

DISEÑO DE UNA APLICACIÓN SOFTWARE PARA LA PLANEACIÓN Y  
SEGUIMIENTO DE OPERACIONES EN UN SISTEMA DE MANUFACTURA

ALVIN DÍAZ GUERRERO

MIGUEL ÁNGEL QUINTERO PIÑEROS

UNIVERSIDAD INDUSTRIAL DE SANTANDER  
FACULTAD DE INGENIERÍAS FISICOMECÁNICAS  
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA  
BUCARAMANGA

2014

DISEÑO DE UNA APLICACIÓN SOFTWARE PARA LA PLANEACIÓN Y  
SEGUIMIENTO DE OPERACIONES EN UN SISTEMA DE MANUFACTURA

ALVIN DÍAZ GUERRERO  
MIGUEL ÁNGEL QUINTERO PIÑEROS

TRABAJO DE GRADO PARA OPTAR POR EL TÍTULO DE  
INGENIERO DE SISTEMAS

DIRECTOR  
FERNANDO ANTONIO ROJAS MORALES

CODIRECTOR  
EDWIN ALBERTO GARAVITO HERNANDEZ

UNIVERSIDAD INDUSTRIAL DE SANTANDER  
FACULTAD DE INGENIERÍAS FISICOMECÁNICAS  
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA  
BUCARAMANGA

2014

## TABLA DE CONTENIDOS

INTRODUCCIÓN.....	15
1. OBJETIVOS.....	16
1.1. OBJETIVO GENERAL.....	16
1.2. OBJETIVOS ESPECÍFICOS.....	16
2. PLANTEAMIENTO Y JUSTIFICACIÓN DEL PROBLEMA.....	17
3. MARCO TEÓRICO.....	18
3.1. FUNDAMIENTOS TEÓRICOS.....	18
4. METODOLOGÍA.....	21
4.1. MODELO DE REQUERIMIENTOS .....	21
4.2. MODELO DE ANÁLISIS .....	21
4.3. MODELO DE DISEÑO.....	21
4.4. MODELO DE IMPLEMENTACIÓN.....	22
4.5. MODELO DE PRUEBAS.....	22
5. DESARROLLO DE LA METODOLOGÍA.....	23
5.1. MODELO DE REQUERIMIENTOS.....	23
5.1.1. Descripción del sistema deseado.....	23
5.1.2. Eventos de negocio.....	24

## TABLA DE CONTENIDOS

5.2.	PROBLEMAS ASOCIADOS A LOS EVENTOS.....	25
5.2.1.	Diagramas de casos de uso.....	26
5.2.2.	Especificaciones de los casos de uso.....	27
5.2.3.	Diagramas de secuencia del sistema.....	34
5.2.4.	Diagrama de clases.....	36
5.2.5.	Contratos de operación del sistema.....	39
5.3.	MODELO DE ANÁLISIS.....	42
5.3.1.	Listado de objetos por caso de uso.....	43
5.3.2.	Diagramas de secuencia con objetos.....	44
5.3.3.	Narrativas de casos de uso con objetos.....	48
5.3.4.	Diagrama de estados.....	51
5.3.5.	Diagrama de navegación por ventanas.....	52
5.4.	MODELO DE DISEÑO.....	54
5.4.1.	Subsistemas.....	54
5.4.2.	Diseño de la base de datos.....	58
5.4.3.	Diccionario de datos.....	59
5.4.4.	Control de accesos.....	66
5.4.5.	Casos de uso de condiciones de frontera.....	66
5.4.6.	Diseño de objetos.....	69
5.5.	MODELO DE PRUEBAS.....	75

## TABLA DE CONTENIDOS

6. CONCLUSIONES.....	80
7. RECOMENDACIONES.....	81
BIBLIOGRAFÍA.....	82
ANEXOS.....	83

## LISTADO DE TABLAS

Tabla 1. Lista de eventos asociadas al negocio.....	24
Tabla 2. Actores presentes en los casos de uso del sistema.....	29
Tabla 3. Narrativas de los casos de uso.....	30
Tabla 4. Listado de clases candidatas.....	36
Tabla 5. Diccionario de clases.....	38
Tabla 6. Contratos de operación del sistema.....	39
Tabla 7. Listado de objetos por caso de uso.....	43
Tabla 8. Narrativas de casos de uso con objetos.....	48
Tabla 9. Diccionario de subsistemas, primera capa.....	54
Tabla 10. Diccionario de subsistemas, segunda capa.....	55
Tabla 11. Diccionario de subsistemas, tercera capa.....	56
Tabla 12. Diccionario de datos del sistema.....	59
Tabla 13. Control de accesos a los usuarios.....	66
Tabla 14. Casos de uso de condiciones de frontera.....	66
Tabla 15. Tarjetas CRC del sistema.....	69
Tabla 16. Resultados pruebas de unidad.....	76

## LISTADO DE FIGURAS

Figura 1. Diagrama de casos de uso Usuario.....	26
Figura 2. Diagrama de casos de uso Gerente (Parte 1).....	27
Figura 3. Diagrama de casos de uso Gerente (Parte 2).....	28
Figura 4. Diagrama de secuencia Crear Pedido.....	34
Figura 5. Diagrama de secuencia Asignar espacio planta.....	35
Figura 6. Diagrama de secuencia Actualizar pedido.....	35
Figura 7. Diagrama de secuencia Consultar actualizaciones.....	36
Figura 8. Diagrama de Clases.....	37
Figura 9. Diagrama de secuencia con objetos Asignar espacio planta.....	45
Figura 10. Diagrama de secuencia con objetos Actualizar pedido.....	46
Figura 11. Diagrama de secuencia con objetos Consultar actualización.....	47
Figura 12. Diagrama de estados de Materia Prima.....	51
Figura 13. Diagrama de estados de Pedido.....	52
Figura 14. Diagrama de navegación por ventanas para Pedido.....	52
Figura 15. Diagrama de división por subsistemas, primera capa.....	54
Figura 16. Diagrama de división por subsistemas, segunda capa.....	55
Figura 17. Diagrama de división por subsistemas, tercera capa.....	56
Figura 18. Modelo entidad relación para la base de datos del sistema, msdb.....	58

## LISTADO DE ANEXOS

A. Narrativas de casos de uso.....	83
B. Diagramas de secuencia del sistema.....	101
C. Contratos de operación del sistema.....	111
D. Listado de objetos por caso de uso.....	126
E. Diagramas de secuencia con objetos.....	130
F. Narrativas de casos de uso con objetos.....	152
G. Diccionario de clases agrupado por módulos.....	167
H. Diagramas de navegación de ventanas.....	173
I. Casos de uso de condiciones de frontera.....	177
J. Tarjetas crc.....	186

## RESUMEN

**TÍTULO:** DISEÑO DE UNA APLICACIÓN SOFTWARE PARA LA PLANEACIÓN Y SEGUIMIENTO DE OPERACIONES EN UN SISTEMA DE MANUFACTURA.\*

**AUTORES:** MIGUEL ÁNGEL QUINTERO PIÑEROS, ALVIN DÍAZ GUERRERO.\*\*

**PALABRAS CLAVE:** Planeación, seguimiento, producción, manufactura, recursos, MVC, Ingeniería software.

### DESCRIPCIÓN:

Empresas pequeñas y medianas dedicadas a la fabricación de productos mediante el uso de pasos sucesivos, carecen de herramientas que permitan hacer un seguimiento a cada una de sus operaciones y procesos asociados. Esto impide tener precisión en la gestión de los recursos y definir tiempos de entrega, para cumplir con los compromisos pactados con los clientes.

La mayoría de esas empresas no cuentan con capital disponible para invertir en plataformas de información de un grado de robustez considerable, dichas organizaciones carecen de herramientas para realizar una correcta gestión tanto de la capacidad como de los recursos disponibles. Es por eso que se hace necesario el diseño de una aplicación que de manera sencilla, gráfica y general ofrezca ventajas sobre el manejo de las operaciones de manufactura presentes en estas organizaciones contribuyendo así con su crecimiento.

Se justifica el diseño de una herramienta software que permita llevar a cabo un seguimiento detallado y en tiempo real de las operaciones involucradas en sus procesos de manufactura y una correcta gestión de la capacidad y recursos disponibles. Para la creación de esta herramienta se usó una metodología de desarrollo software orientado a objetos, por tener la particularidad de ser un proceso iterativo y que cada una de sus etapas se realiza de manera paralela lo que permite un refinamiento constante de cada uno de los modelos y una reducción considerable de posibles fallos o errores en el producto final.

---

\* Trabajo de Grado.

\*\* Facultad de Ingenierías Fisicomecánicas. Escuela de Ingeniería de Sistemas e Informática.

Director: Fernando Antonio Rojas Morales.

Codirector: Edwin Alberto Garavito Hernandez.

## ABSTRACT

**TITLE:** DESIGN OF A SOFTWARE APPLICATION FOR PLANNING AND TRACKING OPERATIONS IN A MANUFACTURING SYSTEM.\*

**AUTHORS:** MIGUEL ANGEL QUINTERO PIÑEROS, ALVIN DIAZ GUERRERO.\*\*

**KEYWORDS:** Planning, Tracking, production, manufacturing, resources, MVC, Software Engineering.

### DESCRIPTION:

Small and medium enterprises dedicated to the manufacture of products using successive steps, do not have tools to track each one of its operations and associated processes. This prevents having precision in resource management and define delivery times, to meet commitments made to customers.

Most of these companies do not have enough capital to invest in information systems of a considerable level of robustness; these organizations do not have tools for appropriate management of both the capacity and the resources available. That is why is needed the design of an application that easily, offer advantages over the management of manufacturing operations involved in these organizations contributing to their growth is necessary.

Is justified, the design of a software application that allows to perform a detailed real-time tracking of the operations involved in their manufacturing processes and management of capacity and resources. In order to create this application an object-oriented methodology for software development was used, because it has the distinction of being an iterative process and in each of its stages is performed in a parallel way, this allows a constant refinement of each of the models and a considerable reduction of possible failures or errors in the final product.

---

\* Work Degree

\*\* Physicomechanical Faculty of Engineering, School of Systems Engineering.

Director: Fernando Antonio Rojas Morales.

Codirector: Edwin Alberto Garavito Hernandez.

## INTRODUCCIÓN

El constante avance de las tecnologías de información pone a disposición un buen número de recursos que permiten la creación de sistemas de información, cuya función sea dar soporte casi a cualquier tipo de organización sin importar cuál sea su razón social.

Dentro de estas organizaciones se encuentran las pequeñas empresas, las cuales no se pueden quedar atrás en el aprovechamiento de los beneficios ofrecidos por el uso de las TIC. Empresas tales como las dedicadas a la fabricación de productos que requieran de pasos sucesivos para su producción; las cuales, entre otras cosas, necesitan tener conocimiento del estado de la fabricación de los productos que allí se hagan en determinado momento; de igual forma, se hace necesario el correcto manejo de los recursos disponibles para la realización de los productos (personal, materiales, espacio en planta y maquinaria).

Este proyecto se centra en el diseño de una aplicación software que permita dar soporte a las actividades anteriormente mencionadas, y que con ayuda del mismo se puedan hacer de una manera más ordenada y eficiente, y que a su vez permita aprovechar la información producto de la en la empresa.

El presente documento incluye los objetivos a alcanzar con el desarrollo de la herramienta, su soporte teórico, los medios tecnológicos usados, la metodología usada y finalmente, el desarrollo de esta.

## **1. OBJETIVOS**

### **1.1. OBJETIVO GENERAL.**

Diseñar una herramienta software que permita hacer planeación y seguimiento de las operaciones en un sistema de manufactura.

### **1.2. OBJETIVOS ESPECIFICOS.**

- Diseñar un sistema de software que permita dar un tratamiento adecuado a la información presente en un sistema de manufactura para realizar de manera más eficiente las labores de planeación y programación de operaciones.
- Diseñar e implementar un módulo que permita visualizar el estado del proceso de fabricación de cada producto que está siendo elaborado en la empresa.
- Crear un modelo de aplicación adaptable a pequeñas empresas cuya estructura de operaciones involucren la ejecución de pasos sucesivos en la fabricación de sus productos.
- Implementar un módulo de software que permita la asignación ordenada de recursos teniendo en cuenta su disponibilidad al momento de programar un nuevo proyecto.

## 2. PLANTEAMIENTO Y JUSTIFICACIÓN DEL PROBLEMA

En el ámbito local, un buen número de empresas pequeñas y algunas medianas, utilizan sistemas de producción para ordenar los procesos de manufactura involucrados en la elaboración de sus productos; dependiendo de la naturaleza del producto, su fabricación involucra cierta cantidad de operaciones, por lo cual es importante tener conocimiento sobre el estado del producto en producción en el momento que sea necesario y en cualquier etapa de su fabricación, de tal manera que de presentarse alguna novedad ésta se pueda conocer de manera rápida y oportuna.

Actualmente varios sistemas se han posicionado en el mercado ofreciendo soluciones completas e integrales a las dificultades involucradas en los sistemas de producción, valiéndose de un control general sobre las variables, recursos, y espacios presentes, respondiendo así ante la creciente necesidad de productividad, innovación y competitividad de las grandes empresas. Pero ¿Qué pasa con aquellas pequeñas y las medianas? dónde no hay gran capital disponible para invertir en plataformas de información de un grado de robustez considerable, donde no hay suficiente personal para recibir capacitaciones para el uso de esas aplicaciones; de igual forma, dichas organizaciones carecen de herramientas para realizar una correcta gestión tanto de la capacidad como de los recursos disponibles, lo cual se traduce en la poca precisión a la hora de definir tiempos de entrega que permitan cumplir con los compromisos preestablecidos con los clientes. Es por eso que se hace necesario el diseño de una aplicación que de manera sencilla, gráfica y general ofrezca ventajas sobre el manejo de las operaciones de manufactura presentes en estas organizaciones contribuyendo así con su crecimiento.

Este proyecto propone el diseño de una herramienta software que de ser implementado en una empresa, permitirá llevar a cabo un seguimiento detallado y en tiempo real de las operaciones involucradas en sus procesos de manufactura y una correcta gestión de la capacidad y recursos disponibles.

### 3. MARCO TEÓRICO

#### 3.1. Fundamentos Teóricos

El uso de las aplicaciones software se ha vuelto imprescindible en la mayoría de empresas (cualquiera que sea su índole) que buscan cada día contar con herramientas que puedan dar solución a posibles problemas y sirvan como apoyo al desarrollo de tareas que se presentan como resultado de la operación diaria.

Se pretende así que una aplicación software sea capaz de apoyar al usuario en la realización de tareas para que con el uso estas herramientas se logre un aumento en el rendimiento y efectividad de las operaciones, que en la mayoría de los casos son realizadas por el ser humano y que al aumentar la complejidad de las mismas la aparición de errores se hace mayor, es aquí dónde estas aplicaciones pueden reducir el margen de error existente si se logra un buen diseño que sea capaz de cubrir los requerimientos exigidos.

**Sistemas de producción:** Un sistema de producción es un conjunto de operaciones y procesos que se relacionan entre si y cuyo fin es transformar las entradas suministradas en productos útiles. Se entiende por entradas las materias primas necesarias para la fabricación de dichos productos. Dentro de un sistema de producción son muchos los factores que juegan papeles de suma importancia para el éxito del mismo, tales son:

- **La gestión de recursos disponibles:** En las organizaciones gran parte de sus actividades y la capacidad de éxito de ella depende en gran parte de la gestión que se realice con sus recursos humanos, físicos o materiales, financieros, tecnológicos y los de conocimiento e información.
- **Recursos Humanos:** son las personas que ofrecen servicios usando sus habilidades y destrezas las cuales influyen en el manejo de los demás recursos de la empresa.
- **Recursos Físicos o materiales:** Son los bienes tangibles con que cuenta la empresa para poder ofrecer sus servicios, tales como: Disponibilidad en sus instalaciones, edificios, oficinas, muebles y

enseres, terrenos, instrumentos, herramientas, materia prima, productos terminados, entre otros.

- **Recursos financieros:** son recursos monetarios con los que cuenta la empresa ya sean propios o ajenos
- **Recursos tecnológicos:** De grado de complejidad mayor, estos recursos requieren de tecnología para su uso o utilización, pueden ser: sistemas de producción, de ventas, de finanzas, administrativos, maquinaria, equipos, vehículos y software. este tipo de recursos en su mayoría son operados por el recurso humano en la empresa
- **Recursos de conocimiento e información:** de gran importancia en una organización ya que son necesarios para la creación de nuevas y ideas y el crecimiento de la misma.

En las pequeñas empresas se omiten las buenas prácticas para una correcta gestión de los recursos debido a la ausencia de conocimiento y de herramientas de información asequibles ignorando así la posibilidad de ganar competitividad.

**Planeación de la producción:** Es la sistematización por anticipado los factores de mano de obra, materias primas, maquinaria y equipo y procesos para realizar la fabricación de un producto. Se pueden identificar tres tipos de planeación:

- **Planeación de procesos:** Es la que permite decidir que procesos y métodos deben usarse y la secuencia, determinar los requerimientos de habilitación de herramientas, seleccionar el equipo y los sistemas de producción
- **Planeación estratégica de la capacidad:** Provee un enfoque para determinar el nivel de capacidad general de los recursos con utilización intensiva de capital (instalaciones, equipos y tamaño global de la fuerza laboral) que mejor respalde la estrategia de competitividad de la compañía.
- **Planeación total:** Traduce los planes estratégicos corporativos y de capacidad a amplias categorías de tamaño de la fuerza laboral, cantidad de inventario y niveles de producción. La planeación se puede dar a

corto (1 día a 6 meses), mediano (6 meses a 18 meses) y largo plazo (1 año en adelante). Sea cual sea el tipo de planeación.

Dentro de una empresa manufacturera el proceso de planeación se resume de la siguiente manera:

1. Se organiza un programa maestro de producción
2. Se verifica la disponibilidad de instalaciones, mano de obra y que haya suministro de material cuando este se requiera esto se logra mediante la planeación de la capacidad a grandes rasgos.
3. Se hace una planeación de requerimientos de materiales lo cual permite la creación de un plan de materiales.
4. Asignar la capacidad de la producción mediante la planeación de requerimientos de capacidad.
5. Realizar una programación semanal o diaria a maquinas específicas, líneas de producción, o centros de trabajo. Esto se logra mediante la planeación final

**Programación:** Especifica un horario para ejecutar actividades utilizando recursos y asignando instalaciones. La capacidad se puede considerar cuando hay carga infinita es decir que está basada en lo que se necesita en el tiempo, no se considera si hay o no capacidad suficiente de recursos mientras que cuando hay carga finita se programa que hará cada recurso durante un día de trabajo. Se distinguen dos tipos de programación:

- **Programación futura:** se toma el pedido y se programa cada operación que se realizará más adelante en el tiempo y puede indicar la fecha en la que un pedido puede completarse.
- **Programación regresiva:** Las actividades se programan en secuencia inversa, indica cuando iniciará el pedido para que se efectúe en una fecha específica.

**Tiempo de entrega:** Es el tiempo que transcurre desde que se inicia un proceso de producción hasta que se completa, incluyendo normalmente el tiempo requerido para entregar ese producto al cliente.

## 4. METODOLOGÍA

Se trabajará con la metodología de desarrollo de software orientado a objetos, que tiene la particularidad de ser un proceso iterativo y que cada una de sus etapas se realiza de manera paralela lo que permite un refinamiento constante de cada uno de los modelos y una reducción considerable de posibles fallos o errores en el producto final. Esta metodología está compuesta de 5 modelos:

### 4.1. Modelo de requerimientos.

Etapa del proceso en la cual se delimitará y describirá el sistema a desarrollar. Es el punto de partida del proyecto y sirve como base para el resto de actividades ya que en éstas se tomará como guía todo lo que en esta etapa se haga, tiene la particularidad de que presentarse cualquier cambio en la funcionalidad del sistema este se puede implementar en esta etapa, ya que resulta mucho más sencillo y menos arriesgado. Este se basará principalmente en el modelo de casos de uso, la definición de cada uno de estos y por último un modelo de dominio del problema representado por un primer modelo de clases.

### 4.2. Modelo de análisis.

En esta etapa nos basaremos en el trabajo realizado en el modelo de requerimientos para generar una arquitectura de objetos lo suficientemente robusta para que permita ser implementada bajo casi cualquier entorno, esto con el fin de proporcionar una base fuerte para la siguiente etapa de esta metodología de desarrollo. Dicha arquitectura será implementada mediante la creación de tres tipos de objetos (borde, control, entidad) para cada uno de los casos de uso identificados en el modelo de requerimientos.

### 4.3. Modelo de diseño.

Con lo obtenido en el modelo de análisis y realizaremos un refinamiento y se trabajará en mayor detalle en la arquitectura del sistema, se definirá de manera específica las características y atributos de cada uno de los objetos. Por último se definirá el lenguaje de programación y el manejador de base de datos a utilizar.

#### **4.4. Modelo de Implementación.**

Basándonos en lo definido en el modelo de diseño iniciaremos la generación del código fuente final en el lenguaje de programación y manejador de base de datos escogido. Se establecerá una guía de programación de permita determinar un estilo de.

#### **4.5. Modelo de pruebas.**

Esta etapa del proceso será ejecutada durante todo el ciclo de desarrollo de la aplicación, lo cual permitirá descubrir posibles fallos en el sistema y se puedan corregir a tiempo. Cuando el modelo de implementación esté totalmente terminado se escogerá una técnica y nivel de pruebas que se considere acorde al sistema para evaluar el resultado final del proceso.

## **5. DESARROLLO DE LA METODOLOGÍA**

### **5.1. Modelo de requerimientos.**

#### **5.1.1. Descripción del sistema deseado.**

MS Seguimiento es un software orientado hacia una empresas pequeñas en la que la elaboración de sus productos requiera de una serie de eventos sucesivos, y que le permita a la gerencia hacer seguimiento a la fabricación de cada uno de sus pedidos, adicional a esto, la aplicación cubre actividades previas, paralelas y posteriores a dicho proceso; desde la consulta de disponibilidad de recursos, lo que permite crear con mayor seguridad el contrato, la creación de un nuevo pedido y su asignación de recursos (personal, materia prima y espacio en planta), la actualización y consulta del estado de cada uno de los pedidos en producción.

El proceso inicia cuando el cliente visita la empresa y manifiesta la intención de adquirir producto con determinadas características, las cuales son ingresadas por el gerente en un formulario dando origen a una cotización, que, de ser aceptada por el cliente servirá de base para la creación del contrato una vez se haya consultado la disponibilidad de recursos haciendo uso de la herramienta, lo que permitirá establecer una fecha de inicio y de finalización para el pedido; toda esta información junto con los datos del cliente y los costos de fabricación, serán consignados en el contrato, el cual se le asigna un código, que será generado por el sistema, dicho código será usado para generar el pedido, el cual será un formato que será usado para relacionar las características del producto a fabricar y la asignación de los recursos disponibles para tal fin.

Para iniciar la fabricación del producto, en primer lugar, se debe generar el pedido el cual reunirá la características de éste, además de ser el que recibirá la asignación de recursos la cual consiste en especificar el personal, la materia prima y el espacio en planta requeridos para la fabricación del producto, todo esto será asignado a un equipo de trabajo que será el responsable de lo anteriormente mencionado.

En lo que se refiere a materia prima, hay dos aspectos adicionales a la asignación: el primero es el ingreso de la materia prima al sistema, esto se hace cuando llegan nuevas existencias de esta. Para tal fin el almacenista es el encargado de hacer dicho ingreso se vale de un módulo de ingreso de

materia prima en el que ingresa el nombre, la cantidad y la marca del producto. El segundo aspecto es la entrega de la materia prima asignada, esto se hace cuando el proceso de fabricación del pedido va iniciar. Dicha entrega la hace el almacenista valiéndose del módulo de entrega de materia prima, el cual se vale del código de proyecto para mostrar la materia prima que se ha asignado y los responsables de la misma.

Una vez el proceso de fabricación haya iniciado será supervisado por una persona que deberá reportar y actualizar el estado del producto durante cada paso en el proceso de fabricación, estas actualizaciones quedarán almacenadas en el sistema para su posterior consulta.

El gerente podrá consultar en todo momento el estado de cada uno de los pedidos en ejecución a los cuales se les haya actualizado su estado. El proceso de consulta se podrá hacer ingresando el código de cada proyecto o mediante un panel de visualización general que permitirá visualizar todos los pedidos en fabricación.

### 5.1.2. Eventos de negocio.

Teniendo en cuenta la descripción del sistema deseado, y la información recolectada, se lograron identificar los siguientes eventos de negocio, este es el paso inicial que permitirá identificar el modelo de casos de uso del sistema.

**Tabla 1. Lista de eventos asociadas al negocio.**

No.	Evento	Entrada al Sistema	Actor que provee la Entrada	Salida del Sistema	Actor que recibe la Salida
1	Cliente encarga producto	Especificaciones del producto	Cliente	Detalles del producto	Gerente
2	Gerente verifica recursos disponibles	Solicitud de información de disponibilidad	Gerente	Información de disponibilidad	Gerente

3	Gerente genera cotización	Detalles del producto solicitado	Gerente	Cotización	Cliente, Gerente
4	Gerente genera contrato	Cotización, fechas de inicio y fin de producción	Gerente	Contrato	Cliente, Gerente
5	Gerente asigna recursos	Contrato	Gerente	Formato asignación de recursos	Gerente, empleados y almacenista
6	Supervisor actualiza estado del producto	Estado actual del producto	Supervisor	Estado del producto actualizado	Supervisor, Gerente
7	Gerente consulta estado del producto	Datos del producto	Gerente	Estado del producto	Gerente

### 5.1.3. Problemas asociados a los eventos.

Se pueden identificar los siguientes problemas:

- La adquisición de la información de la disponibilidad de recursos se hace de manera lenta, y hay riesgo de que la información no sea concreta o fiable ya que esta se da de manera verbal.
- Una información de disponibilidad de recursos errada puede llevar a que se acepten pedidos y que estos no se realicen en el tiempo establecido.
- No se utiliza de una manera eficiente la información generada al recibir un nuevo pedido.
- Se puede dar origen a desperdicio de materia prima al no tener un control en la asignación de esta.

- La información actualizada del estado de un producto no está disponible en tiempo real.
- La información del estado del producto no se presenta a la gerencia de una manera clara y ordenada.
- No hay control sobre los registros de ingreso de material.
- No hay inventario de materia prima.

#### 5.1.4. Diagramas de casos de uso.

Con base en la información obtenida en la descripción del sistema deseado, los eventos de negocio y los problemas identificados, a continuación se presentan los diagramas de casos de uso, los cuales representan la interacción de los actores con el sistema.

