAN MISAEL SARACHO"
FACULTAD DE CIENCIAS Y TECNOLOGÍA
DEPARTAMENTO DE INFORMÁTICA Y SISTEMAS

OTORGA EL PRESENTE CERTIFICADO DE CAPACITACIÓN

A: LUIS DELFÍN BRICEÑO GORDY

Por haber participado en el curso de capacitación de manejo del sistema **Mejoramiento en el manejo de la información y tiempo de atención al cliente mediante la gestión de reservas de pedidos en línea en "My Burguer"**, impartidos en el edificio de "My Burguer", los días lunes 11, martes 12 y miércoles 13 de enero 2016.

Msc. Lic. Octavio D. Aguilar M.
DIRECTOR DPTO. INFORMATICA Y SISTEMAS

Univ. Marcia Milenka Andrade Llanos
EXPOSITORA

UNIVERSIDAD AUTÓNOMA "JUAN MISAEL SARACHO"
FACULTAD DE CIENCIAS Y TECNOLOGÍA
DEPARTAMENTO DE INFORMÁTICA Y SISTEMAS

OTORGA EL PRESENTE CERTIFICADO DE CAPACITACIÓN

A: MARÍA CRISTINA CÁRDENAS MENDOZA

Por haber participado en el curso de capacitación de manejo del sistema **Mejoramiento en el manejo de la información y tiempo de atención al cliente mediante la gestión de reservas de pedidos en línea en "My Burguer"**, impartidos en el edificio de "My Burguer", los días lunes 11, martes 12 y miércoles 13 de enero 2016.

Msc. Lic. Octavio D. Aguilar M.
DIRECTOR DPTO. INFORMATICA Y SISTEMAS

Univ. Marcia Milenka Andrade Llanos
EXPOSITORA

UNIVERSIDAD AUTÓNOMA "JUAN MISAEL SARACHO"
FACULTAD DE CIENCIAS Y TECNOLOGÍA
DEPARTAMENTO DE INFORMÁTICA Y SISTEMAS

OTORGA EL PRESENTE CERTIFICADO DE CAPACITACIÓN

A: RONALDO ANDRÉS ANGULO DÁVILA

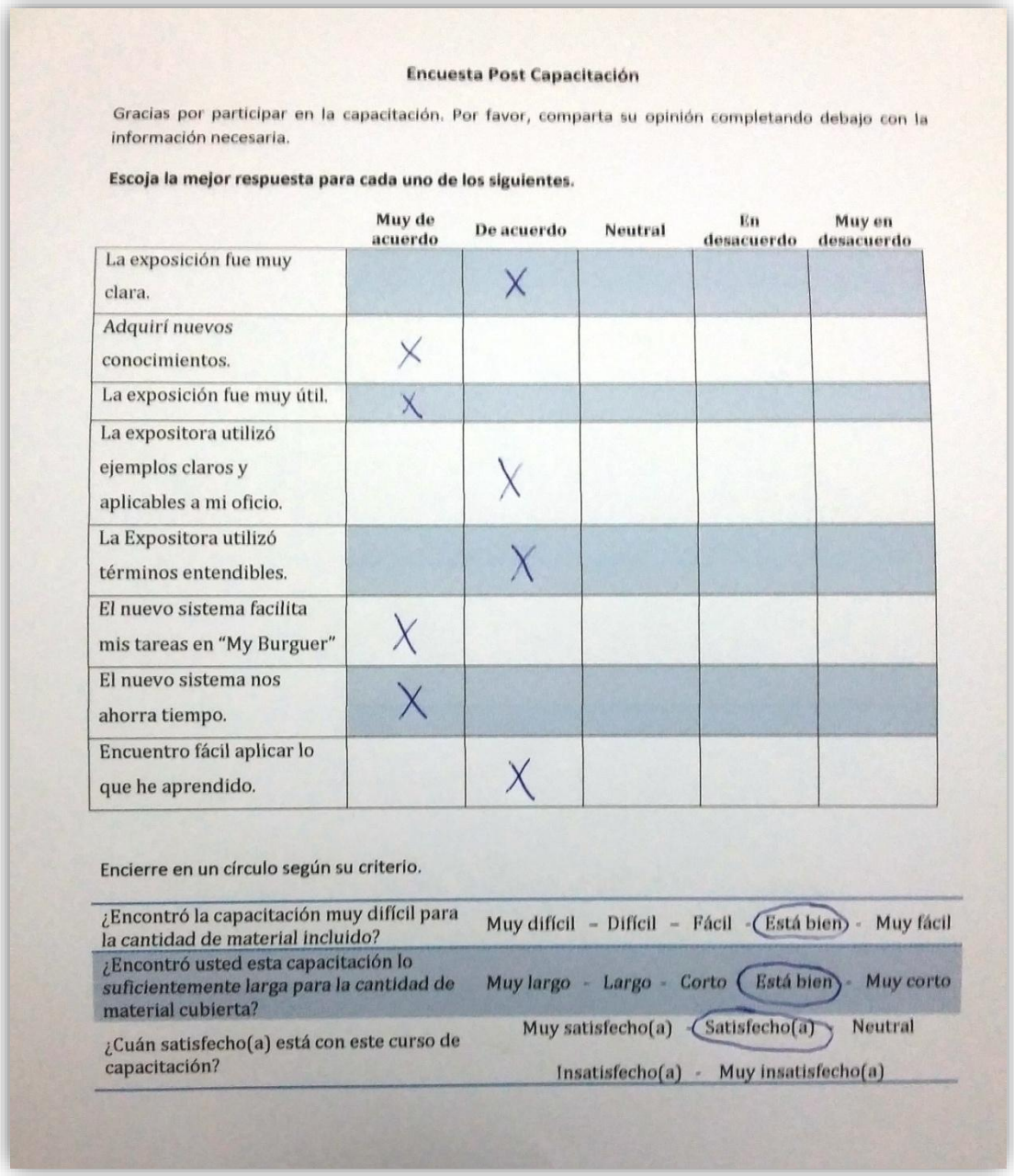
Por haber participado en el curso de capacitación de manejo del sistema **Mejoramiento en el manejo de la información y tiempo de atención al cliente mediante la gestión de reservas de pedidos en línea en "My Burguer"**, impartidos en el edificio de "My Burguer", los días lunes 11, martes 12 y miércoles 13 de enero 2016.