**Figura 1. Diagrama de casos de uso Usuario**

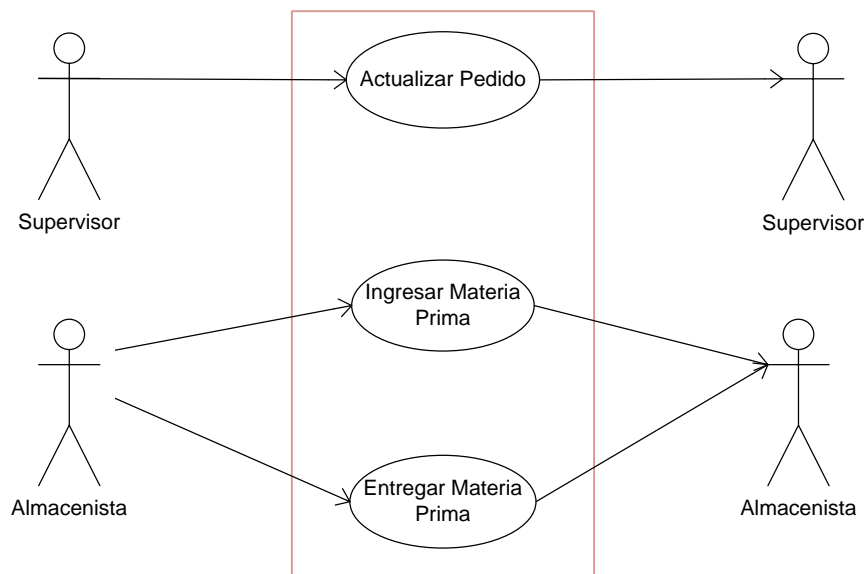


Figura 2. Diagrama de casos de uso Gerente (Parte 1)

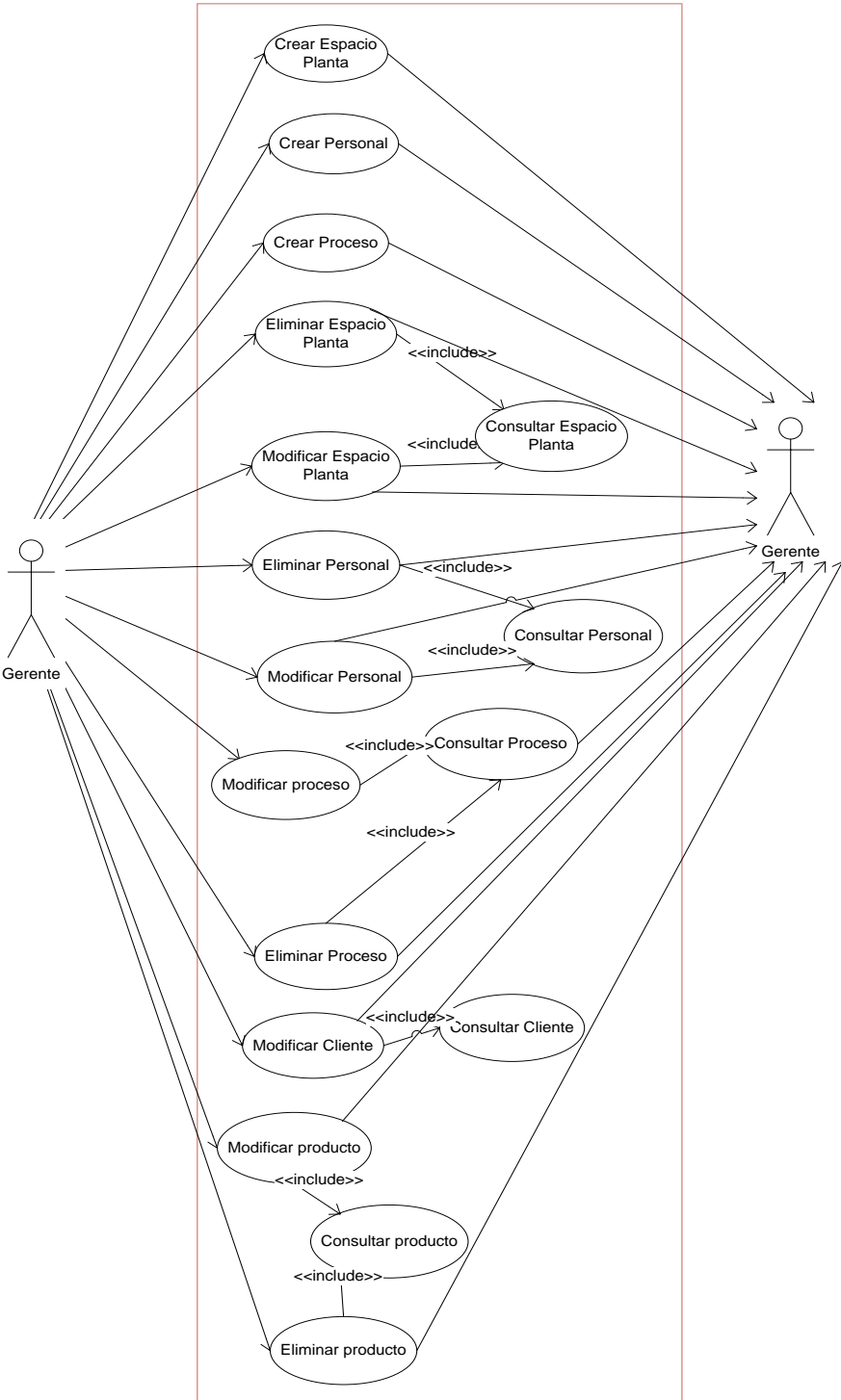
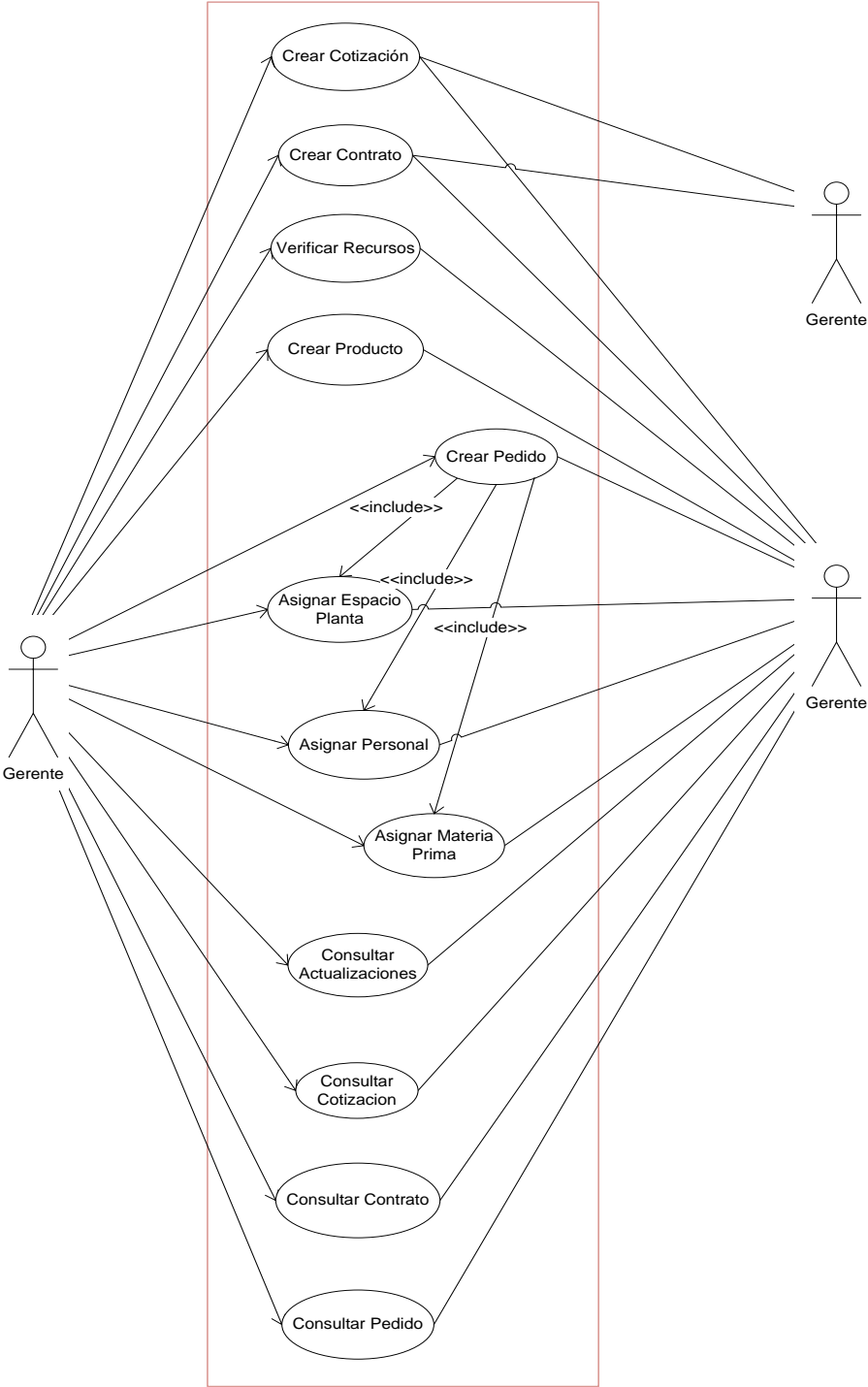


Figura 3. Diagrama de casos de uso Gerente (Parte 2)



### 5.1.5. Especificaciones de los casos de uso

Con el fin de facilitar la comprensión de cada uno de los artefactos producidos, sólo se mostrarán los realizados para los casos de uso que representan la parte central de la aplicación, específicamente, lo que tiene que ver con creación de pedido, asignación de recursos, actualización y consulta de actualización de pedidos. El desarrollo de la metodología para el resto de casos de uso, se podrá encontrar en la sección de anexos.

#### Actores.

La siguiente tabla, muestra a los actores presentes en los casos de uso, relacionándolos con los casos de uso en los que intervienen y una breve descripción de sus funciones.

**Tabla 2. Actores presentes en los casos de uso del sistema.**

<b>Actor</b>	Gerente
<b>Casos de uso</b>	Crear cotización, crear contrato, verificar espacio planta, verificar personal, verificar materia prima, crear pedido, asignar espacio planta, asignar personal, asignar materia prima, consultar cotización, consultar contrato, consultar pedido, consultar actualizaciones, crear espacio planta, crear personal, crear proceso, eliminar espacio planta, eliminar personal, modificar proceso, eliminar proceso, modificar personal, modificar espacio planta, modificar cliente, crear pedido, modificar pedido, eliminar pedido.
<b>Descripción</b>	Es el encargado de hacer la cotización del producto para que sea presentada al cliente, verifica la disponibilidad de recursos (espacio en planta, materia prima y personal), genera el contrato y el pedido para iniciar el proceso de fabricación del mismo una vez haya asignado recursos; adicionalmente es el encargado de realizar las operaciones de creación, modificación y eliminación sobre los recursos y los datos de los clientes. Por último, tiene la facultad de consultar cotización, contrato, pedido y el estado de cada producto en producción.
<b>Actor</b>	Cliente
<b>Casos de uso</b>	Crear cotización, crear contrato.
<b>Descripción</b>	Es el que proporciona los detalles del pedido necesarios

	para que el gerente haga una cotización, también recibe el contrato generado por el sistema.
<b>Actor</b>	Supervisor
<b>Casos de uso</b>	Actualizar pedido.
<b>Descripción</b>	Tiene como tarea principal ingresar las actualizaciones para cada pedido en producción, de ser necesario puede tener acceso a dichas actualizaciones mediante consulta.
<b>Actor</b>	Almacenista
<b>Casos de uso</b>	Ingresar materia prima, entregar materia prima
<b>Descripción</b>	Es el encargado de recibir la materia prima e ingresarla en el sistema, así mismo de hacer efectiva la entrega de materia prima, previamente asignada por el gerente.
<b>Actor</b>	Operario
<b>Casos de uso</b>	Entregar materia prima
<b>Descripción</b>	Es el encargado de recibir por parte del almacenista la materia prima asignada por el gerente.

### **Narrativa Casos de Uso.**

Con el fin de aumentar la comprensión de los diagramas de casos de uso, se presentan las siguientes narrativas que describen el propósito de cada caso de uso, una breve descripción y las secuencias de interacción entre el usuario y el sistema.

**Tabla 3. Narrativas de los casos de uso.**

<b>Caso de uso</b>	Crear pedido
<b>Actor(es)</b>	Gerente
<b>Propósito</b>	Crear un formato que permita relacionar los detalles del producto a fabricar con su respectiva asignación de recursos.
<b>Descripción</b>	El gerente ingresa al módulo de crear pedido e ingresa el código del contrato creado, una vez validado se desplegarán las secciones para la asignación de recursos disponibles de la misma manera que se

	mostrarán los detalles de fabricación del producto correspondiente.	
<b>Precondición(es)</b>	Debe existir un contrato para la creación de su respectivo contrato.	
<b>Flujo Principal</b>	<b>Acción de actor(es)</b>	<b>Respuesta(s) del sistema</b>
	<p>1. El gerente ingresa al módulo de creación de pedido.</p> <p>3. El gerente ingresa un número de contrato para la creación del pedido y pulsa aceptar.</p> <p>5. El gerente realiza la asignación de recursos y pulsa el botón crear.</p> <p>7. El gerente confirma la operación.</p>	<p>2. El sistema despliega la ventana de validación de código de contrato</p> <p>4. El sistema valida el código ingresado y despliega el formulario de creación de pedido el cual contiene toda la información referente al producto.</p> <p>6. El sistema solicita confirmación del envío del formulario.</p> <p>8. El sistema confirma la creación del pedido con asignación de recursos por medio de un mensaje mostrado en pantalla y muestra la información del pedido recién creado.</p>
<b>Sub-flujos</b>	<p>1. El código de contrato no existe, regresa al paso 3.</p> <p>2. Si al pulsar el botón crear, no se ha realizado alguna de las asignaciones el sistema muestra un mensaje solicitando completar la asignación de recursos, regresa al paso 5.</p>	
<b>Pos-condición(es)</b>	El contrato ha sido creado y los recursos asignados.	
<b>Caso de uso</b>	Asignar espacio planta	
<b>Actor(es)</b>	Gerente	
<b>Propósito</b>	Asignar el espacio en planta para la fabricación de un nuevo pedido.	
<b>Descripción</b>	Inicia cuando el pedido requiere espacio en planta para iniciar su fabricación.	
<b>Precondición(es)</b>	Debe existir disponibilidad de recursos	

<b>Flujo Principal</b>	<b>Acción de actor(es)</b>	<b>Respuesta(s) del sistema</b>
	1. Desde el formulario de creación de pedido el gerente selecciona la opción de asignar espacio en planta	2. El sistema despliega el formulario para la asignación de espacio en planta
	3. El gerente selecciona las zonas que desea asignar para la fabricación del producto y presiona el botón asignar espacio.  5. El gerente confirma la asignación.	4. El sistema solicita confirmación de la operación.  6. El sistema despliega el formulario de creación de pedido en donde se muestra la asignación realizada
<b>Sub-flujos</b>	1. Si se presiona el botón Asignar espacio, y no se ha seleccionado ninguna zona, se muestra un mensaje en pantalla solicitando seleccionar por lo menos uno, regresa al paso 3.	
<b>Pos-condición(es)</b>	Se ha seleccionado las zonas a asignar.	
<b>Caso de uso</b>	Consultar actualizaciones	
<b>Actor(es)</b>	Gerente	
<b>Propósito</b>	Desplegar todas las actualizaciones y novedades de un pedido en particular o de los pedidos en fabricación.	
<b>Descripción</b>	Inicia cuando el gerente se dirige al módulo de consulta y ve el estado de todos los pedidos, también puede seleccionar un pedido de la lista para revisarlo en detalle o si lo prefiere ingresar el código del pedido que desea consultar.	
<b>Precondición(es)</b>	Deben existir pedidos creados	
<b>Flujo Principal</b>	<b>Acción de actor(es)</b>	<b>Respuesta(s) del sistema</b>
	1. El gerente ingresa al módulo de consulta de actualizaciones.  3. El gerente ingresa el código del pedido a	2. El sistema despliega una ventana con el listado de pedidos en ejecución y un campo para ingresar un código de pedido.  4. El sistema despliega una ventana en la que se

	consultar, o selecciona un pedido de la lista y presiona el botón Consultar.	muestran los progresos que se han logrado en la elaboración del producto.
<b>Sub-flujos</b>	1. Si al presionar el botón Consultar no hay ningún código ingresado en el campo de código, se mostrará un mensaje en el que informe que el campo se debe diligenciar, regresa al paso 3.	
<b>Pos-condición(es)</b>		
<b>Caso de uso</b>	Actualizar pedido	
<b>Actor(es)</b>	Supervisor	
<b>Propósito</b>	Mantener a la gerencia al tanto de los avances y estados actuales de cada pedido en fabricación.	
<b>Descripción</b>	Inicia cuando el supervisor observa un nuevo avance, novedad o estado sobre el progreso de un pedido en fabricación, entonces el ingresa al módulo de actualización y según el número del pedido ingresa las novedades detectadas.	
<b>Precondición(es)</b>	Deben existir por lo menos un pedido creado	
<b>Flujo Principal</b>	<b>Acción de actor(es)</b>	<b>Respuesta(s) del sistema</b>
	1. El supervisor ingresa al módulo de ingreso de actualizaciones.	2. El sistema despliega una ventana de validación de código de pedido.
	3. El supervisor ingresa el código del pedido a actualizar y presiona el botón Actualizar.	4. El sistema despliega una ventana que contiene todos los procesos a realizar dentro del pedido, y muestra cuales actualizaciones se han ingresado.
	5. El supervisor ingresa las actualizaciones al pedido y las respectivas observaciones a lugar, y presiona el botón Enviar Actualizaciones.	6. El sistema solicita confirmación de la operación.
7. El supervisor confirma la operación.	8. El sistema confirma la actualización del pedido por medio de un mensaje en pantalla y pregunta si se va	

	realizar otra actualización.
<b>Sub-flujos</b>	1. Si al presionar el botón actualizar no hay ningún código ingresado en el campo código, el sistema informará que se debe llenar el campo, regresa el paso 3.
<b>Pos-condición(es)</b>	El pedido ha sido actualizado.

### 5.1.6. Diagramas de secuencia del sistema.

Los diagramas de secuencia del sistema, son la representación gráfica de la interacción descrita en la narrativa de cada caso de uso.

**Figura 4. Diagrama de secuencia Crear Pedido.**

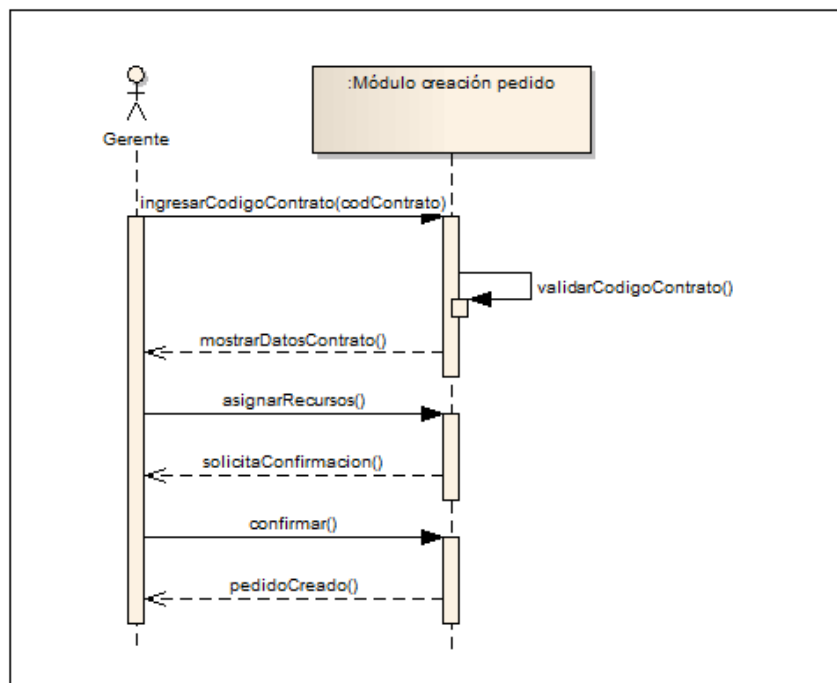


Figura 5. Diagrama de secuencia Asignar espacio planta.

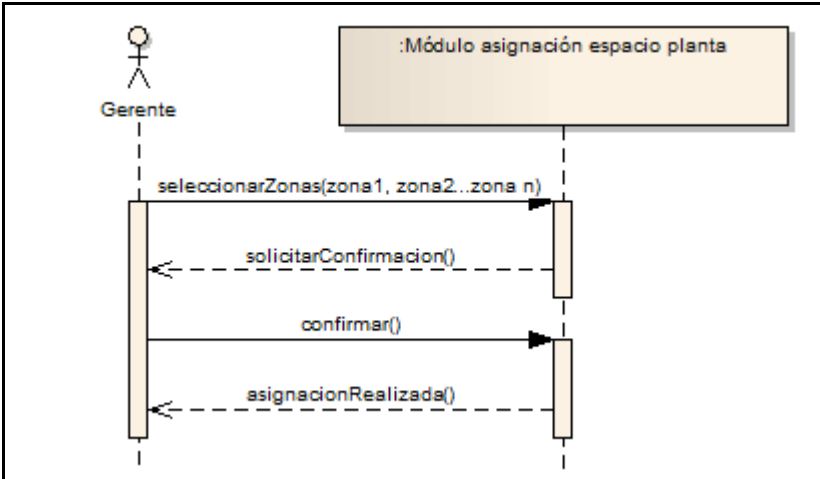
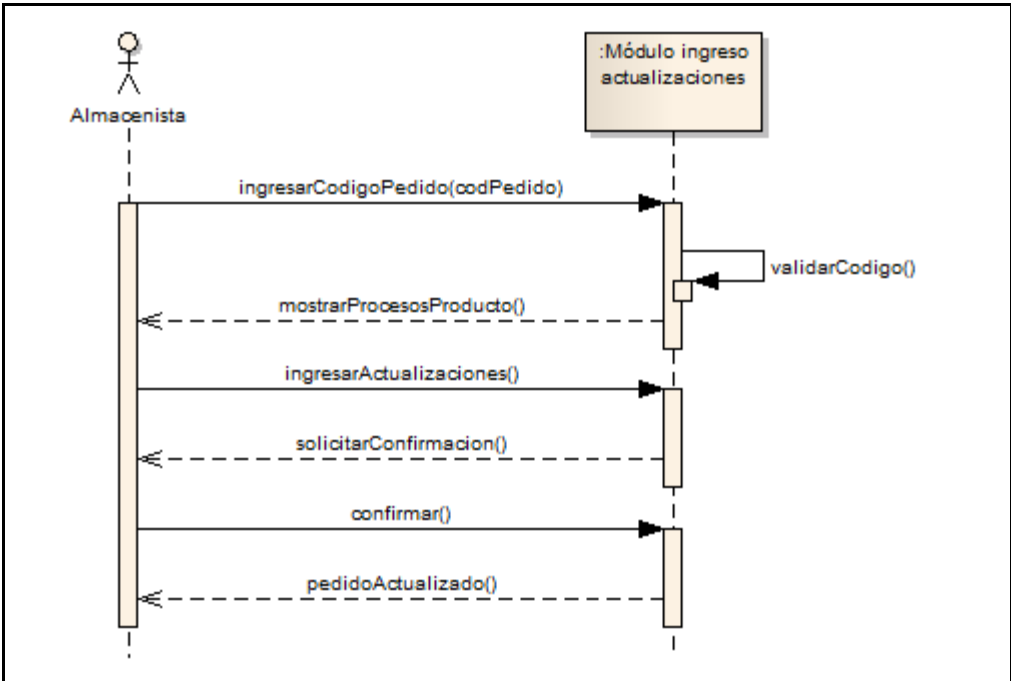
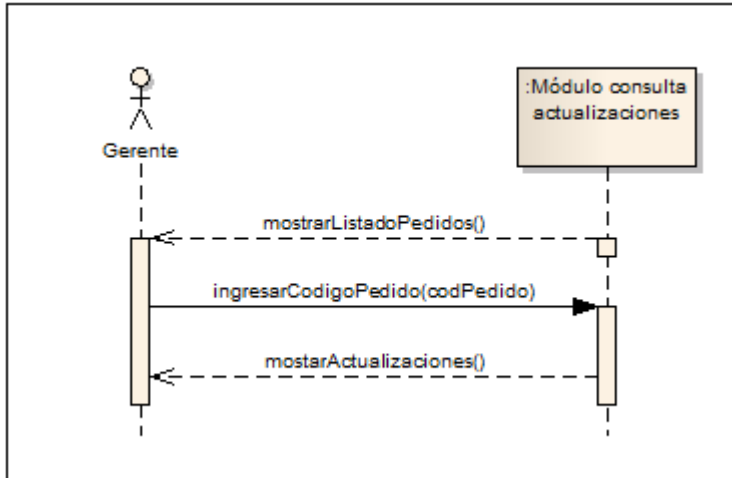


Figura 6. Diagrama de secuencia Actualizar pedido.



**Figura 7. Diagrama de secuencia Consultar actualizaciones.**



#### 5.1.7. Diagrama de clases

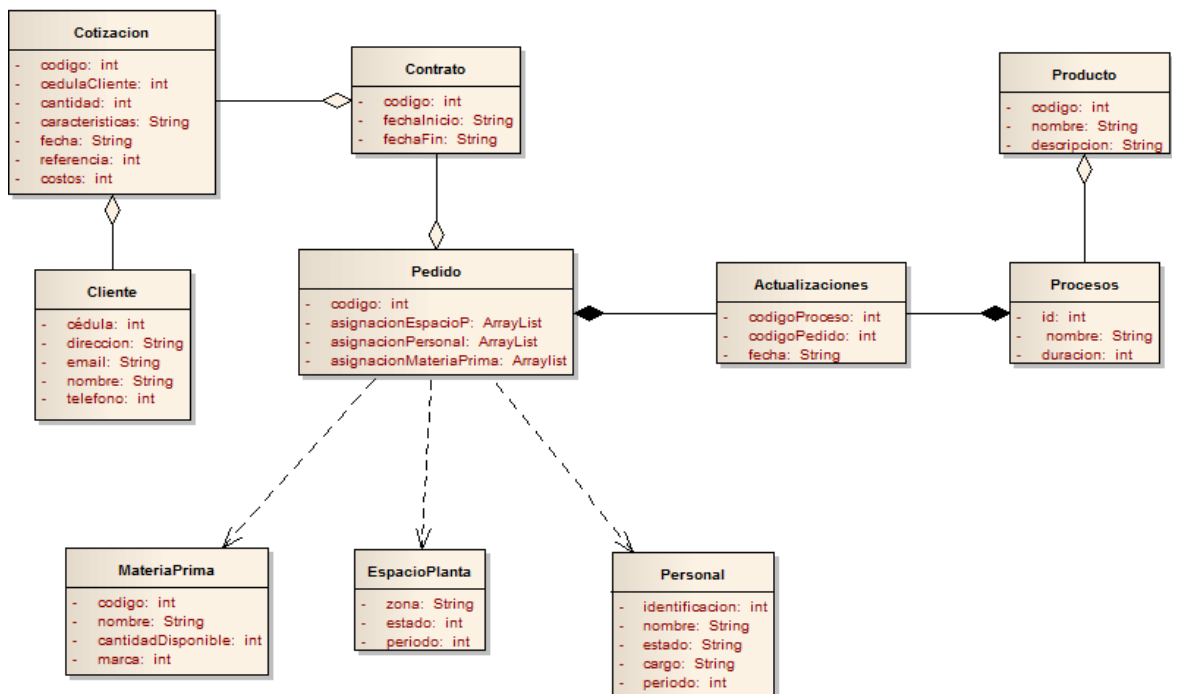
Con base en la descripción del sistema deseado, se identifican todos los sustantivos, los cuales en el campo de la ingeniería del software serán candidatas a ser clases dentro del sistema. Las clases escogidas son las que presentan mayor relevancia al problema descrito anteriormente y tienen sentido dentro de la aplicación.

**Tabla 4. Listado de clases candidatas**

MS Seguimiento	Cotización
Software	Creación del contrato
Gerencia	Herramienta
Seguimiento	Fecha de inicio
Fabricación	Fecha de finalización
Pedidos	Información
Aplicación	Datos del cliente
Actividades	Costos de fabricación
Proceso	Código
Consulta	Sistema
Disponibilidad de recursos	Formato
Contrato	Características del producto
Personal	Asignación de los recursos
Materia prima	Fabricación del producto
Espacio en planta	Personal

Consulta de estado Producción Cliente Producto Características Gerente Formulario	Equipo de trabajo Proceso de fabricación Persona Estado del producto Negocio Panel de visualización
---	--

**Figura 8. Diagrama de Clases.**



### Diccionario de clases

En la siguiente tabla se muestra cada una de las clases y su función dentro del sistema.

**Tabla 5. Diccionario de clases.**

<b>Clase</b>	<b>Descripción</b>
<b>Cotización</b>	Contiene toda la información del cliente y el producto que se está cotizando.
<b>Contrato.</b>	Identificado por el número de contrato contiene toda la información del producto a fabricar, la fecha de inicio y fin de la fabricación. Es necesario que exista una cotización para poder generar un contrato.
<b>Pedido.</b>	Se identifica con el número de pedido. Al momento de su creación debe efectuarse la respectiva asignación de los recursos correspondientes, teniendo en cuenta el producto especificado en el contrato.
<b>Espacio en planta</b>	Se identifica por el número de espacio en planta; especifica la zona que se asignará para la creación de un pedido.
<b>Materia prima</b>	Identificada por el número de materia prima, contiene la información de la materia prima que se utiliza para ser asignada en los pedidos.
<b>Personal.</b>	Representa los empleados de la empresa que serán asignados a cada pedido para su fabricación, cada elemento de esta clase será identificado con un número de persona.
<b>Proceso.</b>	Identificado con el número de proceso, en él se describe cada uno de los pasos y etapas por los que pasa un producto durante su fabricación.
<b>Actualización.</b>	Permite reportar cada una de las novedades o procesos finalizados en la fabricación de un producto.
<b>Producto</b>	Identificado con el número de producto, identifica todos los productos que la compañía fabrica, dichos productos permiten identificar la lista de procesos sobre los cuales serán objeto seguimiento.
<b>Cliente</b>	Identificado con un número de cédula, especifica los clientes que está relacionado con cada una de las cotizaciones.

**Descripción de relaciones del diagrama de clases.**

Inicialmente, se sabe que dentro de cada cotización hay un cliente, y ese cliente puede pertenecer a muchas cotizaciones, dicho de otro modo la información de un cliente está presente en una cotización, por tal motivo se usa una asociación por agregación entre la clase Cliente y la clase Cotización. De igual forma, la existencia de una cotización puede dar origen a un contrato y buena parte de la información presente en la cotización a la de un contrato;

el mismo caso se presenta con pedido y contrato, ya que la existencia de un contrato da origen a un pedido y buena parte de la información presente en ese contrato hará parte de un pedido, por tal motivo al ser la cotización parte de un contrato, y el contrato parte de un pedido, se utiliza una relación de agregación entre estas clases.

Para que una creación de pedido se realice de manera exitosa se requiere que exista materia prima, espacio planta y personal disponible para hacerlo, por lo que se puede decir que pedido depende directamente de estas tres clases; por tal motivo se usa una relación de dependencia entre ellos. En el caso del producto, su fabricación se lleva a cabo realizando una serie de procesos, por lo que se puede decir que un producto está compuesto por la realización de procesos, por lo que la asociación entre ellos es de agregación.

Por último, las actualizaciones contienen el cumplimiento de un proceso para un pedido que se está realizando, así bien, para realizar una actualización se debe contar con que exista un pedido y procesos ligados a este, por lo que la relación que se establece entre actualización y pedido, y actualización y proceso es de composición, ya que una actualización no tendría razón de ser sin estos dos.

#### 5.1.8. Contratos de operación del sistema.

Los contratos de operación describen los cambios de estado de todo el sistema cuando una operación es invocada, estas operaciones son aquellas representadas en el diagrama de secuencia, especifican las condiciones anteriores y posteriores a la realización de dicha tarea.

**Tabla 6. Contratos de operación del sistema.**

<b>Caso de uso</b>	CrearPedido
<b>Nombre del Contrato</b>	IngresarCodigoContrato(codContrato)
<b>Responsabilidades</b>	Recibir y enviar el código del pedido
<b>Tipo</b>	Sistema
<b>Excepciones</b>	-El código recibido y enviado no existe
<b>Salida</b>	-
<b>Pre condición(es)</b>	-El sistema debe tener contratos ya creados -La creación de un pedido debe estar en curso
<b>Post condición(es)</b>	-

<b>Caso de uso</b>	CrearPedido
<b>Nombre del Contrato</b>	asignarRecursos()
<b>Responsabilidades</b>	Recibir la selección de recursos para relacionarlos con cada pedido.
<b>Tipo</b>	Sistema.
<b>Excepciones</b>	No haya recursos disponibles para asignar.
<b>Salida</b>	-
<b>Pre condición(es)</b>	-El sistema debe contar con recursos ingresados. -El proceso de asignación debe estar en curso.
<b>Post condición(es)</b>	-
<b>Caso de uso</b>	CrearPedido
<b>Nombre del Contrato</b>	confirmar()
<b>Responsabilidades</b>	-Efectuar la creación de un nuevo pedido en la base de datos.
<b>Tipo</b>	Sistema
<b>Excepciones</b>	Operación cancelada o declinada
<b>Salida</b>	-Se agregan cada una de las selecciones de recursos al pedido. -Se descuentan de la cantidad de los recursos disponibles los recursos asignados. -Se crea un nuevo registro en la tabla pedido
<b>Pre condición(es)</b>	Recibir la solicitud de creación de pedido
<b>Post condición(es)</b>	-El nuevo pedido es relacionado con su lista de procesos. -El nuevo pedido es relacionado con una selección de recursos disponibles. -La selección de materia prima es relacionada con la selección de personal, ambos recursos disponibles asociados al pedido
<b>Caso de uso</b>	AsignarEspacioPlanta
<b>Nombre del Contrato</b>	seleccionarZona(zona1, zona2, ....., zona n)
<b>Responsabilidades</b>	Capturar y enviar una selección de uno o más zonas para asignar
<b>Tipo</b>	Sistema
<b>Excepciones</b>	-No hay zonas para mostrar. -No se ha seleccionado ninguna zona
<b>Salida</b>	-

<b>Pre condición(es)</b>	-Deben existir EspacioPlanta creado en el sistema
<b>Post condición(es)</b>	-el proceso de creación de pedido debe estar en curso
<b>Caso de uso</b>	
<b>Nombre del Contrato</b>	AsignarEspacioPlanta confirmar()
<b>Responsabilidades</b>	Efectuar la asignación de EspacioPlanta
<b>Tipo</b>	Sistema
<b>Excepciones</b>	La asignación sea cancelada o declinada
<b>Salida</b>	Se agrega una selección de EspacioPlanta dentro de la asignación de recursos asociada a un pedido
<b>Pre condición(es)</b>	Recibir la solicitud de asignar EspacioPlanta
<b>Post condición(es)</b>	-La selección de EspacioPlanta es relacionada con un pedido -La disponibilidad de las zonas es modificada al momento de la asignación.
<b>Caso de uso</b>	
<b>Nombre del Contrato</b>	ActualizarPedido ingresarCodigoPedido(codPedido)
<b>Responsabilidades</b>	Capturar y enviar el código del pedido para cargar la información de sus procesos asociados
<b>Tipo</b>	Sistema
<b>Excepciones</b>	No existe ese número de pedido
<b>Salida</b>	-
<b>Pre condición(es)</b>	Debe existir un pedido creado
<b>Post condición(es)</b>	-
<b>Caso de uso</b>	
<b>Nombre del Contrato</b>	ActualizarPedido ingresarActualizaciones()
<b>Responsabilidades</b>	Capturar y enviar las actualizaciones marcadas como completas
<b>Tipo</b>	Sistema
<b>Excepciones</b>	No se encontró alguna actualización marcada
<b>Salida</b>	-
<b>Pre condición(es)</b>	-Deben existir procesos cargados -El procesos de actualización de pedido debe estar en curso
<b>Post condición(es)</b>	-
<b>Caso de uso</b>	
<b>Nombre del Contrato</b>	ActualizarPedido

<b>Nombre del Contrato</b>	Confirmar()
<b>Responsabilidades</b>	Efectuar el procesos de actualización de pedido
<b>Tipo</b>	Sistema
<b>Excepciones</b>	-La operación de actualización es cancelada o declinada
<b>Salida</b>	-Se agregan en la tabla de actualización las actualizaciones con la fecha de captura
<b>Pre condición(es)</b>	-Recibirla solicitud de actualización
<b>Post condición(es)</b>	-se crea una relación con el número de pedido en cuestión. -las actualizaciones quedan para ser consultadas
<b>Caso de uso</b>	ConsultarActualizacion
<b>Nombre del Contrato</b>	ingresarCodigoPedido(codPedido)
<b>Responsabilidades</b>	Capturar y enviar el código del pedido para cargar la información de las actualizaciones disponibles
<b>Tipo</b>	Sistema
<b>Excepciones</b>	No hay actualizaciones
<b>Salida</b>	-
<b>Pre condición(es)</b>	Debe existir un pedido creado
<b>Post condición(es)</b>	-

## 5.2. MODELO DE ANÁLISIS.

Una de las partes principales de esta metodología es la identificación de los objetos que conformarán la arquitectura del sistema, ya que estos serán parte fundamental para el modelo de análisis. El tipo de arquitectura escogido es la arquitectura MVC (Modelo vista controlador), en el cual la vista hace referencia a los bordes o interfaces con los cuales el usuario interactúa con el sistema; el modelo hace referencia a toda la información presente en el sistema y que requiere ser almacenada en una base de datos; y el controlador se encarga de manipular la información entre la vista y el modelo.

Los objetos borde fueron escogidos identificando los momentos en los que el usuario ingresa información al sistema, así mismo, cuando este emite alguna respuesta. Para los objetos entidad, se identificaron todas las fuentes de información para cada caso de uso dentro del modelo de dominio del sistema, cabe mencionar que dicha información irá cambiando con el tiempo y será

necesario almacenarla en una base de datos. Por último, los objetos control fueron creados a medida que se hacía necesario contar con objetos que manejaran la comunicación entre los otros dos tipos de objetos.

Para tener una mayor comprensión acerca de cada uno de los objetos, se recomienda consultar las tarjetas CRC.

### 5.2.1. Listado de objetos por caso de uso.

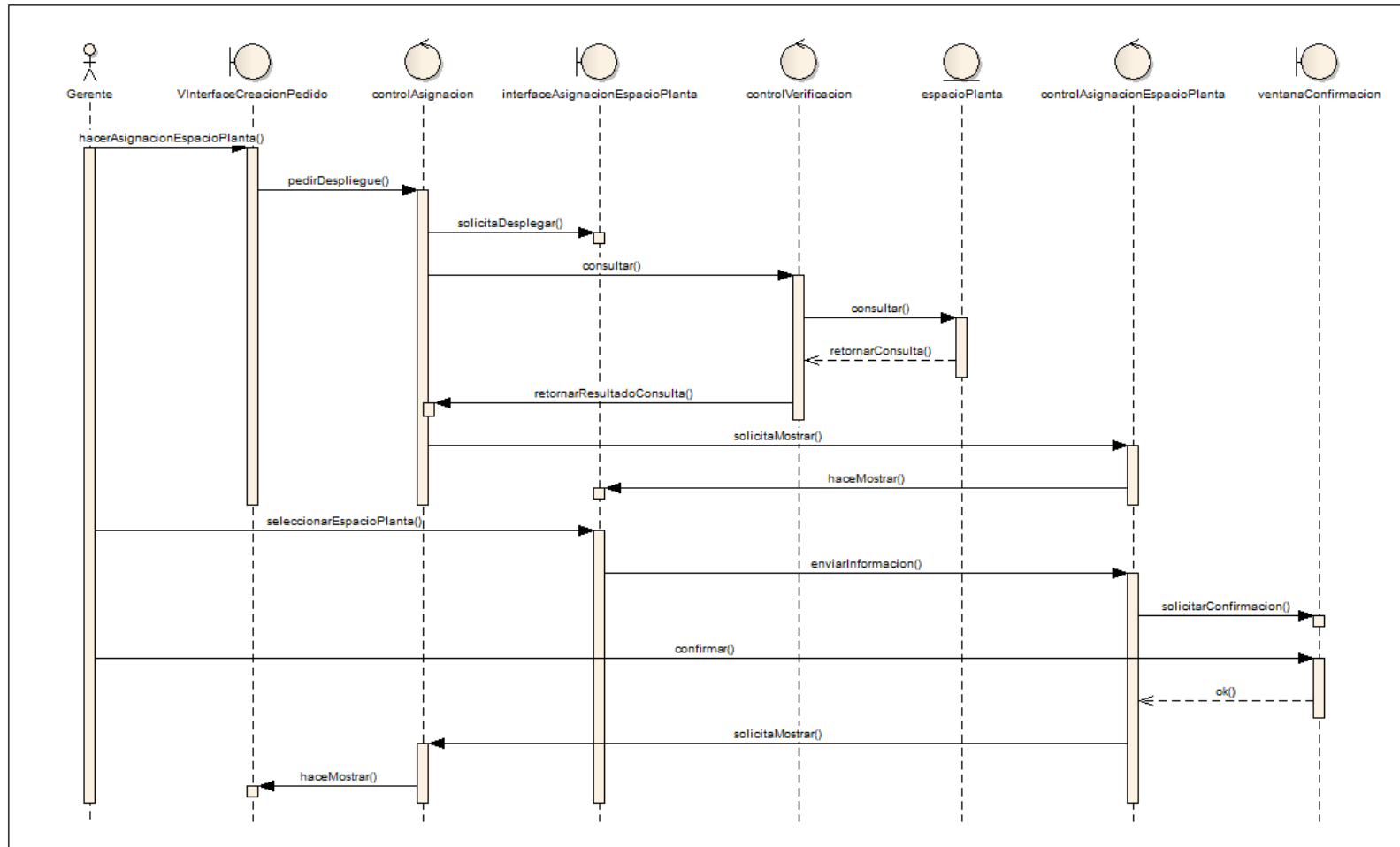
**Tabla 7. Listado de objetos por caso de uso.**

<b>Asignar espacio planta</b>		
<b>Control</b>	<b>Borde</b>	<b>Entidad</b>
controlAsignacion controlAsignacionEspacioPla controlVerificacion	VInterfaceAsignacionEspacioPlant VInterfaceCreacionPedido ventanaConfirmacion	espacioPlanta
<b>Crear pedido</b>		
<b>Control</b>	<b>Borde</b>	<b>Entidad</b>
controladorPrincipal controlAsignacionEspacioPla controlAsignacionMateriaPri controlAsignacionPersonal controlConsultaContrato controlCreacionPedido	VInterfaceCreacionPedido VVentanaValidacionContrato ventanaConfirmacion ventanaMensaje	espacioPlanta materiaPrima pedido personal
<b>Actualizar pedido</b>		
<b>Control</b>	<b>Borde</b>	<b>Entidad</b>
controlActualizacionPedido controlConsultaPedido controladorPrincipal	VFormularioIngresoActualizacion VinterfaceActualizacionPedido ventanaConfirmacion ventanaMensaje	pedido
<b>Consultar actualizaciones</b>		
<b>Control</b>	<b>Borde</b>	<b>Entidad</b>
controlConsultaActualizacion controlConsultaPedido	interfaceConsultaActualizacionPed VPantallaResultadoBusqueda	pedido

### **5.2.2. Diagramas de secuencia con objetos**

Haciendo uso de los objetos previamente identificados, podemos generar los diagramas de secuencia con objetos, los cuales modelan la interacción entre los estos para llevar a cabo cada caso de uso del sistema. Esto será de suma importancia a la hora de realizar una aproximación al comportamiento final del sistema.

Figura 9. Diagrama de secuencia con objetos Asignar espacio planta.



Fuentes: Autores.

Figura 10. Diagrama de secuencia con objetos Actualizar pedido.

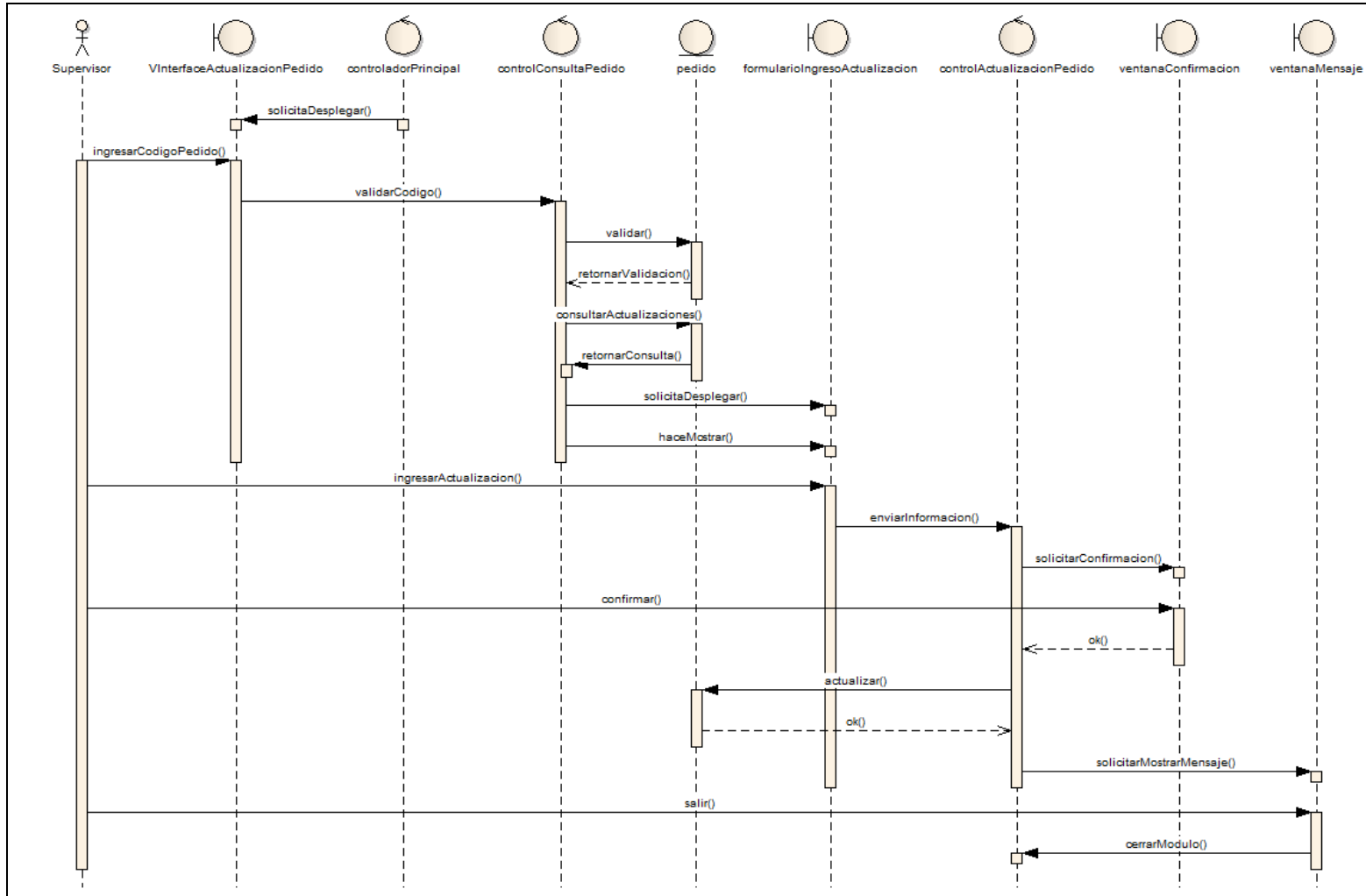
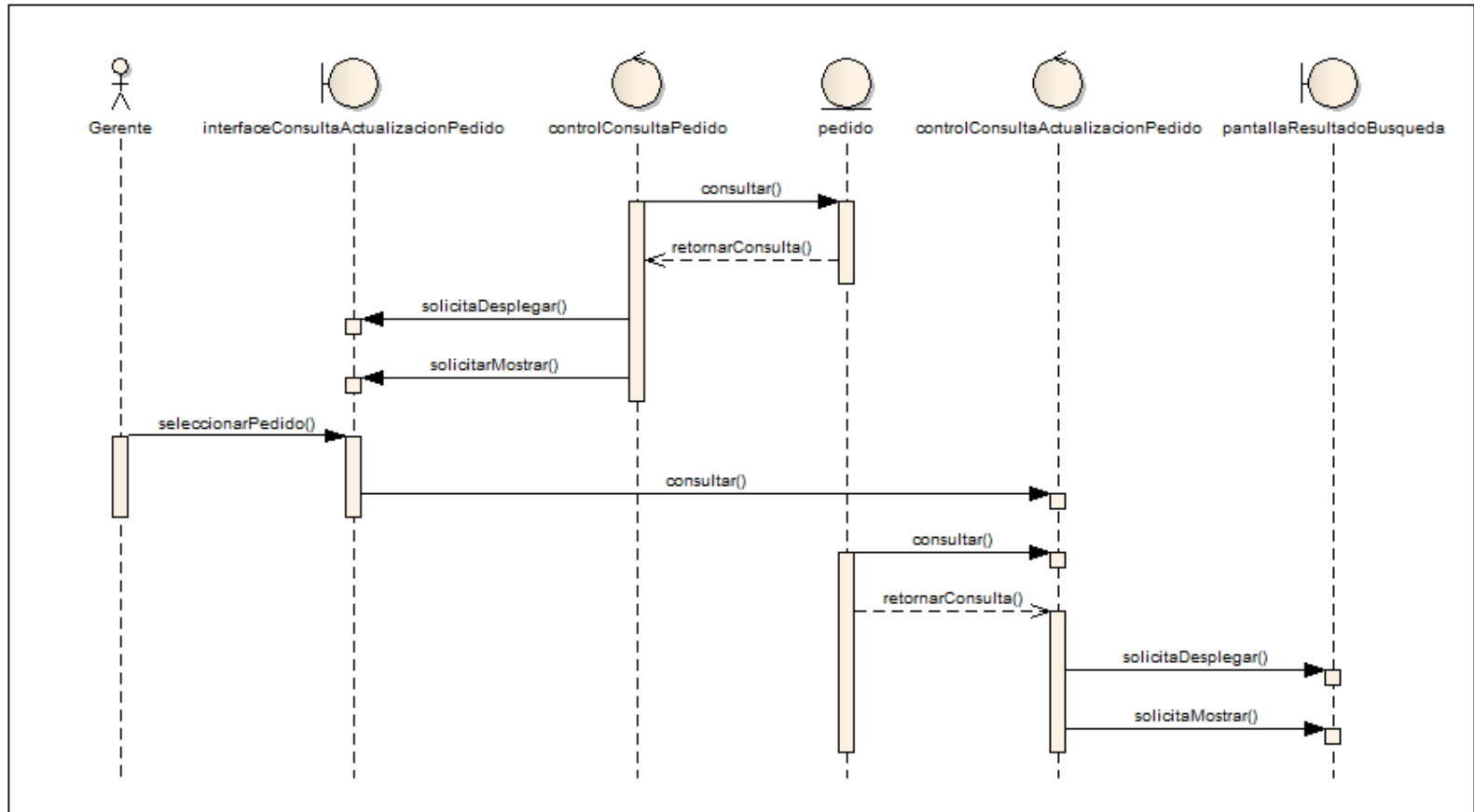


Figura 11. Diagrama de secuencia con objetos Consultar actualización.



### 5.2.3. Narrativas de casos de uso con objetos.

Por medio de estas narrativas se pretende describir en mayor detalle la interacción entre objetos representada en los diagramas de secuencia. Adicionalmente, serán usadas para construir las tarjetas CRC para cada objeto, las cuales serán elaboradas más adelante.

**Tabla 7. Narrativas de casos de uso con objetos.**

Caso de uso	
<b>Caso de uso</b>	Crear pedido
<b>Condición inicial</b>	Debe existir un contrato creado
<b>Flujo principal</b>	El <u>controladorPrincipal</u> <i>solicitaDesplegar</i> la <u>ventanaValidacionContrato</u> sobre la que el <u>gerente</u> podrá <i>ingresarCodigoContrato</i> para iniciar la creación del <u>pedido</u> . Una vez ingresado y enviado el código en la <u>ventanaValidacionContrato</u> esta procede a <i>validarCodigoContrato</i> sobre el <u>controlConsultaContrato</u> el cual realiza <i>validar</i> sobre la entidad <u>contrato</u> quien responde con el evento <i>retornarValidacion</i> lo que permite que pueda <i>consultar</i> la información relacionada con el código ingresado sobre la entidad <u>contrato</u> , el resultado de esta consulta se retorna al control por medio del evento <i>retornarConsulta</i> . Ya con la información el <u>controlConsultaContrato</u> puede <i>exportarInformacion</i> al <u>controlCreacionPedido</u> el cual <i>solicitaDesplegar</i> y <i>solicitaMostrar</i> la información sobre la <u>VInterfaceCreacionPedido</u> , una vez desplegada el gerente podrá <i>asignarRecursos</i> . La información ingresada en la <u>VInterfaceCreacionPedido</u> es enviada al <u>controlCreacionPedido</u> por medio del evento <i>enviarInformacion</i> , para que este pueda <i>solicitarConfirmacion</i> por medio de la <u>ventanaConfirmacion</u> sobre la cual el <u>gerente</u> hará <i>confirmar</i> . Al confirmarse la acción el <u>controlCreacionPedido</u> puede realizar el evento de crear sobre <u>pedido</u> , de igual forma <i>solicitarAsignacion</i> a <u>controlAsignacionEspacioPlanta</u> , <u>controlAsignacionPersonal</u> y <u>controlAsignacionMateriaPrima</u> para que puedan <i>asignar</i> sobre <u>espacioPlanta</u> , <u>personal</u> y <u>materiaPrima</u> ,

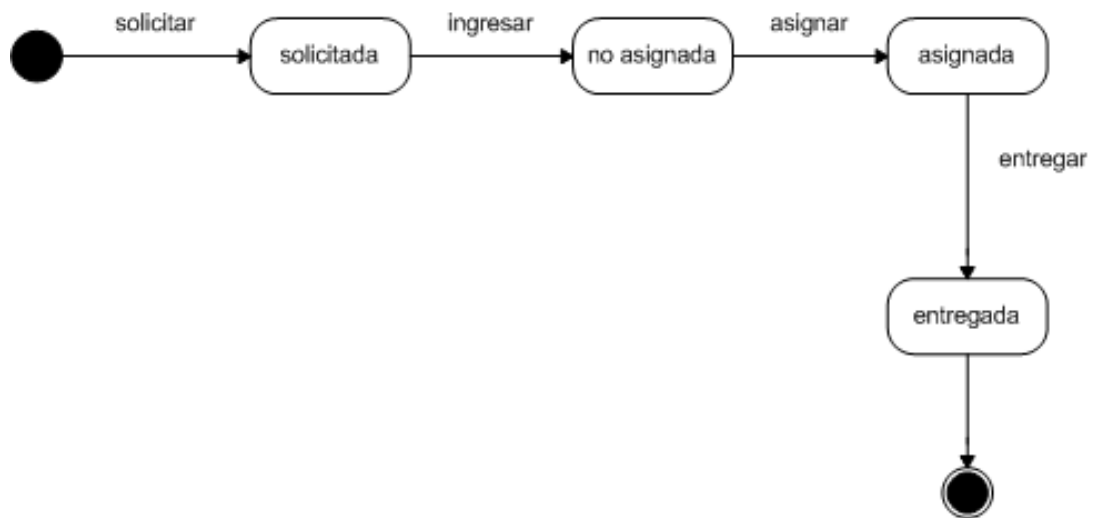
	respectivamente. Una vez realizado esto el <u>controlCreacionPedido</u> puede <i>solicitarMostrarMensaje</i> a la <u>ventanaMensaje</u> para que por medio del evento <i>aceptar</i> pueda visualizar el <u>pedido</u> creado en pantalla.
<b>Condición de salida</b>	Pedido creado
<b>Caso de uso</b>	Asignar espacio planta
<b>Condición inicial</b>	Tener disponibilidad de zonas y estar creando un pedido
<b>Flujo principal</b>	El flujo comienza cuando un nuevo pedido se está creando y la <u>VInterfaceCreacionPedido</u> está desplegada y el <u>gerente</u> realiza <i>hacerAsignacionEspacioPlanta</i> para que la interface haga <i>solicitarDespliegue</i> sobre el <u>controlAsignacion</u> . Este control <i>solicitaDesplegar</i> la <u>VInterfaceAsignacionEspacioPlanta</u> y posteriormente realiza el evento <i>consultar</i> sobre <u>controlVerificacion</u> para que este consulte el listado de zonas sobre <u>espacioPlanta</u> por medio del evento <i>consultar</i> , una vez ha recibido la información realiza el evento <i>retornarResultadoConsulta</i> a <u>controlAsignacion</u> para que este haga el evento <i>solicitaMostrar</i> a <u>controlAsignacionEspacioPlanta</u> quien finalmente muestra el resultado de la consulta sobre la <u>VInterfaceAsignacionEspacioPlanta</u> por medio del evento <i>haceMostrar</i> . El <u>gerente</u> hace la selección zonas a asignar por medio del evento <i>seleccionarEspacioPlanta</i> sobre la <u>VInterfaceAsignacionEspacioPlanta</u> , esta información es enviada al <u>controlAsignacionEspacioPlanta</u> por el evento <i>enviarInformacion</i> para que este realice <i>solicitarConfirmacion</i> sobre <u>ventanaConfirmacion</u> . Por último el <u>gerente</u> hace <i>confirmar</i> sobre <u>ventanaConfirmacion</u> , con esto <u>controlAsignacionEspacioPlanta</u> envía el evento <i>solicitaMostrar</i> a <u>controlAsignacion</u> y esta muestra la selección de zonas sobre la <u>VInterfaceCreacionPedido</u> con el evento <i>haceMostrar</i> .
<b>Condición de salida</b>	Espacio en planta seleccionado para asignar
<b>Caso de uso</b>	Actualizar pedido
<b>Condición inicial</b>	Deben existir pedidos en creados
<b>Flujo principal</b>	El <u>controladorPrincipal</u> <i>solicitaDesplegar</i> la <u>VInterfaceActualizacionPedido</u> para que el <u>supervisor</u>