Msc. Lic. Octavio D. Aguilar M.
DIRECTOR DPTO. INFORMATICA Y SISTEMAS

Univ. Marcia Milenka Andrade Llanos
EXPOSITORA

Anexo 4: Encuestas post capacitación

Las presentes encuestas y evaluaciones fueron entregadas a las personas que asistieron a la capacitación.



Encuesta Post Capacitación

Gracias por participar en la capacitación. Por favor, comparta su opinión completando debajo con la información necesaria.

Escoja la mejor respuesta para cada uno de los siguientes.

	Muy de acuerdo	De acuerdo	Neutral	En desacuerdo	Muy en desacuerdo
La exposición fue muy clara.		X			
Adquirí nuevos conocimientos.	X				
La exposición fue muy útil.	X				
La expositora utilizó ejemplos claros y aplicables a mi oficio.		X			
La Expositora utilizó términos entendibles.		X			
El nuevo sistema facilita mis tareas en "My Burguer"	X				
El nuevo sistema nos ahorra tiempo.	X				
Encuentro fácil aplicar lo que he aprendido.		X			

Encierre en un círculo según su criterio.

¿Encontró la capacitación muy difícil para la cantidad de material incluido?	Muy difícil - Difícil - Fácil - Está bien - Muy fácil
¿Encontró usted esta capacitación lo suficientemente larga para la cantidad de material cubierta?	Muy largo - Largo - Corto - Está bien - Muy corto
¿Cuán satisfecho(a) está con este curso de capacitación?	Muy satisfecho(a) - Satisfecho(a) - Neutral - Insatisfecho(a) - Muy insatisfecho(a)

El resultado de la capacitación muestra buena aceptación por parte de la gerencia y personal de la empresa.

Capítulo III

Conclusiones y Recomendaciones

5. Conclusiones y Recomendaciones

5.1 Conclusiones

5.1.1 Componente I

- Se cumplió el objetivo principal del presente trabajo al desarrollar una herramienta que coadyuvará al área manejo de la información en atención al cliente mediante la gestión de reservas y en la realización de sus actividades.
- Se utilizó la metodología RUP para el desarrollo del sistema, permitió realizar mejoras al software, puesto que el proceso de ésta permite iterar y realizar las modificaciones necesarias para alcanzar una mayor calidad del software.
- Se cumplió el propósito principal donde se pueda contribuir al fin de dicha empresa.
- Se desarrolló satisfactoriamente la parte de reportes en el sistema.
- Se efectuó el registro de reservas de clientes de manera sistematizada y así optimizar los tiempos de atención al cliente y a la vez contribuir al fin.
- Se desarrolló un sistema que permite administrar los datos de combos y productos el cual contempla las siguientes funciones; registro de grupos intercambiables, registro de productos, asignación de productos que se van a intercambiar, registro de combos.

5.1.2 Componente II

- Se desarrolló la capacitacioin satisfactoriamente en la empresa “My Burguer”, la cual permitirá el uso adecuado del sistema para incrementar la productividad y competitividad de la misma.

5.2 Recomendaciones

- Poner en ejecución el sistema lo antes posible para que de esta manera puedan observarse los beneficios que ofrece el sistema en cuanto a seguridad y acceso de la información.
- Se debe contratar un profesional en el área de informática para una mejor administración y automatización de la información.
- Así mismo, se cuenta con un manual de sistema, que es de gran ayuda para la utilización del sistema, si en algún momento se tiene alguna duda en el manejo.
- Se recomienda que en un futuro cercano se implemente la parte de control de inventarios y contabilidad para la empresa “MyBurguer” y así mejorar la calidad de servicios en dicha empresa.