	<p>pueda <i>ingresarCodigoPedido</i>, seguido a esto la interfaz solicita <i>validarCodigo</i> a <u>controlConsultaPedido</u>, el cual realiza la acción de <i>validar</i> con <u>pedido</u> quien confirma el que el código es válido con el evento <i>retornarValidacion</i>, este evento le permite a <u>ControlConsultaPedido</u> <i>consultarActualizaciones</i> sobre <u>pedido</u>. Cuando <u>controlConsultaPedido</u> recibe el evento <i>retornarConsulta</i> con la información de las actualizaciones sobre el <u>pedido</u> consultado, el control <i>solicitaDesplegar</i> el <u>formularioIngresoActualizacion</u> para mostrar dicha información por medio del evento <i>haceMostrar</i>, con lo que el <u>supervisor</u> puede <i>ingresarActualizacion</i> sobre el formulario, el cual realiza <i>enviarInformacion</i> a <u>controlActualizacionPedido</u> para que este realice <i>solicitarConfirmacion</i> sobre la <u>ventanaConfirmacion</u> en la cual el <u>supervisor</u> realiza el evento <i>confirmar</i>. Una vez confirmada la acción el <u>controlActualizacionPedido</u> procede a <i>actualizar</i> sobre <u>pedido</u> y una vez recibido el evento <i>ok</i> que informa sobre el éxito de la actualización el controlador realiza <i>solicitarMostrarMensaje</i> sobre la <u>ventanaMensaje</u> para que el <u>supervisor</u> realice <i>salir</i> y se le indique al <u>controlActualizacionPedido</u> que puede realizar <i>cerrarModulo</i>.</p>
<b>Condición de salida</b>	El pedido ha sido actualizado
<b>Caso de uso</b>	Consultar actualizaciones
<b>Condición inicial</b>	Deben existir pedidos creados
<b>Flujo principal</b>	<p>El <u>controlConsultaPedido</u> realiza <i>consultar</i> sobre la entidad <u>pedido</u>, posteriormente este <i>solicitaDesplegar</i> la <u>VInterfaceConsultaActualizacionPedido</u> para mostrar el listado de pedidos en ejecución por medio del evento <i>solicitarMostrar</i>. El gerente realiza <i>seleccionarPedido</i> lo que hace que la <u>VInterfaceConsultaActualizacionPedido</u> haga <i>consultar</i> sobre <u>ControlConsultaActualizacionPedido</u>, este realiza <i>consultar</i> sobre <u>pedido</u> quien regresa las actualizaciones hechas al pedido para que el control, después de hacer <i>solicitaDesplegar</i>, muestre el resultado de la consulta en la <u>VPantallaResultadoBusqueda</u> con el evento <i>solicitaMostrar</i>.</p>
<b>Cond. salida</b>	

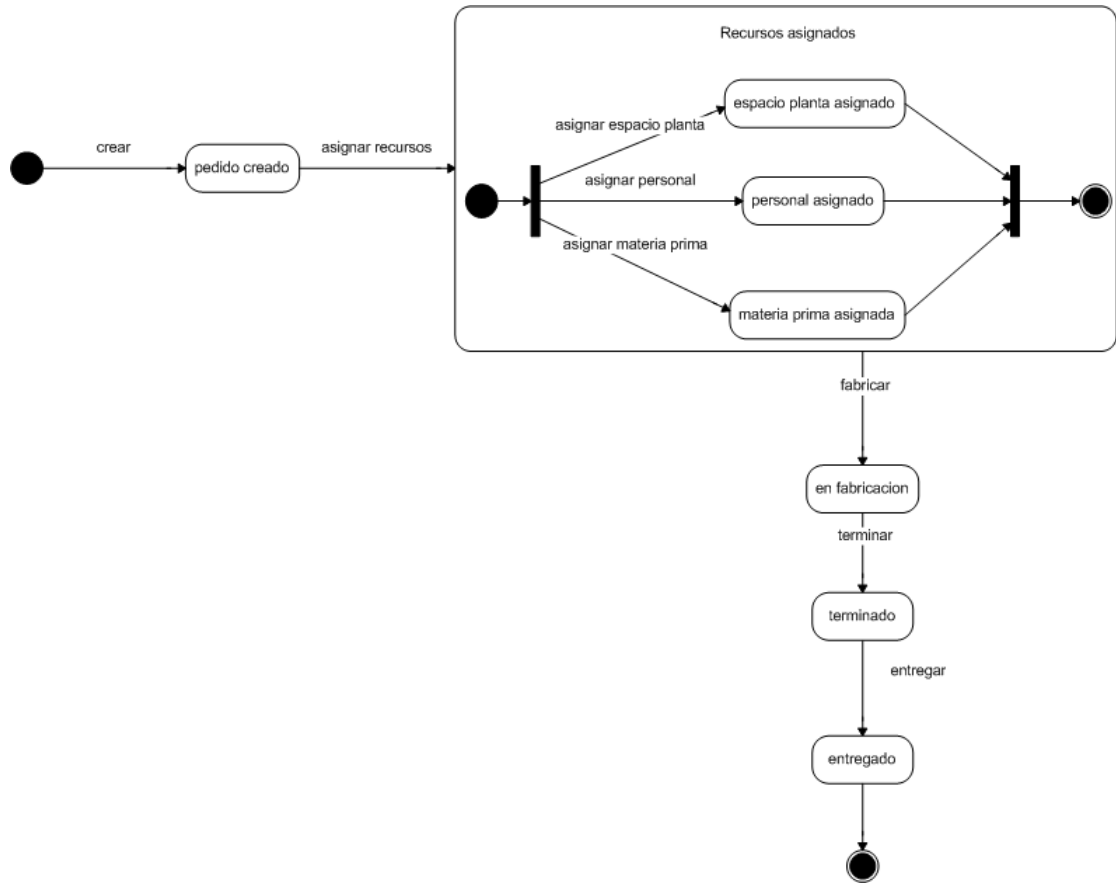
#### 5.2.4. Diagrama de estados

Tienen como fin describir la transición de los objetos por los diferentes eventos y estados, de inicio a fin a lo largo de todo el dominio del sistema. De la lista de objetos, Materia Prima y Pedido fueron seleccionados para esta representación debido a que son los objetos con mayor transición de estados en el sistema.

**Figura 12. Diagrama de estados de Materia Prima**



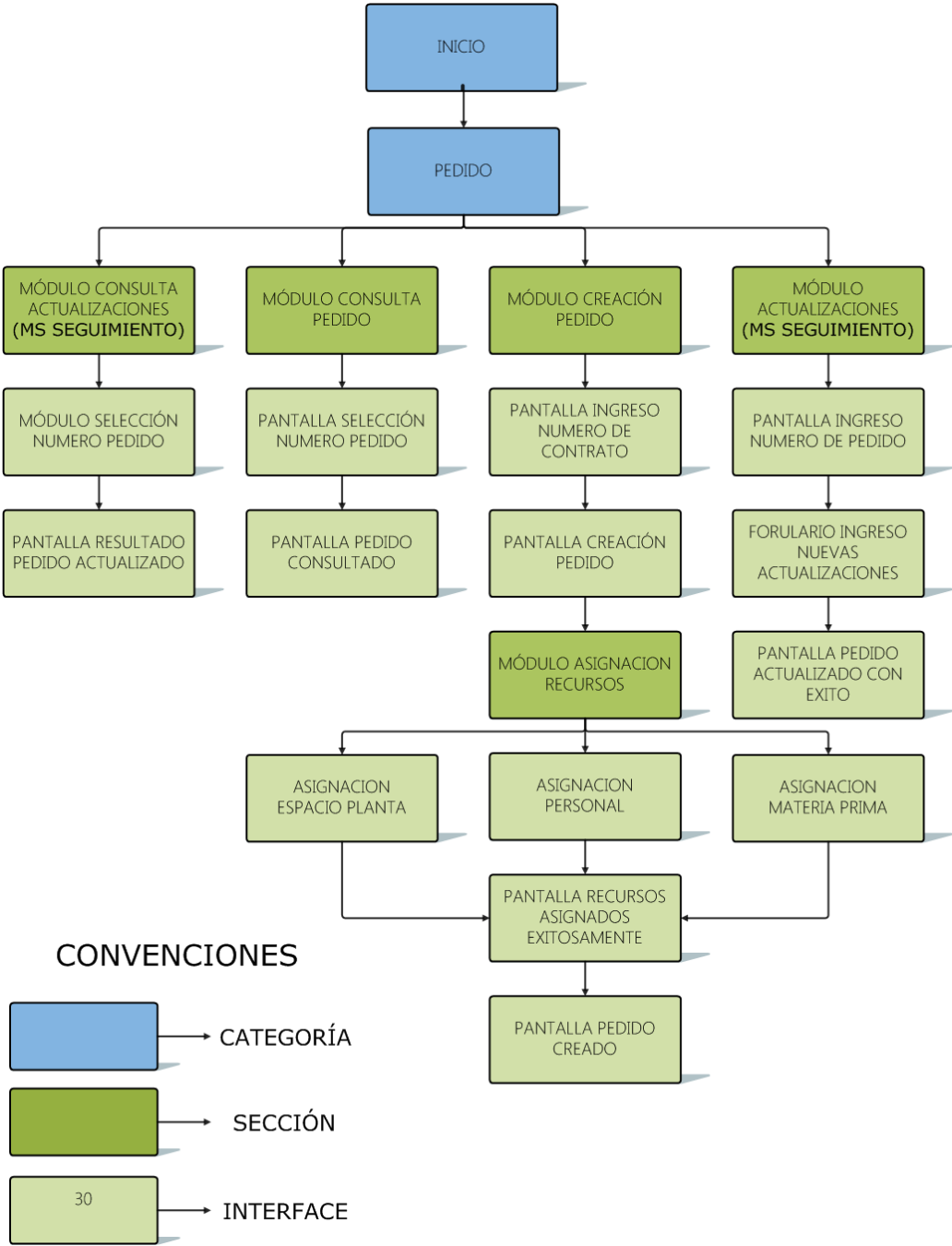
**Figura 13. Diagrama de estados de Pedido**



### 5.2.5. Diagrama de navegación por ventanas.

Tienen como fin brindar una orientación dentro de las distintas interfaces que ofrece la aplicación. Para este caso se decidió agrupar los distintos módulos por secciones para facilidad en su elaboración, lo que nos permite también ver el nivel de profundidad en la navegación a través de la aplicación.

Figura 14. Diagrama de navegación por ventanas para Pedido.

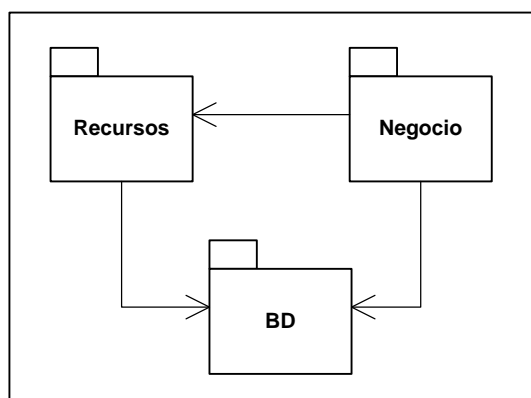


### 5.3. MODELO DE DISEÑO

#### 5.3.1. Subsistemas

A nivel general, el sistema se puede describir en tres subsistemas principales cuya organización y explicación se muestran a continuación. Todos los subsistemas dependen directamente de la base de datos bien sea para guardar información en ella, modificarla, eliminarla o consultarla.

**Figura 15. Diagrama de división por subsistemas, primera capa.**



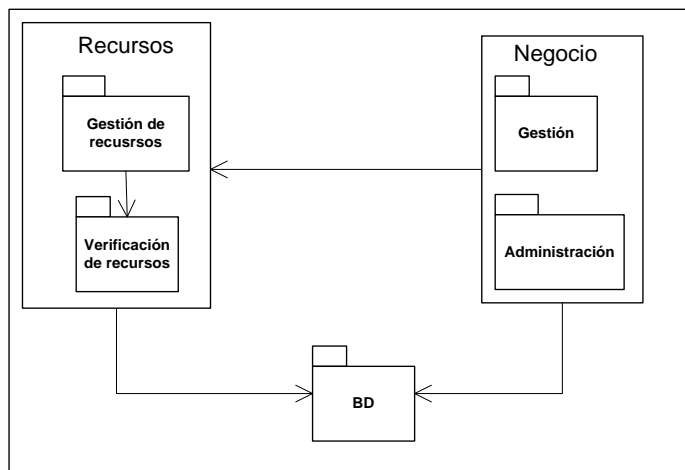
**Tabla 8. Diccionario de subsistemas, primera capa.**

Subsistema	Responsabilidad
Recursos	Es el encargado de generar las consultas para mostrar la disponibilidad de recursos, y en base a esa información realizar la asignación de los mismos según las necesidades del subsistema de Pedido. También tiene como responsabilidad registrar el ingreso de nueva materia prima y la correcta entrega del material asignado para cada pedido. Así mismo es el encargado del manejo del resto de recursos en lo que respecta a modificaciones, eliminaciones y creaciones de los mismos.
Negocio	El subsistema de pedido es el encargado de registrar la creación de cotización, contrato y el pedido, elementos necesarios para registrar la fabricación de un nuevo producto teniendo en la asignación de recursos proporcionada por el

	<p>subsistema de recursos. De igual manera este subsistema debe hacer posible el seguimiento a la fabricación del producto a partir del ingreso y consulta de actualizaciones. Es función de este controlar todo lo relacionado con procesos y productos que se manejen.</p>
--	--

De igual forma, los subsistemas de recursos y negocio se dividen en diferentes subsistemas como se muestra a continuación.

**Figura 16. Diagrama de división por subsistemas, segunda capa.**

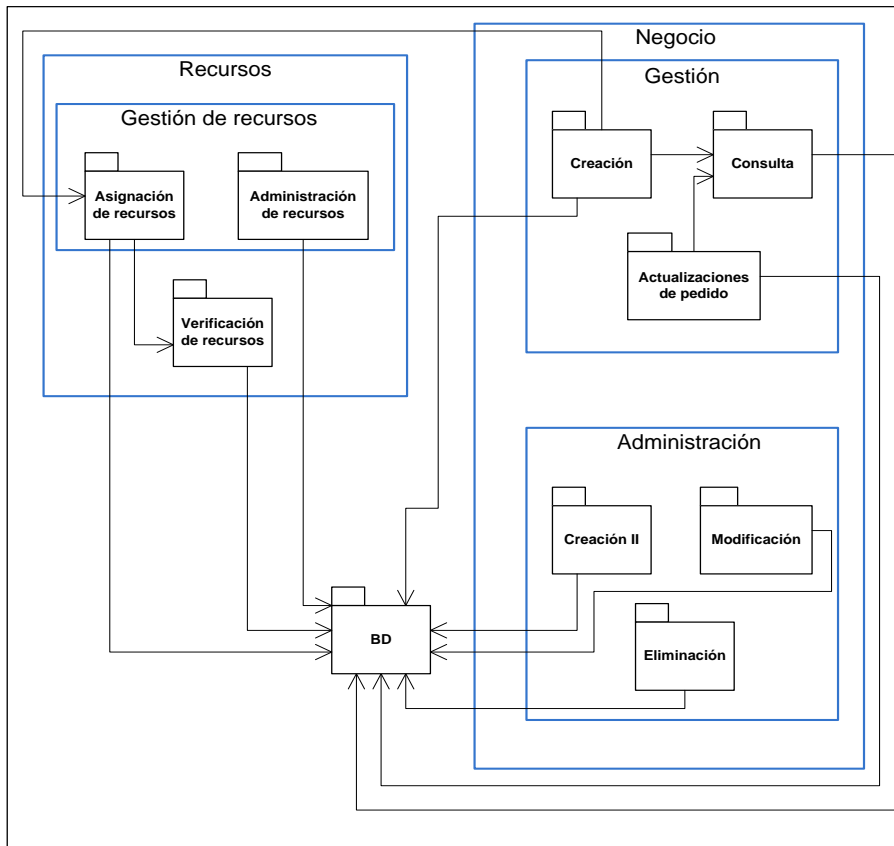


**Tabla 9. Diccionario de subsistemas, segunda capa.**

<b>Subsistema</b>	<b>Responsabilidad</b>
Gestión de recursos	Tiene a cargo lo relacionado con la asignación de recursos, y la creación, modificación y eliminación de nuevos recursos en el sistema.
Verificación de recursos	Es el encargado de desplegar información referente a la disponibilidad de cada tipo de recurso dentro de la organización, información que también es referida al subsistema de asignación para su funcionamiento.
Gestión	Su responsabilidad directa está relacionada con la creación y

	consulta de cotizaciones, contratos y pedidos, además de tener a cargo el ingreso y consulta de actualizaciones de pedido.
Administración	Soporta la creación, modificación y eliminación de los procesos y productos presentes en el sistema

**Figura 17. Diagrama de división por subsistemas, tercera capa.**



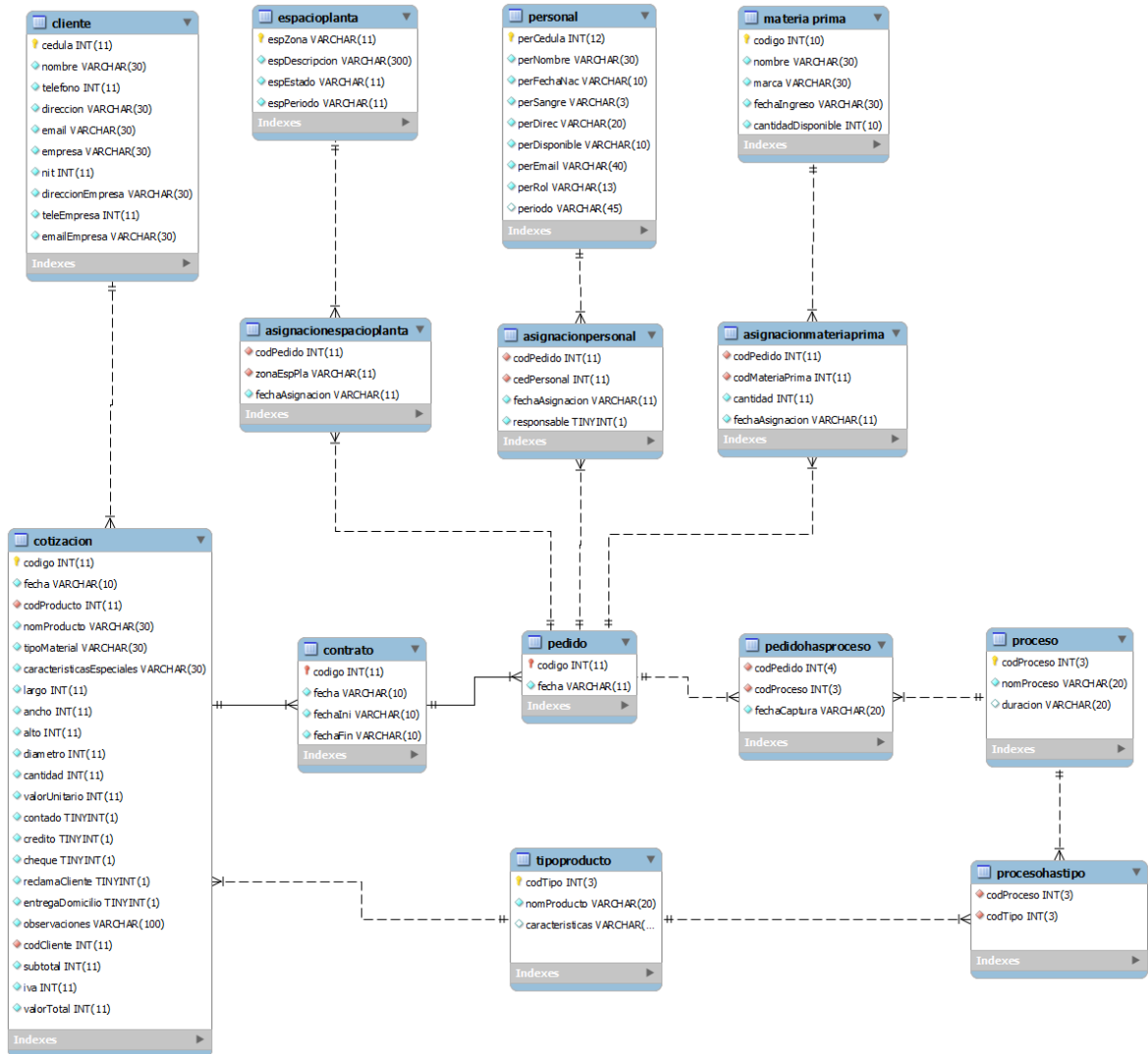
**Tabla 10. Diccionario de subsistemas, tercera capa.**

Subsistema	Responsabilidad
Asignación de recursos	Es el encargado de tomar y responder la solicitud del subsistema de pedido y efectuar la asignación de recursos según lo especificados en la misma.

Administración de recursos	Esta específicamente encargado de las actividades de creación, eliminación y modificación de toda la información referente a los recursos.
Creación	Tiene como responsabilidad tomar toda la información pertinente a la creación de un pedido para más adelante iniciar su fabricación. Este subsistema es el encargado de generar una cotización, un contrato y un pedido, y para este último su respectiva solicitud de asignación de recursos a ese subsistema.
Consulta	Este subsistema debe desplegar la información de todos los pedidos en fabricación, así como de su contrato y cotización respectiva.
Actualizaciones de pedido	Es el encargado de registrar y permitir la consulta de los progresos y el cumplimiento de cada una de las etapas de fabricación del pedido. Esta información sobre el cumplimiento debe permitir medir el progreso de dos formas: específicamente, es decir mostrando cada etapa completada, y generalmente a nivel de porcentajes.
Creación II	Se encarga específicamente de las actividades relacionadas en la creación de información referente a productos y procesos.
Modificación	Se encarga específicamente de las actividades relacionadas en la modificación de información referente a productos y procesos.
Eliminación	Se encarga específicamente de las actividades relacionadas en la modificación de información referente a productos y procesos.

### 5.3.2. Diseño base de datos.

Figura 18. Modelo entidad relación para la base de datos del sistema, msdb



### 5.3.3. Diccionario de Datos del sistema

En el siguiente grupo de tablas se encuentra toda la información de la base de datos usada para soportar la aplicación.

**Tabla 11. Diccionario de datos del sistema**

<b>Nombre de la base de datos: msdb</b>				
<b>Tabla:</b> asignacionespacioplanta		<b>Descripción:</b> reúne las asignaciones de espacio en planta realizadas a todos los pedido		
<b>Columna</b>	<b>Tipo de Dato</b>	<b>PK</b>	<b>No Nulo</b>	<b>Descripción</b>
codPedido	INT(11)		✓	Clave del pedido asignado
zonaEspPla	VARCHAR(11)		✓	Clave de la zona asignada a al pedido
fechaAsignacion	VARCHAR(11)		✓	Fecha en la que se realiza la asignación.
<b>Relaciones:</b> +codPedido con el campo código en pedido +zonaEspPla con el campo espZona en espacioPlanta		<b>Campos Clave:</b> codPedido(FK), zonaEspPla(FK)		
<b>Tabla:</b> Asignacionmateriaprima		<b>Descripción:</b> reúne las asignaciones de materia prima realizadas a todos los pedidos		
<b>Columna</b>	<b>Tipo de Dato</b>	<b>PK</b>	<b>No Nulo</b>	<b>Descripción</b>
codPedido	INT(11)		✓	Clave del pedido asignado
codMateriaPrima	INT(11)		✓	Clave de la materia prima asignada al pedido
cantidad	INT(11)		✓	Cantidad asignada
fechaAsignacion	VARCHAR(11)		✓	Fecha en la que se realiza la asignación
<b>Relaciones:</b> +codPedido con el campo código en pedido. +codMateriaPrima con el campo Código en materia prima		<b>Campos Clave:</b> codPedido(FK), codMateriaPrima(FK)		

<b>Tabla: Asignacionpersonal</b>		<b>Descripción:</b> Reúne las asignaciones de personal realizadas a todos los pedidos		
<b>Columna</b>	<b>Tipo de Dato</b>	<b>PK</b>	<b>No Nulo</b>	<b>Descripción</b>
codPedido	INT(11)		✓	Clave del pedido asignado
cedPersonal	INT(11)		✓	Clave de la persona asignada al pedido
fechaAsignacion	VARCHAR(11)		✓	Fecha en la que se realiza la asignación
responsable	TINYINT(1)		✓	Marca a uno del personal asignado como responsable del pedido
<b>Relaciones:</b> +codPedido con código en la tabla pedido +codPersonal con perCedula en la tabla personal		<b>Campos Clave:</b> codPedido(FK), codPersonal(FK)		
<b>Tabla: cliente</b>		<b>Descripción:</b> Contiene toda la información de los clientes y empresas que han cotizado un producto en la organización		
<b>Columna</b>	<b>Tipo de Dato</b>	<b>PK</b>	<b>No Nulo</b>	<b>Descripción</b>
Cedula	INT(11)	✓	✓	Numero de cedula, identificador del cliente
Nombre	VARCHAR(30)		✓	Nombre del cliente
Teléfono	INT(11)		✓	Teléfono del cliente
Dirección	VARCHAR(30)		✓	Dirección del cliente
Email	VARCHAR(30)		✓	Correo del cliente
Empresa	VARCHAR(30)		✓	Nombre de la empresa
Nit	INT(11)		✓	Identificador de la empresa
direccionEmpresa	VARCHAR(30)		✓	Dirección de la empresa
teleEmpresa	INT(11)		✓	Teléfono de la empresa
emailEmpresa	VARCHAR(30)		✓	Correo de la empresa
<b>Relaciones:</b> +cedula con el campo codCliente en cotización		<b>Campos Clave:</b> cedula		
<b>Tabla: contrato</b>		<b>Descripción:</b> Almacena las fechas que		

		pactan los compromisos de entrega a la hora de hacer negocio con un cliente		
Columna	Tipo de Dato	PK	No Nulo	Descripción
Codigo	INT(11)	✓	✓	Código del contrato
Fecha	VARCHAR(10)		✓	Fecha de creación del contrato
fechaIni	VARCHAR(10)		✓	Fecha en la que se planea iniciar la producción
fechaFin	VARCHAR(10)		✓	Fecha en la que finaliza la producción y se entrega el producto
<b>Relaciones:</b> +código con el campo código en cotización y pedido		<b>Campos Clave:</b> código(FK)		
<b>Tabla:</b> cotizacion		<b>Descripción:</b> Almacena toda la información sobre el producto deseado por el cliente y detalles de la compra, pagos y envíos en caso tal de efectuarse el acuerdo comercial.		
Columna	Tipo de Dato	PK	No Nulo	Descripción
codigo	INT(11)	✓	✓	Código de la cotización
fecha	VARCHAR(10)		✓	Fecha de creación de la cotización
codProducto	INT(11)		✓	Referencia del producto que se está cotizando
nomProducto	VARCHAR(30)		✓	Nombre del producto que se está cotizando
tipoMaterial	VARCHAR(30)		✓	Tipo de material a usar
caracteristicasEspeciales	VARCHAR(30)		✓	Características adicionales del producto
largo	INT(11)		✓	Medida
ancho	INT(11)		✓	Medida
alto	INT(11)		✓	Medida
diametro	INT(11)		✓	Medida
cantidad	INT(11)		✓	Cantidad de unidades a cotizar de ese producto
valorUnitario	INT(11)		✓	Valor por unidad del producto cotizado
contado	TINYINT(1)		✓	Tipo de pago

credito	TINYINT(1)		✓	Tipo de pago
cheque	TINYINT(1)		✓	Tipo de pago
reclamaCliente	TINYINT(1)		✓	Tipo de entrega
entregaDomicilio	TINYINT(1)		✓	Tipo de entrega
observaciones	VARCHAR(99)		✓	Observaciones o comentarios adicionales
codCliente	INT(11)		✓	Cedula del cliente que cotiza
subtotal	INT(11)		✓	Valor subtotal de la cotización
iva	INT(11)		✓	Impuesto
valorTotal	INT(11)		✓	Valor total con impuestos
<b>Relaciones:</b> +codProducto con codTipo en tipoProducto +codCliente con cedula en cliente +codigo con codigo en contrato		<b>Campos Clave:</b> código, codProducto(FK), codCliente(FK)		
<b>Tabla:</b> espacioplanta		<b>Descripción:</b> Almacena toda la información de los espacios dispuestos por la compañía para la realización de sus productos.		
<b>Columna</b>	<b>Tipo de Dato</b>	<b>PK</b>	<b>No Nulo</b>	<b>Descripción</b>
espZona	VARCHAR(11)	✓	✓	Código de la zona
espDescripcion	VARCHAR(300)		✓	Características de la zona
espEstado	VARCHAR(11)		✓	Disponibilidad
espPeriodo	VARCHAR(11)		✓	Periodo de disponibilidad
<b>Relaciones:</b> +espZona con zonaEspPla en asignacionespacioplanta		<b>Campos Clave:</b> espZona		
<b>Tabla:</b> materia prima		<b>Descripción:</b> Almacena toda la información de los materiales e insumos necesarios para la fabricación de los productos; así como también información de la cantidad disponible.		
<b>Columna</b>	<b>Tipo de Dato</b>	<b>PK</b>	<b>No Nulo</b>	<b>Descripción</b>
codigo	INT(10)	✓	✓	Referencia del material

nombre	VARCHAR(30)		✓	Nombre del material
marca	VARCHAR(30)		✓	Marca de la materia prima o insumo
fechaIngreso	VARCHAR(30)		✓	Fecha de ingreso al inventario
cantidadDisponible	INT(10)		✓	Cantidad disponible para asignar
<b>Relaciones:</b> +codigo con codMateriaPrima en asignacionmateriaprima		<b>Campos Clave:</b> codigo		
<b>Tabla:</b> Pedido		<b>Descripción:</b> Registra el momento en que la producción de un producto es puesta en marcha.		
<b>Columna</b>	<b>Tipo de Dato</b>	<b>PK</b>	<b>No Nulo</b>	<b>Descripción</b>
codigo	INT(11)	✓	✓	Codigo del pedido
fecha	VARCHAR(11)		✓	Fecha de creación del pedido
<b>Relaciones:</b> +codigo con codPedido en asignacionespacioplanta, asignacionpersonal, asignacionmateriaprima, +codigo con codigo en contrato +codigo con codPedido en pedidohasproceso		<b>Campos Clave:</b> codigo		
<b>Tabla:</b> pedidohasproceso		<b>Descripción:</b> Es la encargada de registrar todos los procesos completados y no completados (actualizaciones) en la fabricación de cada producto en producción relacionado el mismo con su respectivo número de pedido.		
<b>Columna</b>	<b>Tipo de Dato</b>	<b>PK</b>	<b>No Nulo</b>	<b>Descripción</b>
codPedido	INT(4)		✓	
codProceso	INT(3)		✓	
fechaCaptura	VARCHAR(20)		✓	
<b>Relaciones:</b> +codPedido con codigo en pedido		<b>Campos Clave:</b> codPedido(FK), codProceso(FK)		

+codProceso con codProceso en proceso				
<b>Tabla: Personal</b>		<b>Descripción:</b> Almacena toda la información del capital humano de la empresa, desde el gerente hasta el operario en planta.		
<b>Columna</b>	<b>Tipo de Dato</b>	<b>PK</b>	<b>No Nulo</b>	<b>Descripción</b>
perCedula	INT(12)	✓	✓	Cedula del empleado
perNombre	VARCHAR(30)		✓	Nombre del empleado
perFechaNac	VARCHAR(10)		✓	Fecha de nacimiento
perSangre	VARCHAR(3)		✓	Tipo de sangre
perDirec	VARCHAR(20)		✓	Dirección residencia
perDisponible	VARCHAR(10)		✓	Disponibilidad
perEmail	VARCHAR(40)		✓	Correo electrónico
perRol	VARCHAR(13)		✓	Rol o cargo en la empresa
periodo	VARCHAR(20)			Periodo de disponibilidad
<b>Relaciones:</b> +perCedula con cedPersonal en asignacionpersonal		<b>Campos Clave:</b> perCedula		
<b>Tabla: Proceso</b>		<b>Descripción:</b> Almacena la información de todos los procesos de todos los productos que fabrica en la empresa		
<b>Columna</b>	<b>Tipo de Dato</b>	<b>PK</b>	<b>No Nulo</b>	<b>Descripción</b>
codProceso	INT(3)	✓	✓	Número del proceso
nomProceso	VARCHAR(20)		✓	Nombre del proceso
duracion	VARCHAR(20)			Establece cuánto dura en Tiempo un proceso en completarse
<b>Relaciones:</b> +codProceso con codProceso en procesohastipo		<b>Campos Clave:</b> codProceso		
<b>Tabla: procesohastipo</b>		<b>Descripción:</b> Esta tabla permite clasificar que procesos en la compañía corresponden a cada uno de sus productos.		
<b>Columna</b>	<b>Tipo de Dato</b>	<b>PK</b>	<b>No Nulo</b>	<b>Descripción</b>

codProceso	INT(3)		✓	Codigo de uno de los proceso asociados a un tipo
codTipo	INT(3)		✓	Referencia de producto a la que se le asigna un listado de procesos
<b>Relaciones:</b> +codProceso con codProceso en proceso. +codTipo con codTipo en tipoProducto		<b>Campos Clave:</b> codProceso(FK), codTipo(FK)		
<b>Tabla:</b> tipoproducto		<b>Descripción:</b> Almacena la lista de todos los productos que fabrica la empresa.		
<b>Columna</b>	<b>Tipo de Dato</b>	<b>PK</b>	<b>No Nulo</b>	<b>Descripción</b>
codTipo	INT(3)	✓	✓	Número de la referencia
nomProducto	VARCHAR(20)		✓	Nombre del producto asociado a esa referencia
caracteristicas	VARCHAR(99)			Características de la referencia
<b>Relaciones:</b> +codTipo con CodProducto en cotizacion		<b>Campos Clave:</b> codTipo		

#### 5.3.4. Control de accesos

El sistema cuenta con 3 usuarios: el gerente, el almacenista y el supervisor. El gerente tiene acceso a la mayoría de funciones disponibles con la excepción de las que permiten realizar una actualización de pedido, ingreso y entrega de materia prima; la primera es exclusivamente hecha por el supervisor y es el único que tiene acceso al módulo diseñado para tal fin; y las otras dos son actividades realizadas por el almacenista. A continuación se presenta una tabla que muestra las diferentes acciones que realiza cada uno de los usuarios sobre los objetos de entidad presentes en el sistema:

**Tabla 12. Control de accesos a los usuarios.**

Objetos	Actores		
	Gerente	Almacenista	Supervisor
Cotización	- Creación - Consulta		
Contrato	- Creación - Consulta		
Pedido	- Creación - Consulta - Consulta actualizaciones		- Ingresar actualización
Cliente	- Creación - Modificación - Eliminación		
Espacio Planta	- Asignación - Verificación - Creación - Modificación - Eliminación		
Personal	- Asignación - Verificación - Creación - Modificación - Eliminación		
Materia Prima	- Asignación - Verificación - Eliminación	- Entrega - Ingreso	
Producto	- Creación		

### 5.3.5. Casos de uso de condiciones de frontera.

Este tipo de casos de uso tienen la particularidad de describir situaciones en las que el sistema podría estar expuesto a fallos de comunicación a nivel de bases de datos; se estudian estas condiciones frontera para plantear el manejo de estas situaciones haciendo uso de flujos alternos para el sistema. Esto servirá como base para el futuro diseño de las pruebas unitarias.

**Tabla 13. Casos de uso de condiciones frontera.**

<b>Caso de Uso</b>	Guardar Actualización
<b>Actor(es)</b>	controlActualizaciónPedido, pedido

<b>Propósito</b>	Guardar el registro de actualización en el subsistema de bases de datos manteniendo la relación con cada pedido, garantizando así la persistencia del objeto para futuras interacciones.	
<b>Descripción</b>	Cuando la actualización haya sido realizada debe ser guardada en el subsistema de bases de datos relacional conservando su referencia con el respectivo pedido.	
<b>Precondición(es)</b>	Pedido generado.	
<b>Flujo Principal</b>	<b>Acciones de Actor(es)</b>	<b>Respuestas del Sistema</b>
	1. controlAsignaciónPedido Inicia la comunicación con el subsistema de bases de datos para guardar la actualización.  2. controlActualizaciónPedido hace la inserción del registro conservando la respectiva relación con el pedido.	3. El subsistema de base de datos luego de recibir el ingreso regresa una confirmación de operación exitosa.
<b>Sub-flujos</b>	Paso 3. Si falla la operación de inserción retorna al paso 1.	
<b>Poscondición(es)</b>	Pedido actualizado y actualización persistente.	
<b>Caso de Uso</b>	Guardar Actualización	
<b>Actor(es)</b>	controlActualizaciónPedido, pedido	
<b>Propósito</b>	Guardar el registro de actualización en el subsistema de bases de datos manteniendo la relación con cada pedido, garantizando así la persistencia del objeto para futuras interacciones.	
<b>Descripción</b>	Al momento de realizar la actualización cuando esta ya haya sido realizada debe ser guardada en el subsistema de bases de datos relacional conservando su referencia con el respectivo pedido.	
<b>Precondición</b>	Pedido generado.	

<b>n(es)</b>		
<b>Flujo Principal</b>	<b>Acciones de Actor(es)</b>	<b>Respuestas del Sistema</b>
	<p>1. controlActualizaciónPedido Inicia la comunicación con el subsistema de bases de datos para guardar la actualización.</p> <p>2. controlActualizaciónPedido hace la inserción del registro conservando la respectiva relación con el pedido.</p>	<p>3. El subsistema de base de datos luego de recibir el ingreso regresa una confirmación de operación exitosa.</p>
<b>Sub-flujos</b>	Paso 3. Si falla la operación de inserción retorna al paso 1.	
<b>Poscondición(es)</b>	Pedido actualizado y actualización persistente.	
<b>Caso de Uso</b>	Guardar Asignación de Recursos	
<b>Actor(es)</b>	controlAsignación, recursos	
<b>Propósito</b>	Hacer persistente cada asignación para su posterior consulta o interacción con los componentes de entregaMateriaPrima o consulta de asignación	
<b>Descripción</b>	Al momento de realizar la asignación cuando esta ya haya sido realizada debe ser guardada en el subsistema de bases de datos relacional conservando su referencia con el respectivo pedido.	
<b>Precondición(es)</b>	-	
<b>Flujo Principal</b>	<b>Acciones de Actor(es)</b>	<b>Respuestas del Sistema</b>
	<p>1. controlAsignaciónPedido Inicia la comunicación con el subsistema de bases de datos para guardar la asignación de recursos.</p>	

	2. controlAsignaciónPedido hace la inserción del registro conservando la respectiva relación con el pedido.	3. El subsistema de base de datos luego de recibir el ingreso regresa una confirmación de operación exitosa.
<b>Sub-flujos</b>	Paso 3. Si falla la operación de inserción retorna al paso 1.	
<b>Poscondición(es)</b>	Pedido con asignación de recursos realizada y asignaciones persistentes.	

### 5.3.6. Diseño de objetos

El proceso de diseño de objetos se basa en las actividades realizadas en el modelo de análisis, más específicamente, en la interacción descrita entre los objetos de entidad, control y borde. En esta parte se aprecia cómo se analizan cada uno de los objetos involucrados en determinada actividad; esto se logra haciendo uso de las tarjetas CRC (Clase Responsabilidad Colaboración) y el diseño basado en responsabilidades, en el que se enmarca el comportamiento de los objetos basándose en una comunicación “cliente - servidor” en la cual el objeto que provee un servicio, oficia como servidor y el objeto que lo solicita hace las veces de cliente.

Como parte inicial del proceso se asigna una tarjeta CRC para cada objeto la cual contendrá los eventos realizados por este y el objeto colaborador de dicho evento, si lo hay. Una vez hechas todas las tarjetas CRC se procede a crear las superclases identificando grupos de objetos que prestan servicios comunes

**Tabla 13. Tarjetas CRC del sistema.**

<b>Objeto:</b>	VInterfaceActualizacionPedido
<b>Descripción:</b>	Recibe el código del pedido a actualizar
<b>Módulo:</b>	-
<b>Estereotipo:</b>	Borde
<b>Propiedades:</b>	Concreta
<b>Superclases:</b>	pantallaValidacionCodigo
<b>Subclases:</b>	-

<b>Atributos:</b>	-
<b>1. Desplegar Pantalla</b>	
Desplegar()	
<b>5. Validar código</b>	
validarCodigo()	<u>controlConsultaPedido</u>
getObjIntActualizaPedido() Devuelve VInterfaceActualizacionPedido	
<b>4. Cerrar Módulo</b>	
enviaCerrar() devuelve void	
getOpcion() devuelve int	
<b>Tarjetas CRC</b>	
<b>Objeto:</b>	VFormularioIngresoActualizacion
<b>Descripción:</b>	Recibe la actualización del pedido seleccionado
<b>Módulo:</b>	-
<b>Estereotipo:</b>	Borde
<b>Propiedades:</b>	Concreta
<b>Superclases:</b>	pantallaValidacion
<b>Subclases:</b>	-
<b>Atributos:</b>	CheckBoxes : ArrayList<JCheckBox>, Labels : ArrayList<JLabel>, Labels3 : ArrayList<JLabel> l = int, jcheck : JCheckBox[], objCtrlActualizaPedido : ControlActualizaciónPedido
<b>1. Desplegar Pantalla</b>	
Desplegar() devuelve void	
<b>9. Validar Campos</b>	
validarCampos() devuelve void	
<b>3. Solicitar Confirmación</b>	
enviarInformación() devuelve void	<u>controlActualizacionPedido</u>
<b>4. Cerrar Módulo</b>	
cerrarInterface() devuelve void	
getObjFormActualización() devuelve VFormularioIngresoActualización	
<b>Tarjetas CRC</b>	
<b>Objeto:</b>	controladorPrincipal
<b>Descripción:</b>	Controlador encargado de solicitar el despliegue de la mayoría de interfaces y realizar la comunicación entre controladores especializados en una actividad
<b>Módulo:</b>	
<b>Estereotipo:</b>	Control
<b>Propiedades:</b>	Concreto
<b>Superclases:</b>	controlDespliegue

<b>Subclases:</b>	
<b>Atributos:</b>	
<b>1. Desplegar pantalla</b>	
solicitaDesplegar(VinterfaceActualizacionPedido) devuelve void	VinterfaceActualizaciónPedido
solicitaDesplegar(interfacelInicio) devuelve void	interfacelInicio
solicitaDesplegar(VPantallaValidacionCodigoPedido) devuelve void	VPantallaValidacionCodigoPedido
solicitaDesplegar(VinterfaceIngresoMatPrima ) devuelve void	VinterfaceIngresoMatPrima
solicitaDesplegar(VPedido) devuelve void	VPedido
solicitaDesplegar(VInterfaceAdministracion) devuelve void	VInterfaceAdministracion
solicitaDesplegar(VinterfaceVerificacionRecursosDisponibles) devuelve void	VinterfaceVerificacionRecursosDisponibles
solicitaDesplegar(InterfaceSuper) devuelve void	InterfaceSuper
solicitaDesplegar(InterfaceGerente) devuelve void	InterfaceGerente
solicitaDesplegar(InterfaceAlmacenista) devuelve void	InterfaceAlmacenista
solicitaDesplegar(VInterfaceEliminacion) devuelve void	VInterfaceEliminacion
solicitaDesplegar(VInterfaceCreacion) devuelve void	VInterfaceCreacion
solicitaDesplegar(VInterfaceModificacion) devuelve void	VInterfaceModificacion
solicitaDesplegar(VPantallaMostrarContrato) devuelve void	VPantallaMostrarContrato
solicitaDesplegar(VPantallaMostrarCotizacion) devuelve void	VPantallaMostrarCotizacion
solicitaDesplegar(VPantallaMostrarPedido) devuelve void	VPantallaMostrarPedido
solicitaDesplegar(VinterfaceCreacionProceso) devuelve void	VinterfaceCreacionProceso
solicitaDesplegar(VinterfaceCreacionEspacioPlanta)	VinterfaceCreaciónEspacioPlanta
solicitaDesplegar(VinterfaceCrecionPersonal) devuelve void	VinterfaceCreaciónPersonal
solicitaDesplegar(VinterfaceConsultaContrato) devuelve void	VinterfaceConsultaContrato

solicitaDesplegar(VinterfaceConsultaCotizacion) devuelve void	VinterfaceConsultaCotizacion
solicitaDesplegar(VinterfaceConsultaPedido) devuelve void	VinterfaceConsultaPedido
solicitaDesplegar(interfaceConsultaActualizacionPedido) devuelve void	interfaceConsultaActualizacionPedido
solicitaDesplegar(interfaceEntregaMateriaPrima) devuelve void	interfaceEntregaMateriaPrima
solicitaDesplegar(interfaceEntregaMateriaPrima) devuelve void	
solicitaDesplegar(interfaceCreacionCotizacion) devuelve void	interfaceCreacionCotizacion
solicitaDesplegar(VVentanaValidacionCotizacion) devuelve void	VVentanaValidacionCotizacion
solicitaDesplegar(VVentanaValidacionContrato) devuelve void	VVentanaValidaciónContrato
crearNumeroCotizacion() devuelve int	
getFechaActual() devuelve String	
getIbjContPrinc() devuelve ControladorPrincipal	
<b>Tarjetas CRC</b>	
<b>Objeto:</b>	controlConsultaPedido
<b>Descripción:</b>	Objeto encargado de controlar todas las consultas hechas a un pedido, así como de la validación de códigos de este
<b>Módulo:</b>	
<b>Estereotipo:</b>	Control
<b>Propiedades:</b>	Concreta
<b>Superclases:</b>	controlValidacion
<b>Subclases:</b>	
<b>Atributos:</b>	Opcion (int) contConsPed (ControlConsultaPedido) objIntActualizaPedido (VInterfaceActualizacionPedido) objPed (Pedido) objProceso (Proceso)
<b>7. Consultar</b>	
consultar(int codigo) devuelve ArrayList	pedido
consultar(int codigo) devuelve ArrayList	pedido
<b>1. Desplegar Pantalla</b>	
solicitaDesplegar(VInterfaceConsultaActualizacionPedido) devuelve void	VInterfaceConsultaActualizacionPedido

solicitaDesplegar(VFormularioIngresoActualizacion) devuelve void	VFormularioIngresoActualizacion
<b>2. Mostrar Información</b>	
solicitarMostrar(ArrayList datos) devuelve void	VInterfaceConsultaActualizacion Pedido
solicitarMostrar(ArrayList datos) devuelve void	VFormularioIngresoActualizacion
solicitarMostrar(ArrayList datos) devuelve void	VInterfaceConsultaPedido
<b>5. validar Código</b>	
validar(int codigo) devuelve void	pedido
getContConsPed() devuelve ControlConsultaPedido	
paginarArray(ArrayList datos) devuelve void	
<b>Tarjetas CRC</b>	
<b>Objeto:</b>	controlActualizacionPedido
<b>Descripción:</b>	Objeto encargado de controlar el proceso de actualización de actividades en un pedido
<b>Módulo:</b>	
<b>Estereotipo:</b>	Control
<b>Propiedades:</b>	Concreta
<b>Superclases:</b>	controlMostrarMensaje, controlDespliegue
<b>Subclases:</b>	
<b>Atributos:</b>	datosA (ArrayList) objCtrlActualizaPedido (ControlActualizacionPedido) objCtrlConsultaPedido(ControlConsultaPedido) objPedido (Pedido)
<b>4. Cerrar Módulo</b>	
cerrarModulo() devuelve void	VFormularioIngresoActualizacion
getObjCtrlActualizaPedido() devuelve controlActualizacionPedido	
<b>6. Mostrar Mensaje</b>	
solicitaMostrarMensaje() devuelve void	ventanaMensaje
<b>3. Solicitar confirmación</b>	
solicitarConfirmacion(ArrayList datos) devuelve void	ventanaConfirmación
actualizarPedido(ArrayList datosA)	pedido
<b>Tarjetas CRC</b>	
<b>Objeto:</b>	Pedido

<b>Descripción:</b>	Contiene toda la información pertinente a los pedidos creados en el sistema	
<b>Módulo:</b>		
<b>Estereotipo:</b>	Entidad	
<b>Propiedades:</b>	Concreta	
<b>Superclases:</b>	Datos	
<b>Subclases:</b>		
<b>Atributos:</b>	codigoPedido (int) con (Conectar) datos (ArrayList) pedido (Pedido) reg (Connection) sql (String)	
<b>7. Consultar</b>		
retornarConsulta(int codigo) devuelve ArrayList	controlConsultaPedido	
<b>8. Crear</b>		
retornarCreacion(ArrayList datos) devuelve boolean	controlCreaciónPedido	
<b>5. Validar Código</b>		
retornarValidacion(int codigo) devuelve boolean	controlConsultaPedido	
actualizarPedido(ArrayList datos) devuelve boolean		
consultarActualizacion(int codigo) devuelve ArrayList		
consultarTipoProducto(int codigo) devuelve int		
generarActualizacion(ArrayList datos) devuelve void		
getCodigoPedido() devuelve int		
getPedido() devuelve Pedido		
validacionActualizacion(int codigo) devuelve boolean		

#### **5.4. Modelo de Pruebas.**

En esta sección se mostrarán los resultados de las pruebas unitarias hechas al grupo de clases que tienen comunicación directa con la base de datos y con aquellas clases que son encargadas de mostrar información producto de una consulta. Cabe mencionar que dichas pruebas fueron dirigidas principalmente a las clases cuyos métodos trabajaban sobre los casos de uso en condiciones frontera. Adicionalmente se tuvieron en cuenta otros métodos los cuales tienen como característica tener la misma estructura general y estar presentes en muchas clases, por lo que se escogieron algunos ejemplares para realizar la prueba y dependiendo de los resultados aplicar correctivos (si los hay) en el resto de clases que los utilizan.

**Tabla 14. Resultados pruebas de unidad.**

Clase / método	Prueba	Resultado esperado	Aprobada	Correcciones
<p>Cotizacion</p> <p>Método: retornarCreacion (ArrayList datos) devuelve boolean.</p>	<p>Se le pasará como parámetro al método un ArrayList nulo.</p>	<p>El método debe devolver "false" ya que sólo devuelve "true" cuando el ArrayList tiene la longitud especificada o no es nulo.</p>	<p>No</p>	<p>Capturar la excepción "NullPointerException" para que en los casos en los que se ingrese un ArrayList nulo devuelva "false" y se vuelva a ingresar el array. Aplicar corrección a todos los métodos retornarCreacion de las clases del paquete Datos.</p>
<p>Cotizacion</p> <p>Método: retornarCreacion (ArrayList datos) devuelve boolean.</p>	<p>Se le pasará como parámetro al método un ArrayList de longitud mayor al esperado.</p>	<p>El método debe devolver "false" ya que sólo devuelve "true" cuando el ArrayList tiene la longitud especificada especificada o no es nulo.</p>	<p>Si</p>	

<p>Cotizacion</p> <p>Método: retornarCreacion (ArrayList datos) devuelve boolean.</p>	<p>Se le pasará como parámetro al método un ArrayList de longitud menor al esperado.</p>	<p>El método debe devolver "false" ya que sólo devuelve "true" cuando el ArrayList tiene la longitud especificada especificada o no es nulo.</p>	<p>Si</p>	
<p>Cotizacion</p> <p>Método: retornarValidacion(intCodigo) devuelve boolean</p>	<p>Se pasará como parámetro un número entero negativo.</p>	<p>El método debe devolver "false" ya que los códigos deben ser positivos.</p>	<p>Si</p>	
<p>Cliente</p> <p>Método: modificacion(ArrayList datosC) devuelve boolean</p>	<p>Se le pasará como parámetro al método un ArrayList nulo.</p>	<p>El método debe devolver "false" ya que sólo devuelve "true" cuando el ArrayList tiene la longitud especificada especificada o no es nulo.</p>	<p>No</p>	<p>Capturar la excepción "NullPointerException" para que en los casos en los que se ingrese un ArrayList nulo devuelva "false" y se vuelva a ingresar el array. Aplicar corrección a todos los métodos de modificación de registros de las clases del paquete Datos.</p>

<p>Pedido</p> <p>Método:</p> <p>actualizarPedido(ArrayList datos)</p>	<p>Se le pasará como parámetro al método un ArrayList nulo.</p>	<p>El método debe devolver "false" ya que solo devuelve true cuando el ArrayList no es nulo.</p>	<p>No</p>	<p>Capturar la excepción "NullPointerException" para que en los casos en los que se ingrese un ArrayList nulo devuelva "false" y se vuelva a ingresar el array.</p>
<p>EspacioPlanta</p> <p>Método:</p> <p>eliminar(ArrayList datosE)</p>	<p>Se le pasará como parámetro al método un ArrayList nulo.</p>	<p>El método debe devolver "false" ya que solo devuelve true cuando el ArrayList no es nulo.</p>	<p>No</p>	<p>Capturar la excepción "NullPointerException" para que en los casos en los que se ingrese un ArrayList nulo devuelva "false" y se vuelva a ingresar el array.</p>
<p>EspacioPlanta</p> <p>Método:</p> <p>Asignar(ArrayList zona)</p>	<p>Se le pasará como parámetro al método un ArrayList nulo.</p>	<p>El método debe devolver "false" ya que solo devuelve true cuando el ArrayList no es nulo.</p>	<p>No</p>	<p>Capturar la excepción "NullPointerException" para que en los casos en los que se ingrese un ArrayList nulo devuelva "false" y se vuelva a ingresar el array.</p>

<p>VInterfaceConsulta ActualizacionPedido Método:</p> <p>haceMostrar(ArrayList datos, int opción) devuelve void</p>	<p>Se le pasará como parámetro un ArrayList vacío.</p>	<p>El método debe funcionar sin errores.</p>	<p>No</p>	<p>Controlar el caso en el que el método recibe un ArrayList vacío para que no se produzcan errores, así mismo replicar este control en todos las clases del paquete Ventana que implementen el método haceMostrar.</p>
<p>VInterfaceConsulta ActualizacionPedido Método:</p> <p>haceMostrar(ArrayList datos, int opción) devuelve void</p>	<p>Se le pasará al método un ArrayList nulo</p>	<p>El método debe funcionar sin errores</p>	<p>No</p>	<p>Capturar la excepción "NullPointerException" para que en los casos en los que el método reciba un ArrayList la excepción se maneje . Aplicar corrección a todos los métodos hacenMostrar implementados en las clases del paquete Ventana.</p>

## 6. CONCLUSIONES.

- Gracias a la aplicación de la arquitectura MVC se logró llevar a cabo un desarrollo más sencillo de la aplicación ya que esta permite implementar independientemente la forma en la que se presenta, se controla, se consulta y almacena la información.
- Se diseñó una aplicación software que permite realizar labores de seguimiento, planeación y programación de actividades en pequeñas empresas cuya producción está dada por una cadena de manufactura.
- El hacer uso de una metodología de desarrollo iterativo permitió la fácil detección y corrección de errores, así como la aplicación de cambios en el software antes de que esta fuera implementada.
- La realización de pruebas de unidad permitieron verificar el correcto funcionamiento de la aplicación y así mismo detectar fallos para su respectiva corrección.
- A través de visitas a empresas locales se lograron levantar una serie de requerimientos basados en las características de un proceso de manufactura, que permitieron establecer un modelo de seguimiento a los procesos, el cual dio origen a un módulo adaptable que sirve para hacer dicho seguimiento a los procesos y tareas completadas para cada uno de los productos que se fabriquen en la empresa.
- Se elaboró un módulo que permite la asignación ordenada de recursos según su disponibilidad al momento de poner en firme la fabricación de un nuevo producto.

## 7. RECOMENDACIONES.

- Implementar funcionalidades que brinden información y permitan controlar temas relacionados con costos de manufactura que estén ligados a la materia prima, el personal, el espacio en plata y las demoras en el cumplimiento de las etapas en el proceso de fabricación.
- Crear una funcionalidad que permita generar archivos con la información ofrecida por los diferentes módulos presentes en la aplicación, tales como el módulo de cotización, contrato, pedido, y actualización.
- Crear un módulo en la herramienta para la gestión y manejo de la información de los clientes que han tenido algún tipo de relación con la compañía.
- Se propone la migración de la aplicación a plataformas web y móviles con el fin de mejorar su accesibilidad para los usuarios al adoptar las múltiples ventajas de estos entornos.
- Como función adicional a la creación del pedido se propone Incluir una función de reprogramación de prioridades que permita para los pedidos en fabricación teniendo en cuenta la asignación y cambio de prioridades en los mismos
- Facilitar a los usuarios opciones de personalización de la aplicación a nivel de formularios y despliegue de información todo en aras de aumentar la adaptabilidad de la aplicación.

## BIBLIOGRAFÍA.

BARNES, David. "Programación orientada a objetos con java: Una introducción práctica usando BlueJ" [Pearson][2007].

CHASE, Richard; AQUILANO, Nicholas y JACOBS, F. Robert. "Administración de Producción y Operaciones" [Mc Graw Hill][2001].

RIGGS, James L. "Sistemas de producción: Planeación, análisis y control". [Limusa Wiley][2008]

WEITZENFELD, Alfredo "Ingeniería de Software Orientada a Objetos: Teoría y práctica con UML y JAVA" [Departamento académico de computación del instituto tecnológico autónomo de México][2002]

ACTIVA SISTEMAS. OpenXpertya ERP completo de licencia libre [en línea]. [Sevilla: España] Activa Sistemas, 2013. Disponible en Internet: [http://www.google.com.co/url?sa=t&rct=j&q=&esrc=s&source=web&cd=5&cad=rja&uact=8&ved=0CDMQFjAE&url=http%3A%2F%2Fwww.activasistemas.com%2Ffilea%2Fmedia%2Fdocs%2FMaterial\\_promocional%2FOXP\\_Dossier\\_Activa%2252Sistemas.pdf&ei=qfj9U8zSFsGvogTjm4G4Dw&usg=AFQjCNEEx\\_5nnBGznNylRnnBG9uVqweVldw&bvm=bv.74035653,d.eXY](http://www.google.com.co/url?sa=t&rct=j&q=&esrc=s&source=web&cd=5&cad=rja&uact=8&ved=0CDMQFjAE&url=http%3A%2F%2Fwww.activasistemas.com%2Ffilea%2Fmedia%2Fdocs%2FMaterial_promocional%2FOXP_Dossier_Activa%2252Sistemas.pdf&ei=qfj9U8zSFsGvogTjm4G4Dw&usg=AFQjCNEEx_5nnBGznNylRnnBG9uVqweVldw&bvm=bv.74035653,d.eXY)

MySQL. La API C de MySQL, [en línea]. [Estados Unidos] Oracle, 2011 [citado 10 ago 2014] Disponible en Internet: <http://dev.mysql.com/doc/refman/5.0/es/c.html>

OPEN BRAVO. Características de la Plataforma ERP de Openbravo, [en línea]. [Pamplona, España] Openbravo SL, 2014 [citado oct. 2013]. Disponible en Internet: <http://www.openbravo.com/es/business-processes-master-data-management>

ORACLE. Java™ Platform, Standard Edition 7-API Specification, [en línea]. [Estados Unidos] Oracle, 2014 [citado 10 ago. 2014] Disponible en Internet: <http://docs.oracle.com/javase/7/docs/api/>

SISTEMAS ERP. Marco Conceptual de APS, [en línea], Google Sites. Nov. 2012 [citado 21 jul. 2014] Disponible en Internet: <https://sites.google.com/site/sistemaserp1/7-1-marco-conceptual-de-aps>

## ANEXOS

### A. Narrativas de casos de uso

<b>Caso de uso</b>	Crear cotización	
<b>Actor(es)</b>	Gerente, cliente	
<b>Propósito</b>	Reunir en un formato las características del producto especificadas por el cliente para tener un sustento claro que permita generar el contrato.	
<b>Descripción</b>	La cotización se genera cuando el cliente le especifica al gerente los detalles del producto que quiere comprar, el gerente valiéndose de un módulo en el sistema toma registro de dichos detalles y genera la cotización.	
<b>Precondición(es)</b>		
<b>Flujo Principal</b>	<b>Acción de actor(es)</b>	<b>Respuesta(s) del sistema</b>
	1. El gerente ingresa al módulo de creación de cotización.  3. El gerente ingresa todos los datos de la cotización y pulsa el botón crear.  5. El gerente confirma la operación.	2. El sistema despliega el formulario de creación de cotización con un número de cotización generado según la secuencia.  4. El sistema valida los campos del formulario y solicita confirmación del envío del formulario.  6. El sistema confirma la creación de la cotización por medio de un mensaje mostrado en pantalla y muestra la información de la cotización recién creada.
<b>Sub-flujo(s)</b>	1. Si al pulsar el botón crear hay campos sin diligenciar muestra mensaje solicitando al usuario llenar los campos vacíos, regresa al paso 3.	
<b>Pos-condición(es)</b>	La cotización ha sido creada.	
<b>Caso de uso</b>	Crear contrato	
<b>Actor(es)</b>	Gerente, cliente	
<b>Propósito</b>	Reunir en un formulario toda la información relacionada con lo solicitado por el cliente, teniendo en cuenta la información indexada en la cotización y adicionalmente	

	los datos de cliente, costos de operación, compromisos de entrega entre otra información pertinente a un contrato comercial.	
<b>Descripción</b>	Inicia cuando el gerente ingresa al módulo de creación de contrato y selecciona la opción crear contrato, allí se relacionará la cotización existente del cliente en cuestión entre otra información como las fechas de entrega.	
<b>Precondición(es)</b>	Debe existir una cotización para la creación de su respectivo contrato.	
<b>Flujo Principal</b>	<b>Acción de actor(es)</b>	<b>Respuesta(s) del sistema</b>
	<p>1. El gerente ingresa al módulo de creación de contrato.</p> <p>3. El gerente ingresa un número de cotización para la creación de contrato y pulsa aceptar.</p> <p>5. El gerente selecciona las fechas de inicio y de finalización de fabricación y pulsa el botón crear.</p> <p>7. El gerente confirma la operación.</p>	<p>2. El sistema despliega la ventana de validación de código de cotización</p> <p>4. El sistema valida el código ingresado y despliega el formulario de creación de contrato el cual contiene toda la información ingresada en la cotización.</p> <p>6. El sistema valida los campos del formulario y solicita confirmación del envío del formulario.</p> <p>8. El sistema confirma la creación del contrato por medio de un mensaje mostrado en pantalla y muestra la información del contrato recién creado.</p>
<b>Sub-flujos</b>	<p>1. Código de cotización no existe, regresa al paso 3.</p> <p>2. Si al pulsar el botón crear hay campos sin diligenciar muestra mensaje solicitando al usuario llenar los campos vacíos, regresa al paso 5.</p>	
<b>Pos-condición(es)</b>	El contrato ha sido creado	
<b>Caso de uso</b>	Asignar personal	
<b>Actor(es)</b>	Gerente	
<b>Propósito</b>	Asignar el personal encargado de la fabricación de un nuevo pedido.	

<b>Descripción</b>	Inicia cuando el pedido requiere personal para iniciar su fabricación.	
<b>Precondición(es)</b>	Debe existir disponibilidad de recursos	
<b>Flujo Principal</b>	<b>Acción de actor(es)</b>	<b>Respuesta(s) del sistema</b>
	<p>1. Desde el formulario de creación de pedido el gerente selecciona la opción de asignar personal</p> <p>3. El gerente selecciona los operarios que desea asignar para la fabricación del producto y presiona el botón asignar personal.</p> <p>5. El gerente confirma la asignación.</p>	<p>2. El sistema despliega el formulario para la asignación de personal</p> <p>4. El sistema solicita confirmación de la operación.</p> <p>6. El sistema despliega el formulario de creación de pedido en donde se muestra la asignación realizada</p>
<b>Sub-flujos</b>	1. Si se presiona el botón Asignar personal, y no se ha seleccionado ningún operario, se muestra un mensaje en pantalla solicitando seleccionar por lo menos uno, regresa al paso 3.	
<b>Pos-condición(es)</b>	Se ha seleccionado los operarios a asignar.	
<b>Caso de uso</b>	Asignar materia prima	
<b>Actor(es)</b>	Gerente	
<b>Propósito</b>	Asignar la materia prima necesaria para la fabricación de un nuevo pedido.	
<b>Descripción</b>	Inicia cuando el pedido requiere materia prima para iniciar su fabricación.	
<b>Precondición(es)</b>	Debe existir disponibilidad de recursos	
<b>Flujo Principal</b>	<b>Acción de actor(es)</b>	<b>Respuesta(s) del sistema</b>
	<p>1. Desde el formulario de creación de pedido el gerente selecciona la opción de asignar materia prima</p> <p>3. El gerente selecciona uno a uno los elementos</p>	<p>2. El sistema despliega el formulario para la asignación de materia prima</p> <p>4. El sistema solicita confirmación de la</p>

	que requiere asignar y les asigna una cantidad a ser asignada, por último presiona el botón asignar materiales.  5. El gerente confirma la asignación.	operación.  6. El sistema despliega el formulario de creación de pedido en donde se muestra la asignación realizada
<b>Sub-flujos</b>	1. Si se presiona el botón Asignar material, y no se ha seleccionado ni asignado ningún elemento, se muestra un mensaje en pantalla solicitando seleccionar por lo menos uno, regresa al paso 3.	
<b>Pos-condición(es)</b>	Se ha seleccionado la materia prima a asignar.	
<b>Caso de uso</b>	Ingresar materia prima	
<b>Actor(es)</b>	Almacenista	
<b>Propósito</b>	Ingresar la materia prima al sistema para que haga parte del inventario.	
<b>Descripción</b>	Este caso de uso inicia cuando nueva materia prima solicitada requiere ser ingresada al sistema. El almacenista entra al módulo de ingreso de materia prima e ingresa el nombre, marca y cantidad del producto entrante, dependiendo del producto las cantidades estarán especificadas.	
<b>Precondición(es)</b>		
<b>Flujo Principal</b>	<b>Acción de actor(es)</b>	<b>Respuesta(s) del sistema</b>
	1. El almacenista ingresa al módulo de ingreso de materia prima.  3. El almacenista ingresa la información del material a ingresar y presiona el botón ingresar.  5. El almacenista confirma la operación.	2. El sistema despliega el formulario de ingreso de materia prima.  4. El sistema solicita confirmación de la operación.  6. El sistema confirma el ingreso de la materia prima con un mensaje mostrado en pantalla.
<b>Sub-flujos</b>	1. Si el código ingresado pertenece a material existente	

	<p>en la base de datos, importa los campos de nombre y marca.</p> <p>2. Si se presiona el botón ingresar y no se han llenado todos los campos el sistema mostrará un mensaje solicitando que se llenen, regresa al paso 3.</p>	
<b>Pos-condición(es)</b>	Se ha ingresado la materia prima	
<b>Caso de uso</b>	Entrega materia prima	
<b>Actor(es)</b>	Almacenista	
<b>Propósito</b>	Entregar la materia prima asignada a los grupos de trabajo para realizar el pedido	
<b>Descripción</b>	Este caso de uso inicia cuando un nuevo pedido se empezará a realizar, para lo cual los operarios que confirman el equipo de trabajo reciben toda la materia prima necesaria para la elaboración del producto (previamente asignada por el gerente) por parte del almacenista, quien ingresa en el módulo de entrega de materia prima para hacer la operación, valiéndose del número de pedido.	
<b>Precondición(es)</b>	La materia prima debe haber sido asignada	
<b>Flujo Principal</b>	<b>Acción de actor(es)</b>	<b>Respuesta(s) del sistema</b>
	<p>1. El almacenista ingresa al módulo de entrega de materia prima.</p> <p>3. El almacenista selecciona el código del pedido al cual se hará la entrega de la materia prima.</p> <p>5. El almacenista selecciona el operario encargado de recibir la entrega e ingresa la cantidad de material que se va a entregar, y presiona el botón Aceptar.</p> <p>7. El almacenista confirma la entrega del material.</p>	<p>2. El sistema despliega la ventana de selección de código de pedido.</p> <p>4. El sistema despliega el formulario de entrega de materia prima.</p> <p>6. El sistema solicita la confirmación de la operación.</p> <p>8. El sistema confirma la entrega del material por medio de un mensaje en</p>

		pantalla.
<b>Sub-flujos</b>	1. Si al presionar el botón de aceptar no se ha asignado al operario responsable o no se han ingresado en el formulario la cantidad de material a entregar, el sistema muestra un mensaje solicitando el ingreso de los campos vacíos, regresa al paso 5.	
<b>Pos-condición(es)</b>	El material ha sido entregado.	
<b>Caso de uso</b>	Consultar cotización	
<b>Actor(es)</b>	Gerente	
<b>Propósito</b>	Consultar la información pertinente a una cotización almacenada en el sistema.	
<b>Descripción</b>	El caso de uso inicia cuando el gerente desea conocer información de una cotización creada con anterioridad, para lo cual se dirige al módulo de consulta cotización, en el cual ingresa el código de la cotización a consultar, o selecciona una de la lista, para ver la información de la misma.	
<b>Precondición(es)</b>	Deben existir cotizaciones	
<b>Flujo Principal</b>	<b>Acción de actor(es)</b>	<b>Respuesta(s) del sistema</b>
	1. El gerente ingresa al módulo de consulta de cotización.	2. El sistema despliega el listado de todas las cotizaciones almacenadas en el sistema
	3. El gerente selecciona una de las cotizaciones de la lista o ingresa el código de la cotización a consultar y presiona el botón Buscar.	4. El sistema despliega una ventana con toda la información de la cotización consultada.
<b>Sub-flujos</b>	1. Si al presionar el botón buscar el campo del código de la cotización está vacío, el sistema despliega un mensaje solicitando llenar ese campo, regresa al paso 3. 2. El código de cotización ingresado no existe, regresa al paso 3.	
<b>Pos-condición(es)</b>		
<b>Caso de uso</b>	Consultar contrato	
<b>Actor(es)</b>	Gerente	
<b>Propósito</b>	Consultar la información pertinente a un contrato	

	almacenado en el sistema.	
<b>Descripción</b>	El caso de uso inicia cuando el gerente desea conocer información de un contrato creado con anterioridad, para lo cual se dirige al módulo de consulta contrato, en el cual ingresa el código del contrato a consultar, o selecciona uno de la lista, para ver la información de este.	
<b>Precondición(es)</b>	Deben existir cotizaciones	
<b>Flujo Principal</b>	<b>Acción de actor(es)</b>	<b>Respuesta(s) del sistema</b>
	1. El gerente ingresa al módulo de consulta de contrato.  3. El gerente selecciona uno de los contratos de la lista o ingresa el código del contrato a consultar y presiona el botón Buscar.	2. El sistema despliega el listado de todos los contratos almacenadas en el sistema  4. El sistema despliega una ventana con toda la información del contrato consultado.
<b>Sub-flujos</b>	1. Si al presionar el botón buscar el campo del código del contrato está vacío, el sistema despliega un mensaje solicitando llenar ese campo, regresa al paso 3. 2. El código de contrato ingresado no existe, regresa al paso 3.	
<b>Pos-condición(es)</b>		
<b>Caso de uso</b>	Consultar pedido	
<b>Actor(es)</b>	Gerente	
<b>Propósito</b>	Consultar la información pertinente a un pedido almacenado en el sistema.	
<b>Descripción</b>	El caso de uso inicia cuando el gerente desea conocer información de un pedido creado con anterioridad, para lo cual se dirige al módulo de consulta pedido, en el cual ingresa el código del pedido a consultar, o selecciona uno de la lista, para ver la información de este.	
<b>Precondición(es)</b>	Deben existir cotizaciones	
<b>Flujo Principal</b>	<b>Acción de actor(es)</b>	<b>Respuesta(s) del sistema</b>
	1. El gerente ingresa al módulo de consulta de pedido.  3. El gerente selecciona	2. El sistema despliega el listado de todos los pedidos almacenadas en el sistema  4. El sistema despliega una ventana con toda la

	uno de los pedidos de la lista o ingresa el código del pedido a consultar y presiona el botón Buscar.	información del pedido consultado.
<b>Sub-flujos</b>	<p>1. Si al presionar el botón buscar el campo del código del pedido está vacío, el sistema despliega un mensaje solicitando llenar ese campo, regresa al paso 3.</p> <p>2. El código de pedido ingresado no existe, regresa al paso 3.</p>	
<b>Pos-condición(es)</b>		
<b>Caso de uso</b>	Crear espacio planta	
<b>Actor(es)</b>	Gerente	
<b>Propósito</b>	Crear un registro de todo el Espacio Planta en la organización	
<b>Descripción</b>	El gerente desea agregar nuevo Espacio Planta al sistema para que su información pueda ser utilizada en la aplicación.	
<b>Precondición(es)</b>		
<b>Flujo Principal</b>	<b>Acción de actor(es)</b>	<b>Respuesta(s) del sistema</b>
	<p>1. El gerente ingresa al módulo de creación de espacio planta.</p> <p>3. El gerente ingresa la información del nuevo espacio en planta a crear y presiona el botón Crear.</p> <p>5. El gerente confirma la acción</p>	<p>2. El sistema despliega el formulario de creación de espacio planta.</p> <p>4. El sistema solicita confirmación de la operación.</p> <p>6. El sistema crea el nuevo espacio en planta, muestra un mensaje informando el éxito de la operación y pregunta si se va a agregar un nuevo espacio planta.</p>
<b>Sub-flujos</b>	<p>1. El código de zona ingresado ya existe, regresa al paso 3.</p> <p>2. Si al presionar el botón Crear uno de los campos está vacío, el sistema informa que los campos se deben llenar, regresa al paso 3.</p>	
<b>Pos-condición(es)</b>	Se ha creado un nuevo espacio planta	

<b>Caso de uso</b>	Crear Personal	
<b>Actor(es)</b>	Gerente	
<b>Propósito</b>	Crear un registro de todo el personal presente en la organización	
<b>Descripción</b>	El gerente desea agregar nuevo personal al sistema para que su información pueda ser utilizada en la aplicación.	
<b>Precondición(es)</b>		
<b>Flujo Principal</b>	<b>Acción de actor(es)</b>	<b>Respuesta(s) del sistema</b>
	1. El gerente ingresa al módulo de creación de personal.  3. El gerente ingresa la información del nuevo empleado a crear y presiona el botón Crear.  5. El gerente confirma la acción	2. El sistema despliega el formulario de creación de personal.  4. El sistema solicita confirmación de la operación.  6. El sistema ingresa el nuevo empleado en el sistema, muestra un mensaje informando el éxito de la operación y pregunta si se va a agregar un nuevo empleado.
<b>Sub-flujos</b>	1. La identificación del empleado ingresado ya existe, regresa al paso 3. 2. Si al presionar el botón Crear uno de los campos está vacío, el sistema informa que los campos se deben llenar, regresa al paso 3.	
<b>Pos-condición(es)</b>	Se ha ingresado un nuevo empleado al sistema.	
<b>Caso de uso</b>	Crear proceso	
<b>Actor(es)</b>	Gerente	
<b>Propósito</b>	Crear un registro de todos los procesos presentes en la organización	
<b>Descripción</b>	El gerente desea agregar nuevo proceso al sistema para que su información pueda ser utilizada en la aplicación.	
<b>Precondición(es)</b>		
<b>Flujo Principal</b>	<b>Acción de actor(es)</b>	<b>Respuesta(s) del sistema</b>
	1. El gerente ingresa al	2. El sistema despliega el

	<p>módulo de creación de proceso.</p> <p>3. El gerente ingresa la información del nuevo proceso a crear y presiona el botón Crear.</p> <p>5. El gerente confirma la acción</p>	<p>formulario de creación de proceso.</p> <p>4. El sistema solicita confirmación de la operación.</p> <p>6. El sistema ingresa el nuevo proceso en el sistema, muestra un mensaje informando el éxito de la operación y pregunta si se va a agregar un nuevo proceso.</p>
<b>Sub-flujos</b>	<p>1. El número del proceso ingresado ya existe, regresa al paso 3.</p> <p>2. Si al presionar el botón Crear uno de los campos está vacío, el sistema informa que los campos se deben llenar, regresa al paso 3.</p>	
<b>Pos-condición(es)</b>	Se ha ingresado un nuevo proceso al sistema.	
<b>Caso de uso</b>	Eliminar espacio planta	
<b>Actor(es)</b>	Gerente	
<b>Propósito</b>	Eliminar un espacio en planta del sistema	
<b>Descripción</b>	El caso de uso comienza cuando el gerente decide eliminar un espacio en planta almacenado en el sistema, por lo que se dirige al módulo de eliminación de espacio en planta y selecciona el espacio en planta a eliminar.	
<b>Precondición(es)</b>	Deben existir espacio en planta almacenados	
<b>Flujo Principal</b>	<b>Acción de actor(es)</b>	<b>Respuesta(s) del sistema</b>
	<p>1. El gerente ingresa en el módulo de eliminación de espacio en planta.</p> <p>3. El gerente selecciona el espacio en planta a eliminar, y pulsa el botón Eliminar Espacio.</p>	<p>2. El sistema despliega una ventana con un listado de los espacios en planta que se encuentran almacenados en el sistema</p> <p>4. El sistema solicita confirmación de la operación.</p>

	5. El gerente confirma la operación.	6. El sistema informa del éxito de la operación por medio de un mensaje, y pregunta si se va a eliminar otro espacio en planta.
<b>Sub-flujos</b>	1. Si se pulsa el botón Eliminar Espacio sin haber seleccionado ningún espacio en planta de la lista, el sistema informa que se debe seleccionar un espacio en planta, regresa al paso 3.	
<b>Pos-condición(es)</b>	Se ha eliminado el espacio en planta.	
<b>Caso de uso</b>	Eliminar personal	
<b>Actor(es)</b>	Gerente	
<b>Propósito</b>	Eliminar un empleado del sistema	
<b>Descripción</b>	El caso de uso comienza cuando el gerente decide eliminar un empleado almacenado en el sistema, por lo que se dirige al módulo de eliminación de personal y selecciona el empleado a eliminar.	
<b>Precondición(es)</b>	Deben existir empleados almacenados	
<b>Flujo Principal</b>	<b>Acción de actor(es)</b>	<b>Respuesta(s) del sistema</b>
	1. El gerente ingresa en el módulo de eliminación de personal.	2. El sistema despliega una ventana con un listado de los empleados que se encuentran almacenados en el sistema
	3. El gerente selecciona el empleado a eliminar, y pulsa el botón Eliminar Empleado.	4. El sistema solicita confirmación de la operación.
	5. El gerente confirma la operación.	6. El sistema informa del éxito de la operación por medio de un mensaje, y pregunta si se va a eliminar otro empleado.
<b>Sub-flujos</b>	1. Si se pulsa el botón Eliminar Empleado sin haber seleccionado ningún empleado de la lista, el sistema informa que se debe seleccionar un empleado, regresa al paso 3.	
<b>Pos-condición(es)</b>	Se ha eliminado el empleado.	

<b>Caso de uso</b>	Eliminar proceso.	
<b>Actor(es)</b>	Gerente	
<b>Propósito</b>	Eliminar un proceso del sistema	
<b>Descripción</b>	El caso de uso comienza cuando el gerente decide eliminar un proceso almacenado en el sistema, por lo que se dirige al módulo de eliminación de procesos y selecciona el proceso a eliminar.	
<b>Precondición(es)</b>	Deben existir empleados almacenados	
<b>Flujo Principal</b>	<b>Acción de actor(es)</b>	<b>Respuesta(s) del sistema</b>
	1. El gerente ingresa en el módulo de eliminación de procesos.  3. El gerente selecciona el proceso a eliminar, y pulsa el botón Eliminar Proceso.  5. El gerente confirma la operación.	2. El sistema despliega una ventana con un listado de los procesos que se encuentran almacenados en el sistema  4. El sistema solicita confirmación de la operación.  6. El sistema informa del éxito de la operación por medio de un mensaje, y pregunta si se va a eliminar otro proceso.
<b>Sub-flujos</b>	1. Si se pulsa el botón Eliminar Proceso sin haber seleccionado ningún proceso de la lista, el sistema informa que se debe seleccionar un proceso, regresa al paso 3.	
<b>Pos-condición(es)</b>	Se ha eliminado el proceso.	
<b>Caso de uso</b>	Eliminar producto.	
<b>Actor(es)</b>	Gerente	
<b>Propósito</b>	Eliminar un producto del sistema	
<b>Descripción</b>	El caso de uso comienza cuando el gerente decide eliminar un producto almacenado en el sistema, por lo que se dirige al módulo de eliminación de producto y selecciona el producto a eliminar.	
<b>Precondición(es)</b>	Deben existir empleados almacenados	
<b>Flujo Principal</b>	<b>Acción de actor(es)</b>	<b>Respuesta(s) del sistema</b>
	1. El gerente ingresa en el módulo de eliminación de producto.	2. El sistema despliega una ventana con un listado de los productos que se

	<p>3. El gerente selecciona el producto a eliminar, y pulsa el botón Eliminar Producto.</p> <p>5. El gerente confirma la operación.</p>	<p>encuentran almacenados en el sistema</p> <p>4. El sistema solicita confirmación de la operación.</p> <p>6. El sistema informa del éxito de la operación por medio de un mensaje, y pregunta si se va a eliminar otro producto.</p>
<b>Sub-flujos</b>	1. Si se pulsa el botón Eliminar Producto sin haber seleccionado ningún producto de la lista, el sistema informa que se debe seleccionar un producto, regresa al paso 3.	
<b>Pos-condición(es)</b>	Se ha eliminado el producto.	
<b>Caso de uso</b>	Modificar espacio en planta	
<b>Actor(es)</b>	Gerente	
<b>Propósito</b>	Modificar la información de un espacio en planta almacenado en el sistema	
<b>Descripción</b>	El caso de uso comienza cuando el gerente decide modificar la información de un espacio en planta almacenado en el sistema, para lo cual ingresa al módulo de modificación de espacio en planta y selecciona el que se va a modificar.	
<b>Precondición(es)</b>	Deben existir espacios en planta almacenados	
<b>Flujo Principal</b>	<b>Acción de actor(es)</b>	<b>Respuesta(s) del sistema</b>
	<p>1. El gerente ingresa al módulo de modificación de espacio en planta.</p> <p>3. El gerente selecciona el campo a modificar</p> <p>5. El gerente cambia la</p>	<p>2. El sistema despliega una ventana con el listado de espacios en planta almacenados y un formulario para ingresar los campos a modificar.</p> <p>4. El sistema coloca la información del espacio en planta seleccionado en los campos del formulario.</p> <p>6. El sistema solicita</p>

	información deseada en el formulario y presiona el botón Modificar Selección.  7. El gerente confirma la acción.	confirmación de la operación.  8. El sistema informa del éxito de la operación con un mensaje en pantalla y pregunta si se modificará algún otro registro.
<b>Sub-flujos</b>	1. Si al presionar el botón Modificar Selección hay algún campo vacío en el formulario, el sistema indica que se deben llenar todos los campos, regresa al paso 5.	
<b>Pos-condición(es)</b>	Se ha modificado el espacio en planta.	
<b>Caso de uso</b>	Modificar personal	
<b>Actor(es)</b>	Gerente	
<b>Propósito</b>	Modificar la información de un empleado almacenado en el sistema	
<b>Descripción</b>	El caso de uso comienza cuando el gerente decide modificar la información de un empleado almacenado en el sistema, para lo cual ingresa al módulo de modificación de personal y selecciona el que se va a modificar.	
<b>Precondición(es)</b>	Deben existir empleados almacenados	
<b>Flujo Principal</b>	<b>Acción de actor(es)</b>	<b>Respuesta(s) del sistema</b>
	1. El gerente ingresa al módulo de modificación de personal.  3. El gerente selecciona el campo a modificar  5. El gerente cambia la información deseada en el formulario y presiona el botón Modificar Empleado.	2. El sistema despliega una ventana con el listado de empleados almacenados y un formulario para ingresar los campos a modificar.  4. El sistema coloca la información del empleado seleccionado en los campos del formulario.  6. El sistema solicita confirmación de la operación.

	7. El gerente confirma la acción.	8. El sistema informa del éxito de la operación con un mensaje en pantalla y pregunta si se modificará algún otro registro.
<b>Sub-flujos</b>	1. Si al presionar el botón Modificar Empleado hay algún campo vacío en el formulario, el sistema indica que se deben llenar todos los campos, regresa al paso 5.	
<b>Pos-condición(es)</b>	Se ha modificado el empleado.	
<b>Caso de uso</b>	Modificar proceso	
<b>Actor(es)</b>	Gerente	
<b>Propósito</b>	Modificar la información de un proceso almacenado en el sistema	
<b>Descripción</b>	El caso de uso comienza cuando el gerente decide modificar la información de un proceso almacenado en el sistema, para lo cual ingresa al módulo de modificación de proceso y selecciona el que se va modificar.	
<b>Precondición(es)</b>	Deben existir procesos almacenados	
<b>Flujo Principal</b>	<b>Acción de actor(es)</b>	<b>Respuesta(s) del sistema</b>
	1. El gerente ingresa al módulo de modificación de proceso.	2. El sistema despliega una ventana con el listado de procesos almacenados y un formulario para ingresar los campos a modificar.
	3. El gerente selecciona el campo a modificar	4. El sistema coloca la información del proceso seleccionado en los campos del formulario.
	5. El gerente cambia la información deseada en el formulario y presiona el botón Modificar Proceso.	6. El sistema solicita confirmación de la operación.
	7. El gerente confirma la acción.	8. El sistema informa del éxito de la operación con un mensaje en pantalla y

		pregunta si se modificará algún otro registro.
<b>Sub-flujos</b>	1. Si al presionar el botón Modificar Proceso hay algún campo vacío en el formulario, el sistema indica que se deben llenar todos los campos, regresa al paso 5.	
<b>Pos-condición(es)</b>	Se ha modificado el proceso.	
<b>Caso de uso</b>	Modificar producto	
<b>Actor(es)</b>	Gerente	
<b>Propósito</b>	Modificar la información de un producto almacenado en el sistema	
<b>Descripción</b>	El caso de uso comienza cuando el gerente decide modificar la información de un producto almacenado en el sistema, para lo cual ingresa al módulo de modificación de producto y selecciona el que se va modificar.	
<b>Precondición(es)</b>	Deben existir productos almacenados	
<b>Flujo Principal</b>	<b>Acción de actor(es)</b>	<b>Respuesta(s) del sistema</b>
	1. El gerente ingresa al módulo de modificación de producto.  3. El gerente selecciona el campo a modificar  5. El gerente cambia la información deseada en el formulario y presiona el botón Modificar Producto.  7. El gerente confirma la acción.	2. El sistema despliega una ventana con el listado de productos almacenados y un formulario para ingresar los campos a modificar.  4. El sistema coloca la información del producto seleccionado en los campos del formulario.  6. El sistema solicita confirmación de la operación.  8. El sistema informa del éxito de la operación con un mensaje en pantalla y pregunta si se modificará algún otro registro.
<b>Sub-flujos</b>	1. Si al presionar el botón Modificar Producto hay algún campo vacío en el formulario, el sistema indica que se	

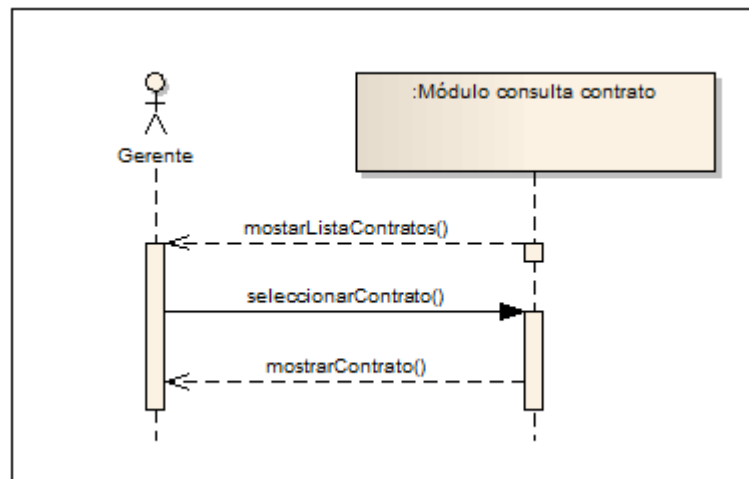
	deben llenar todos los campos, regresa al paso 5.	
<b>Pos-condición(es)</b>	Se ha modificado el producto.	
<b>Caso de uso</b>	Modificar cliente	
<b>Actor(es)</b>	Gerente	
<b>Propósito</b>	Modificar la información de un cliente almacenado en el sistema	
<b>Descripción</b>	El caso de uso comienza cuando el gerente decide modificar la información de un cliente almacenado en el sistema, para lo cual ingresa al módulo de modificación de cliente y selecciona el que se va modificar.	
<b>Precondición(es)</b>	Deben existir cliente almacenados	
<b>Flujo Principal</b>	<b>Acción de actor(es)</b>	<b>Respuesta(s) del sistema</b>
	1. El gerente ingresa al módulo de modificación de cliente.  3. El gerente selecciona el campo a modificar  5. El gerente cambia la información deseada en el formulario y presiona el botón Modificar Cliente.  7. El gerente confirma la acción.	2. El sistema despliega una ventana con el listado de clientes almacenados y un formulario para ingresar los campos a modificar.  4. El sistema coloca la información del cliente seleccionado en los campos del formulario.  6. El sistema solicita confirmación de la operación.  8. El sistema informa del éxito de la operación con un mensaje en pantalla y pregunta si se modificará algún otro cliente.
<b>Sub-flujos</b>	1. Si al presionar el botón Modificar Cliente hay algún campo vacío en el formulario, el sistema indica que se deben llenar todos los campos, regresa al paso 5.	
<b>Pos-condición(es)</b>	Se ha modificado el cliente.	
<b>Caso de uso</b>	Verificar disponibilidad de recursos	
<b>Actor(es)</b>	Gerente	

<b>Propósito</b>	Verificar la disponibilidad de recursos (materia prima, espacio planta, personal) para la fabricación de un pedido.	
<b>Descripción</b>	El caso de uso comienza cuando el gerente va a celebrar un nuevo contrato y requiere conocer si cuenta con los recursos necesarios para la elaboración del producto, para lo cual ingresa en el módulo de verificación de recursos.	
<b>Precondición(es)</b>		
<b>Flujo Principal</b>	<b>Acción de actor(es)</b>	<b>Respuesta(s) del sistema</b>
	1. El gerente ingresa al módulo de verificación de recursos.	2. El sistema despliega 3 listados en pestañas diferentes en los que muestra el estado de los recursos y su respectiva disponibilidad.
<b>Sub-flujos</b>		
<b>Pos-condición(es)</b>		
<b>Caso de uso</b>	Crear producto	
<b>Actor(es)</b>	Gerente	
<b>Propósito</b>	Ingresar al sistema información pertinente sobre un nuevo producto en el sistema.	
<b>Descripción</b>	El caso de uso inicia cuando el gerente requiere ingresar un nuevo producto al sistema, para lo cual se dirige al módulo de creación de producto	
<b>Precondición(es)</b>		
<b>Flujo Principal</b>	<b>Acción de actor(es)</b>	<b>Respuesta(s) del sistema</b>
	1. El gerente ingresa en el módulo de creación de producto.	2. El sistema despliega el formulario de ingreso de producto, el cual contiene los campos que recibirán la información del nuevo producto.
	3. El gerente ingresa la información del nuevo producto, y presiona el botón crear.	4. El sistema solicita la confirmación de la operación.
	5. El gerente confirma la operación.	6. El sistema informa del éxito de la operación por medio de un mensaje en

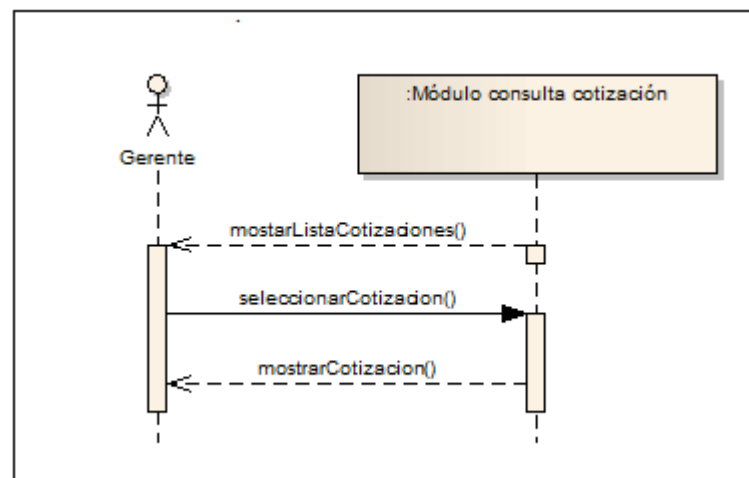
		pantalla y pregunta si se va ingresar otro producto.
<b>Sub-flujos</b>	1. El código de producto ingresado ya existe, regresa al paso 3. 2. Si al presionar el botón crear hay campos sin diligenciar, el sistema informa que todos los campos deben ser diligenciados, regresa al paso 3.	
<b>Pos-condición(es)</b>	El nuevo producto ha sido ingresado al sistema.	

## B. Diagramas de secuencia del sistema

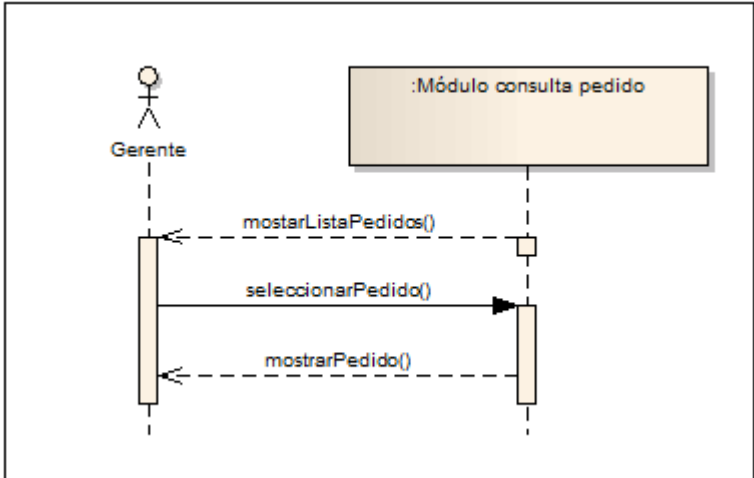
### Consultar Contrato



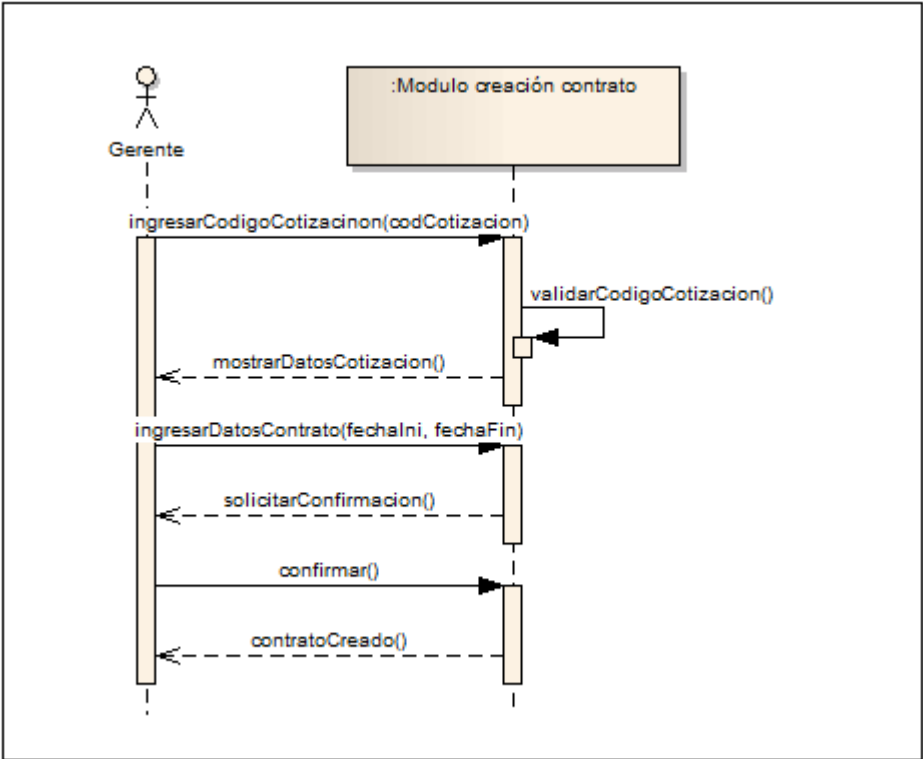
### Consultar cotización



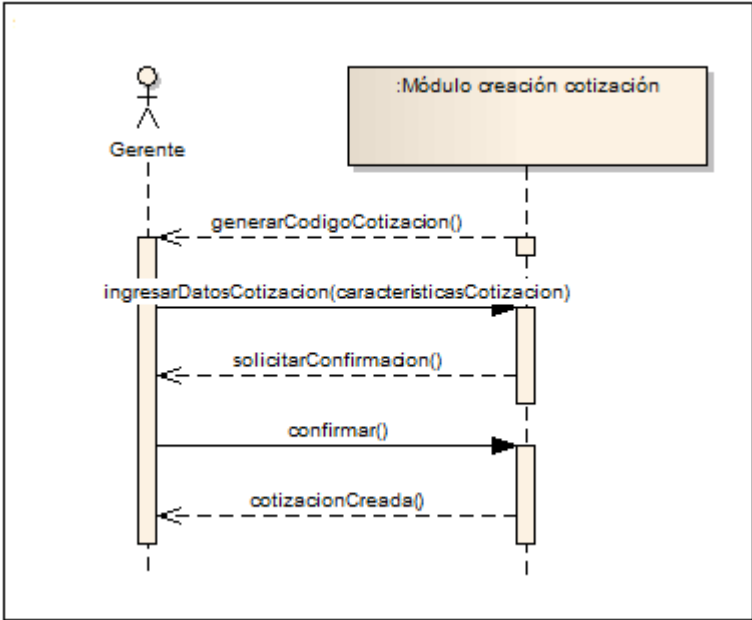
### Consultar pedido



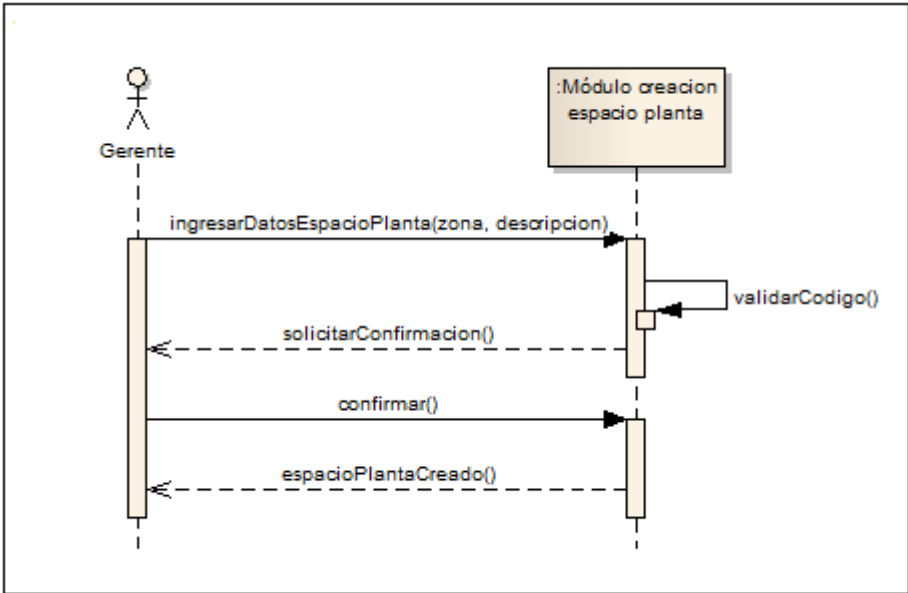
### Crear contrato



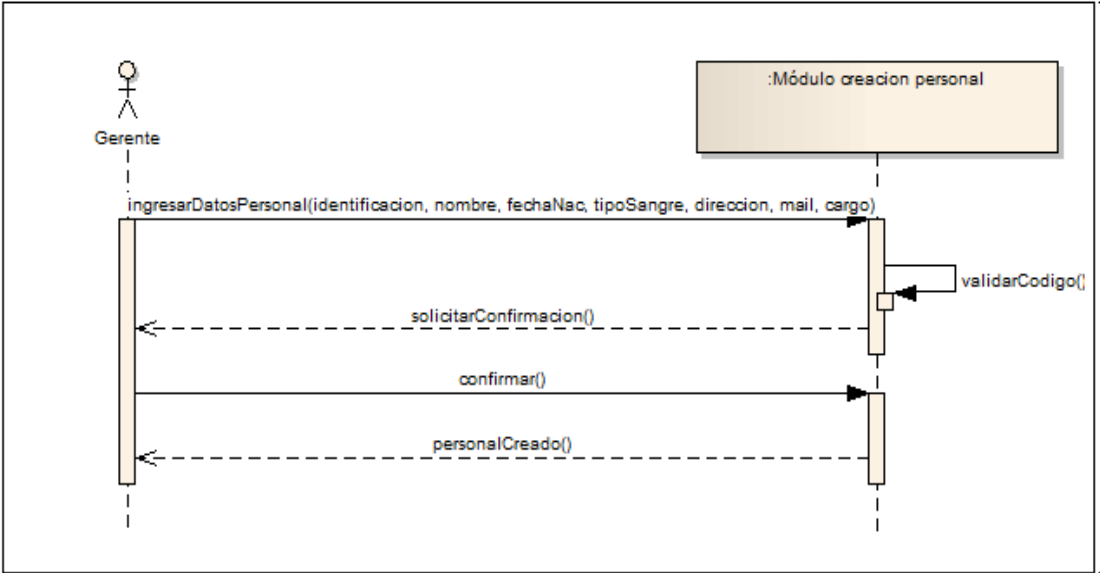
### Crear cotización



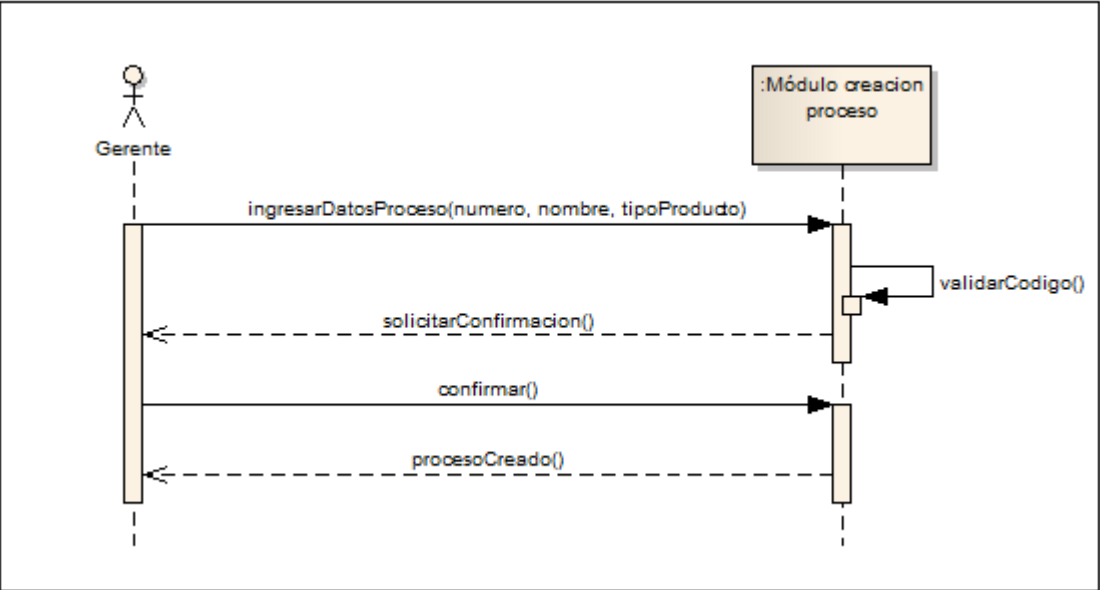
### Crear espacio planta



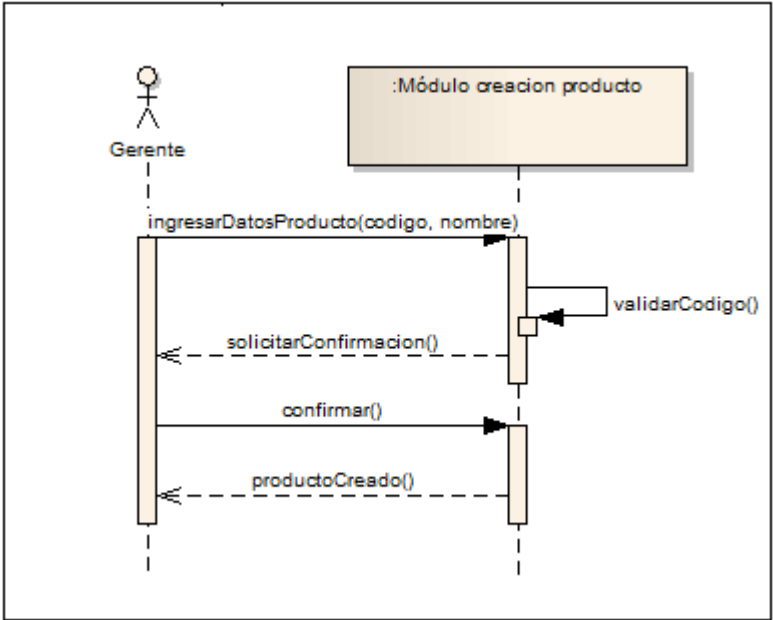
### Crear personal



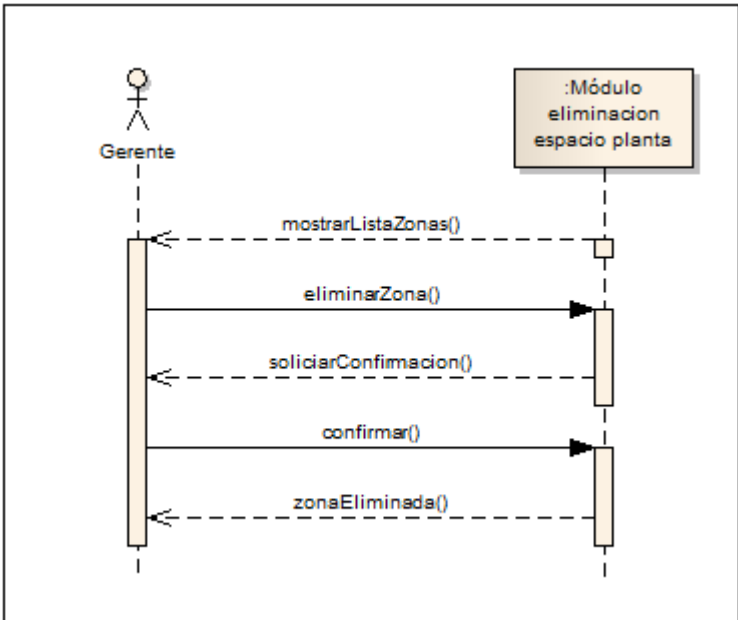
### Crear proceso



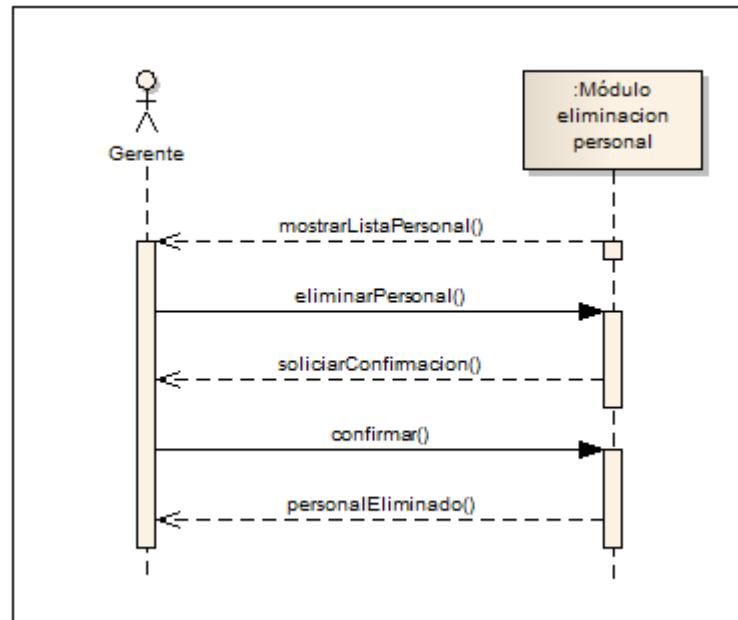
### Crear producto



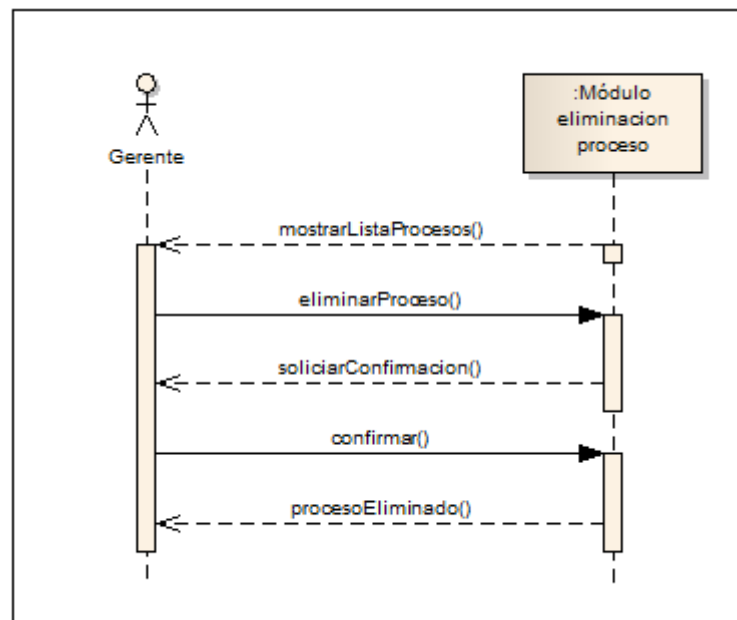
### Eliminar espacio planta



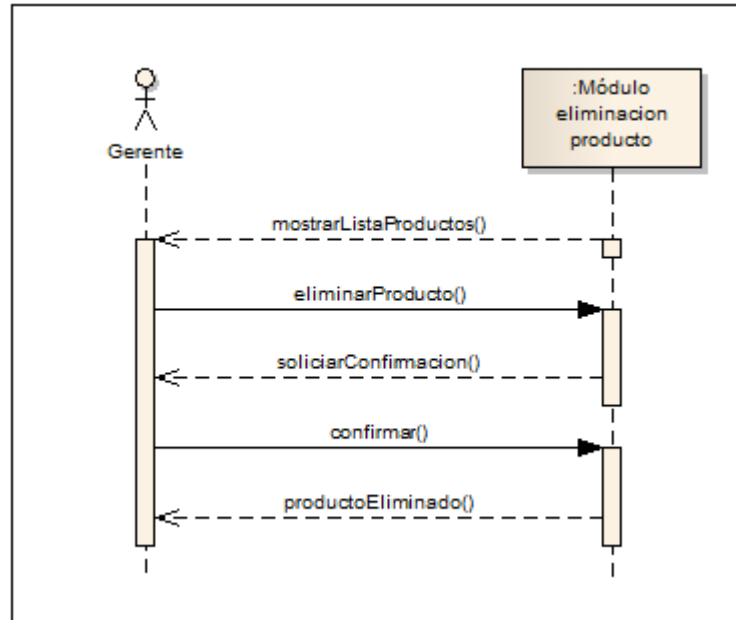
## Eliminar personal



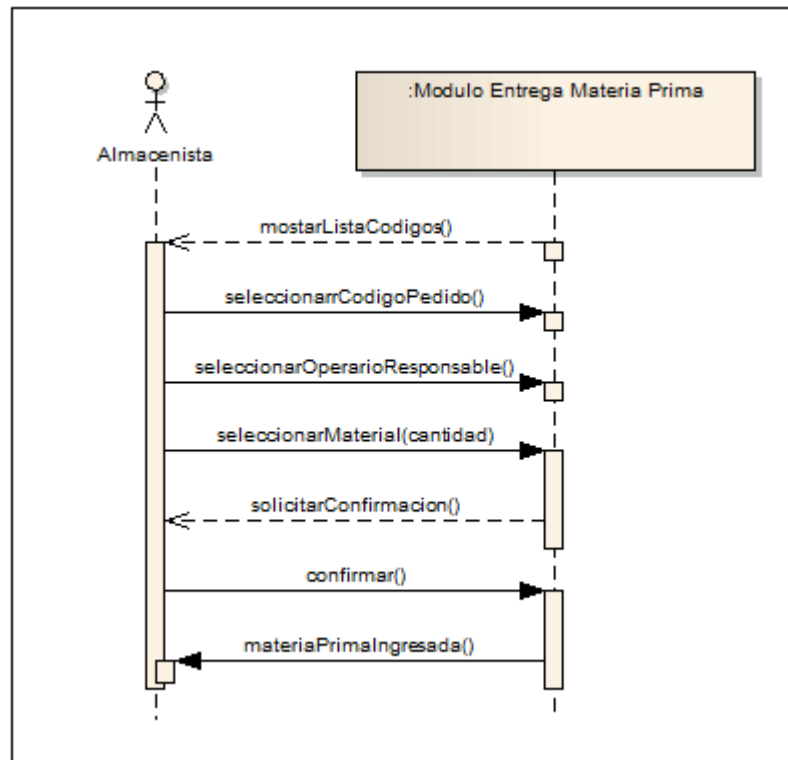
## Eliminar proceso



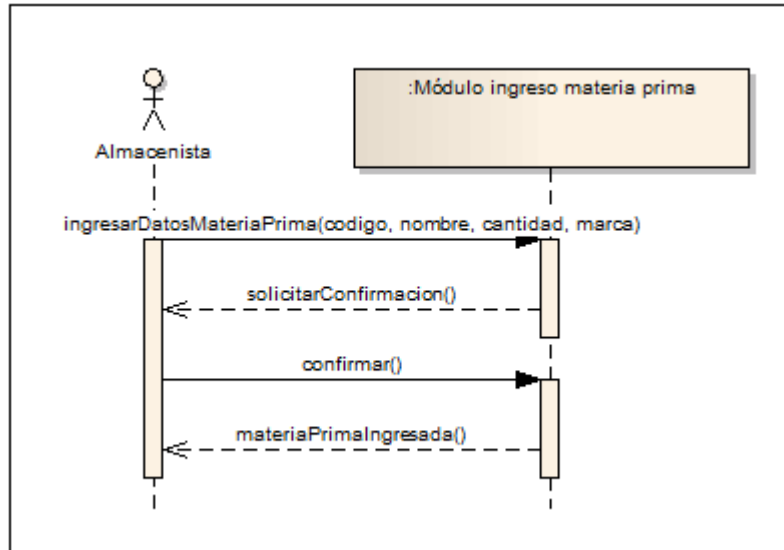
## Eliminar producto



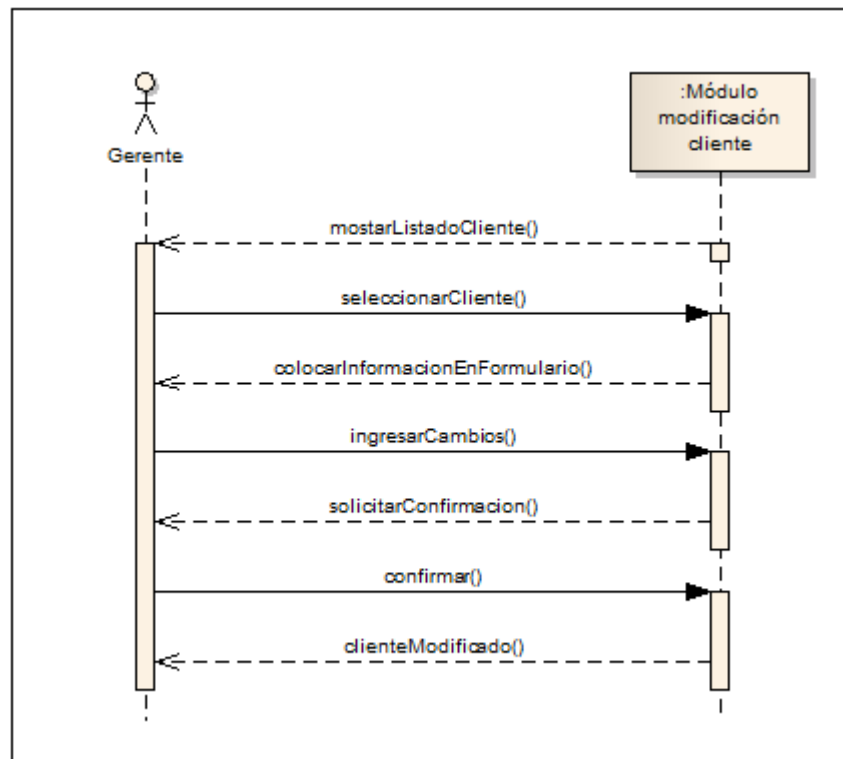
## Entregar materia prima



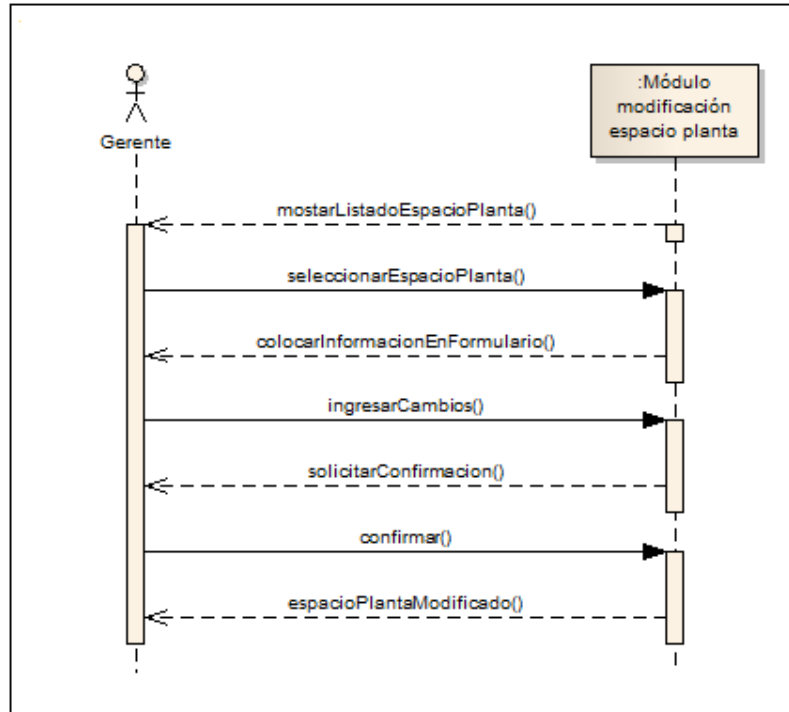
## Ingresar materia prima



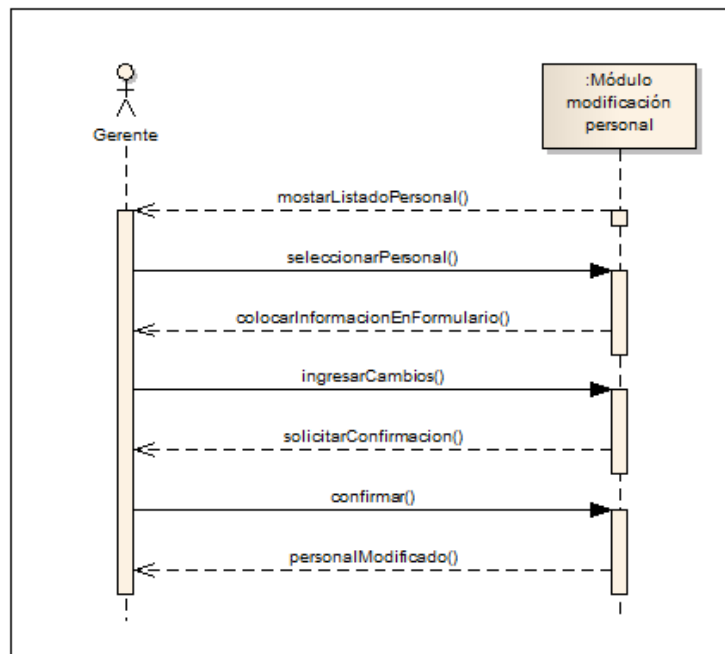
## Modificar cliente



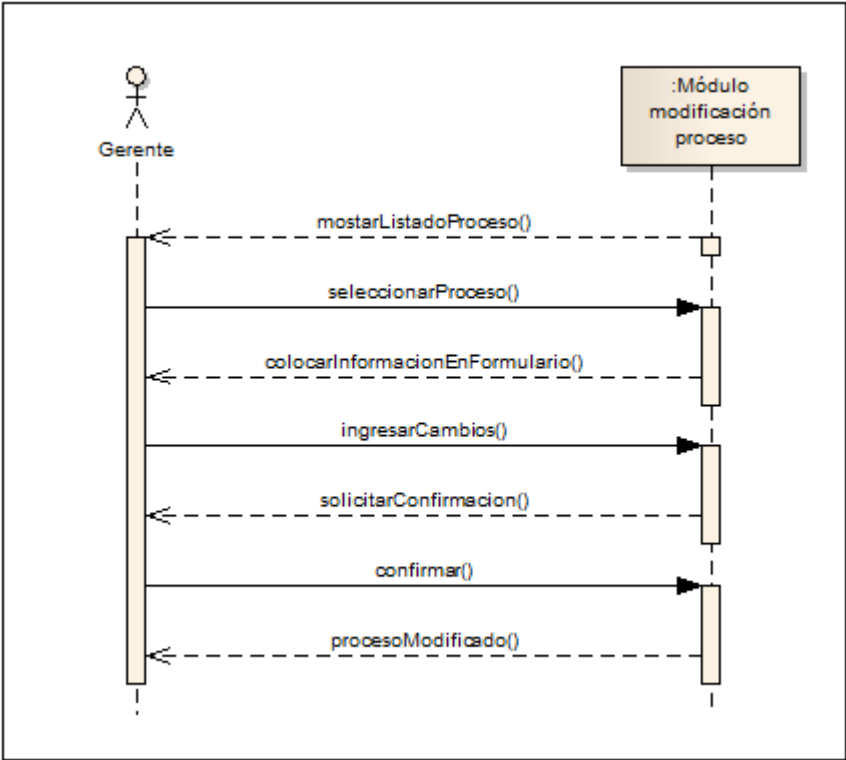
## Modificar espacio planta



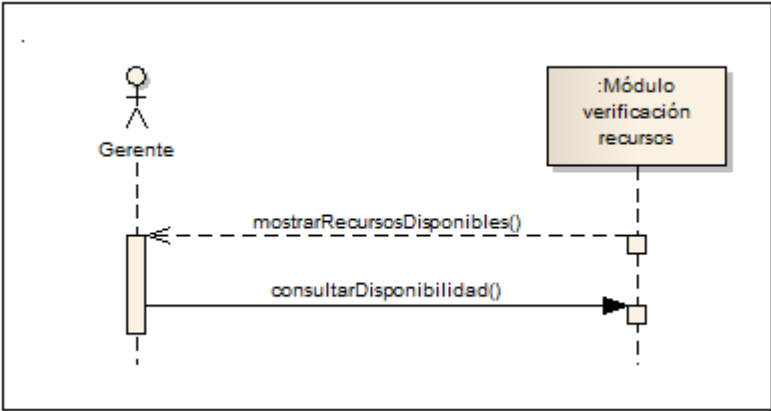
## Modificar personal



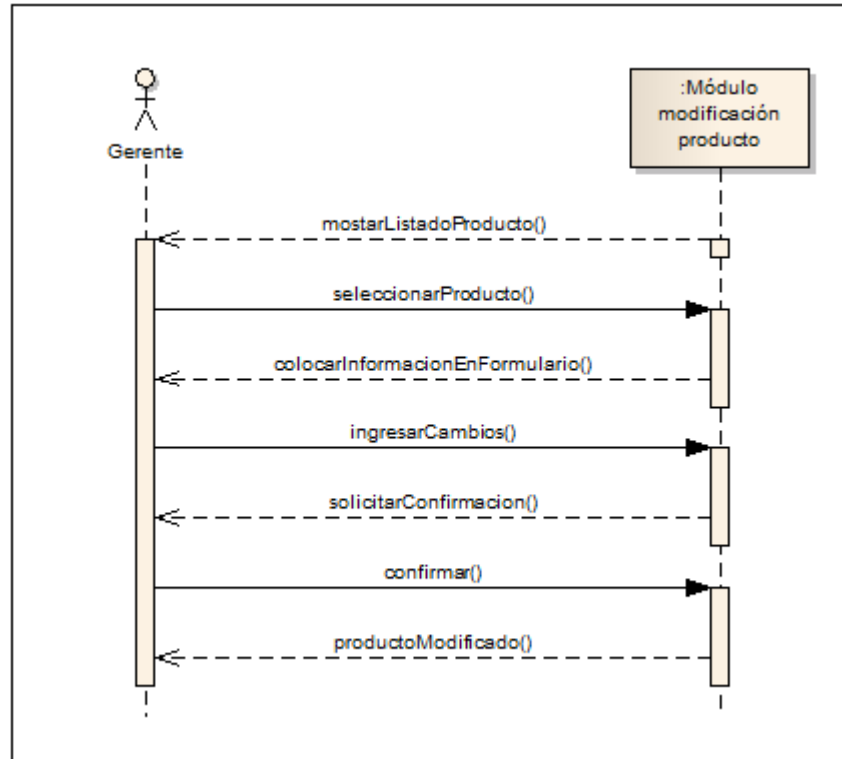
### Modificar proceso



### Verificar recursos



## Modificar producto



### C. Contratos de operación del sistema.

<b>Caso de uso</b>	AsignarMateriaPrima
<b>Nombre del Contrato</b>	seleccionarElementos(cantidad)
<b>Responsabilidades</b>	Capturar y enviar una selección de uno o más elementos para asignar
<b>Tipo</b>	Sistema
<b>Excepciones</b>	-No hay ítems para cargar. -No se ha seleccionado ningún ítem
<b>Salida</b>	-
<b>Pre condición(es)</b>	-Deben existir ítems creados en el sistema
<b>Post condición(es)</b>	-el proceso de creación de pedido debe estar en curso
<b>Caso de uso</b>	AsignarMateriaPrima
<b>Nombre del</b>	confirmar()

<b>Contrato</b>	
<b>Responsabilidades</b>	Efectuar la asignación de la MateriaPrima
<b>Tipo</b>	Sistema
<b>Excepciones</b>	La asignación sea cancelada o declinada
<b>Salida</b>	Se agrega una selección de MateriaPrima dentro de la asignación de recursos asociada a un pedido
<b>Pre condición(es)</b>	Recibir la solicitud de asignar recursos
<b>Post condición(es)</b>	-La selección de materia prima es relacionada con un pedido -La disponibilidad de los ítems es modificado al momento de la asignación.
<b>Caso de uso</b>	AsignarPersonal
<b>Nombre del Contrato</b>	seleccionarPersonal(operario1, operario2, ....., operario n)
<b>Responsabilidades</b>	Capturar y enviar una selección de uno o más operarios para asignar
<b>Tipo</b>	Sistema
<b>Excepciones</b>	-No hay operarios para cargar. -No se ha seleccionado ningún operario
<b>Salida</b>	-
<b>Pre condición(es)</b>	-Deben existir personal creado en el sistema
<b>Post condición(es)</b>	-el proceso de creación de pedido debe estar en curso
<b>Caso de uso</b>	AsignarPersonal
<b>Nombre del Contrato</b>	confirmar()
<b>Responsabilidades</b>	Efectuar la asignación de personal
<b>Tipo</b>	Sistema
<b>Excepciones</b>	La asignación sea cancelada o declinada
<b>Salida</b>	Se agrega una selección de personal dentro de la asignación de recursos asociada a un pedido
<b>Pre condición(es)</b>	Recibir la solicitud de asignar personal
<b>Post condición(es)</b>	-La selección de personal es relacionada con un pedido -La disponibilidad de los operarios es modificada al momento de la asignación.
<b>Caso de uso</b>	ConsultarContrato
<b>Nombre del Contrato</b>	seleccionarContrato
<b>Responsabilidades</b>	Capturar y enviar la selección del contrato para cargar

	la información del contrato
<b>Tipo</b>	Sistema
<b>Excepciones</b>	No hay contratos para consultar
<b>Salida</b>	-
<b>Pre condición(es)</b>	-Debe existir un contrato creado -Debe existir una cotización creada
<b>Post condición(es)</b>	-
<b>Caso de uso</b>	ConsultarCotización
<b>Nombre del Contrato</b>	seleccionarCotización
<b>Responsabilidades</b>	Capturar y enviar la selección de la cotización para cargar su información
<b>Tipo</b>	Sistema
<b>Excepciones</b>	No hay cotizaciones para consultar
<b>Salida</b>	-
<b>Pre condición(es)</b>	-Debe existir una cotización creada
<b>Post condición(es)</b>	-
<b>Caso de uso</b>	ConsultarPedido
<b>Nombre del Contrato</b>	SeleccionarPedido()
<b>Responsabilidades</b>	Capturar y enviar la selección de un pedido para cargar su información
<b>Tipo</b>	Sistema
<b>Excepciones</b>	No hay pedidos para consultar
<b>Salida</b>	-
<b>Pre condición(es)</b>	-Debe existir un pedido creado -Debe existir un contrato creado -Debe existir una cotización creada
<b>Post condición(es)</b>	-
<b>Caso de uso</b>	CrearCotización
<b>Nombre del Contrato</b>	IngresarDatosCotizacion(nomCli, cedCli, dirCli, telCli, mailCli, empresa, nit, dirEmp, telEmp, mailEmp, refProd, cant, nomProd, tipoMat, caracEspec, largo, ancho, alto, diametro, tipoPago, tipoEntrega, observaciones, valUni, subTot, iva, valTot)
<b>Responsabilidades</b>	Capturar y enviar toda la información requerida para la creación de una cotización
<b>Tipo</b>	Sistema
<b>Excepciones</b>	No se capturó información requerida

<b>Salida</b>	-
<b>Pre condición(es)</b>	-El número de cotización debe estar generado -El proceso de cotización de un producto debe estar en curso
<b>Post condición(es)</b>	-
<b>Caso de uso</b>	CrearCotización
<b>Nombre del Contrato</b>	Confirmar
<b>Responsabilidades</b>	Efectúa la operación de creación de la cotización
<b>Tipo</b>	Sistema
<b>Excepciones</b>	La operación de creación es declinada o cancelada
<b>Salida</b>	-se agrega la parte información capturada en la tabla cotización y la otra parte en cliente
<b>Pre condición(es)</b>	-recibir la solicitud de creación de la cotización
<b>Post condición(es)</b>	-Se agrega la información de un nuevo cliente o en su defecto se relaciona la información de la nueva cotización con uno existente -Se permite la creación de un contrato usando el número de la cotización creada -la nueva cotización aparece en las listas de consulta
<b>Caso de uso</b>	CrearContrato
<b>Nombre del Contrato</b>	IngresarCodigoCotizacion(codCotización)
<b>Responsabilidades</b>	Capturar y enviar el código de la cotización para crear su respectivo contrato
<b>Tipo</b>	Sistema
<b>Excepciones</b>	No se capturó información requerida
<b>Salida</b>	-
<b>Pre condición(es)</b>	-El número de cotización debe estar generado -El proceso de creación de un contrato debe estar en curso
<b>Post condición(es)</b>	-
<b>Caso de uso</b>	CrearContrato
<b>Nombre del Contrato</b>	IngresarDatosContrato(fechalni, fechaFin)
<b>Responsabilidades</b>	Capturar y enviar toda la información requerida para la creación de un contrato
<b>Tipo</b>	Sistema
<b>Excepciones</b>	No se capturó información requerida
<b>Salida</b>	-

<b>Pre condición(es)</b>	-El número de cotización debe estar generado -El número de contrato debe estar asignado -El proceso de contrato debe estar en curso
<b>Post condición(es)</b>	-
<b>Caso de uso</b>	
<b>Nombre del Contrato</b>	CrearContrato
<b>Responsabilidades</b>	Confirmar
<b>Tipo</b>	Efectúa la operación de creación del contrato
<b>Excepciones</b>	Sistema
<b>Salida</b>	La operación de creación es declinada o cancelada
<b>Pre condición(es)</b>	-se agrega la parte información capturada en la tabla contrato
<b>Post condición(es)</b>	-recibir la solicitud de creación del contrato -Se permite la creación de un pedido usando el número del contrato creado -el nueva contrato aparece en las listas de consulta
<b>CONTRATOS DE OPERACIÓN DEL SISTEMA</b>	
<b>Caso de uso</b>	Crear EspacioPlanta
<b>Nombre del Contrato</b>	ingresarDatos EspacioPlanta (zona, descripción)
<b>Responsabilidades</b>	Recibir y enviar la información digitada por el usuario sobre un nuevo EspacioPlanta
<b>Tipo</b>	Sistema
<b>Excepciones</b>	No se recibe un numero de zona en el formulario
<b>Salida</b>	-
<b>Pre condición(es)</b>	Se procede ingresar los datos al sistema
<b>Post condición(es)</b>	Se requiere la confirmación de esta acción
<b>Caso de uso</b>	
<b>Nombre del Contrato</b>	Crear EspacioPlanta
<b>Responsabilidades</b>	Confirmar
<b>Tipo</b>	Efectuar la creación de un nuevo EspacioPlanta en la base de datos
<b>Excepciones</b>	Sistema
<b>Salida</b>	Operación cacelada o declinada
<b>Pre condición(es)</b>	EspacioPlanta creado en la base de datos
<b>Post condición(es)</b>	Recibir la solicitud de creación de EspacioPlanta -Nuevo EspacioPlanta ha sido agregado. -Nuevo EspacioPlanta puede ser asignado a un pedido en el sistema. -Nuevo EspacioPlanta puede ser modificado o eliminado en el sistema.

<b>Caso de uso</b>	Crear Personal
<b>Nombre del Contrato</b>	ingresarDatosPersonal(identificacion, nombre, fechaNac, tipoSangre, direccion, mail, cargo)
<b>Responsabilidades</b>	Recibir y enviar la información digitada por el usuario sobre un nuevo cliente
<b>Tipo</b>	Sistema
<b>Excepciones</b>	No se recibe un numero de cedula o nombre en el formulario
<b>Salida</b>	-
<b>Pre condición(es)</b>	Se procede ingresar los datos al sistema
<b>Post condición(es)</b>	Se requiere la confirmación de esta acción
<b>Caso de uso</b>	Crear Personal
<b>Nombre del Contrato</b>	Confirmar
<b>Responsabilidades</b>	Efectuar la creación de una nueva persona en la base de datos
<b>Tipo</b>	Sistema
<b>Excepciones</b>	Operación cancelada o declinada
<b>Salida</b>	Cliente creado en la base de datos
<b>Pre condición(es)</b>	Recibir la solicitud de creación de personal
<b>Post condición(es)</b>	-Nuevo personal ha sido agregado. -Nuevo persona puede ser asignado a un pedido en el sistema. -Nuevo Personal puede ser modificado o eliminado en el sistema.
<b>Caso de uso</b>	CrearProceso
<b>Nombre del Contrato</b>	ingresarDatosProceso(numero, nombre, tipoProducto)
<b>Responsabilidades</b>	Recibir y enviar la información para la creación de un proceso
<b>Tipo</b>	Sistema
<b>Excepciones</b>	-No recibir número, nombre, tipo de producto
<b>Salida</b>	-
<b>Pre condición(es)</b>	-Tipo producto debe estar creado. -El proceso de creación de proceso debe estar en curso.
<b>Post condición(es)</b>	-
<b>Caso de uso</b>	CrearProceso

<b>Nombre del Contrato</b>	Confirmar
<b>Responsabilidades</b>	-Efectuar la creación de un nuevo proceso en la base de datos
<b>Tipo</b>	Sistema.
<b>Excepciones</b>	Operación cancelada o declinada
<b>Salida</b>	-Se agrega un nuevo proceso en la base de datos. -se agrega una nueva relación proceso tipo producto en la base de datos.
<b>Pre condición(es)</b>	-Recibir la solicitud de creación de proceso
<b>Post condición(es)</b>	-el nuevo proceso puede ser asignado a un nuevo pedido. -El nuevo proceso puede ser eliminado -El nuevo proceso es relacionado con un tipo de producto
<b>Caso de uso</b>	CrearProducto
<b>Nombre del Contrato</b>	ingresarDatosProducto(codigo, nombre)
<b>Responsabilidades</b>	Recibir y enviar la información necesaria para el ingreso de un nuevo producto al sistema.
<b>Tipo</b>	Sistema.
<b>Excepciones</b>	No se recibe código de producto
<b>Salida</b>	-
<b>Pre condición(es)</b>	El proceso de creación de producto debe estar en curso
<b>Post condición(es)</b>	-
<b>Caso de uso</b>	CrearProducto
<b>Nombre del Contrato</b>	Confirmar
<b>Responsabilidades</b>	-Efectuar la creación de un nuevo producto en la base de datos
<b>Tipo</b>	Sistema.
<b>Excepciones</b>	Operación cancelada o declinada
<b>Salida</b>	-Se agrega un nuevo producto en la base de datos.
<b>Pre condición(es)</b>	-Recibir la solicitud de creación de producto
<b>Post condición(es)</b>	-El nuevo producto ser asignado a una nueva cotización -El nuevo producto puede ser eliminado -El nuevo producto es relacionado con varios procesos

<b>Caso de uso</b>	EliminarEspacioPlanta
<b>Nombre del Contrato</b>	eliminarZona()
<b>Responsabilidades</b>	Recibir y enviar la zona a ser eliminada
<b>Tipo</b>	Sistema
<b>Excepciones</b>	-
<b>Salida</b>	-
<b>Pre condición(es)</b>	Debe existir Espacio en Planta creado
<b>Post condición(es)</b>	-
<b>Caso de uso</b>	
<b>Caso de uso</b>	EliminarEspacioPlanta
<b>Nombre del Contrato</b>	Confirmar
<b>Responsabilidades</b>	Efectuar la eliminación de un Espacio en Planta
<b>Tipo</b>	Sistema
<b>Excepciones</b>	La operación es cancelada o declinada
<b>Salida</b>	Se suprime el registro de un espacio en planta en la base de datos
<b>Pre condición(es)</b>	Se debe recibir la solicitud de eliminación
<b>Post condición(es)</b>	Se retira la zona de las listas de asignación y verificación de recursos.
<b>Caso de uso</b>	
<b>Caso de uso</b>	EliminarPersonal
<b>Nombre del Contrato</b>	eliminarPersona()
<b>Responsabilidades</b>	Recibir y enviar el personal a ser eliminado
<b>Tipo</b>	Sistema
<b>Excepciones</b>	-
<b>Salida</b>	-
<b>Pre condición(es)</b>	Debe existir personal creado
<b>Post condición(es)</b>	-
<b>Caso de uso</b>	
<b>Caso de uso</b>	EliminarPersonal
<b>Nombre del Contrato</b>	Confirmar()
<b>Responsabilidades</b>	Efectuar la eliminación de Personal
<b>Tipo</b>	Sistema
<b>Excepciones</b>	La operación es cancelada o declinada
<b>Salida</b>	Se suprime el registro de una persona en la base de datos
<b>Pre condición(es)</b>	Se debe recibir la solicitud de eliminación
<b>Post condición(es)</b>	Se retira la persona de las listas de asignación y

	verificación de recursos.
<b>Caso de uso</b>	EliminarProceso
<b>Nombre del Contrato</b>	eliminarProceso()
<b>Responsabilidades</b>	Recibir y enviar el proceso a ser eliminado
<b>Tipo</b>	Sistema
<b>Excepciones</b>	-
<b>Salida</b>	-
<b>Pre condición(es)</b>	Deben existir procesos creados
<b>Post condición(es)</b>	-
<b>Caso de uso</b>	EliminarProceso
<b>Nombre del Contrato</b>	Confirmar()
<b>Responsabilidades</b>	Efectuar la eliminación de un Proceso
<b>Tipo</b>	Sistema
<b>Excepciones</b>	La operación es cancelada o declinada
<b>Salida</b>	Se suprime el registro de un proceso en la base de datos
<b>Pre condición(es)</b>	Se debe recibir la solicitud de eliminación
<b>Post condición(es)</b>	-Se retira el proceso de la tabla relación con producto. -Se retira el proceso de la tabla de procesos -El proceso persiste en la tabla de actualización de un peido determinado.
<b>Caso de uso</b>	EliminarProducto
<b>Nombre del Contrato</b>	eliminarProducto ()
<b>Responsabilidades</b>	Recibir y enviar el producto a ser eliminado
<b>Tipo</b>	Sistema
<b>Excepciones</b>	-
<b>Salida</b>	-
<b>Pre condición(es)</b>	Deben existir productos creados
<b>Post condición(es)</b>	-
<b>Caso de uso</b>	EliminarProducto
<b>Nombre del Contrato</b>	Confirmar()
<b>Responsabilidades</b>	Efectuar la eliminación de un Producto
<b>Tipo</b>	Sistema
<b>Excepciones</b>	La operación es cancelada o declinada

<b>Salida</b>	Se suprime el registro de un Producto en la base de datos
<b>Pre condición(es)</b>	Se debe recibir la solicitud de eliminación
<b>Post condición(es)</b>	-Se retira el Producto de la tabla relación con producto. -Se retira el Producto de la tabla de procesos -El Producto persiste en la tabla asociado a un pedido determinado.
<b>Caso de uso</b>	EntregarMateriaPrima
<b>Nombre del Contrato</b>	seleccionarCodigoPedido
<b>Responsabilidades</b>	Recibir y enviar el código de pedido para hacer la entrega
<b>Tipo</b>	Sistema
<b>Excepciones</b>	-
<b>Salida</b>	-
<b>Pre condición(es)</b>	Pedido creado
<b>Post condición(es)</b>	-
<b>Caso de uso</b>	EntregarMateriaPrima
<b>Nombre del Contrato</b>	seleccionarOperarioResponsable
<b>Responsabilidades</b>	Recibir el operario responsable
<b>Tipo</b>	Sistema
<b>Excepciones</b>	-
<b>Salida</b>	-
<b>Pre condición(es)</b>	La entrega de material debe estar en curso
<b>Post condición(es)</b>	-
<b>Caso de uso</b>	EntregarMateriaPrima
<b>Nombre del Contrato</b>	seleccionarMaterial
<b>Responsabilidades</b>	Recibir la cantidad a entregar de cada material
<b>Tipo</b>	Sistema
<b>Excepciones</b>	-
<b>Salida</b>	-
<b>Pre condición(es)</b>	La entrega de material debe estar en curso
<b>Post condición(es)</b>	-
<b>Caso de uso</b>	EntregarMateriaPrima
<b>Nombre del Contrato</b>	Confirmar

<b>Contrato</b>	
<b>Responsabilidades</b>	-Efectuar la entrega de materia prima. -marcar al operario responsable de la entrega.
<b>Tipo</b>	Sistema
<b>Excepciones</b>	-
<b>Salida</b>	-Se descuenta la cantidad de material entregado a la cantidad de material persistente en la asignación inicial. -Se establece en la asignación el operario responsable de la entrega
<b>Pre condición(es)</b>	-La entrega de material debe estar en curso -recibir la solicitud de entrega de material -campos requeridos deben estar marcados
<b>Post condición(es)</b>	-disminuye la cantidad de material asignado para ese pedido.
<b>Caso de uso</b>	IngresarMateriaPrima
<b>Nombre del Contrato</b>	ingresarDatosMAteriaPrima (codigo, nombre, marca, cantidad)
<b>Responsabilidades</b>	Recibir y enviar la información necesaria para el ingreso de un nuevo ítem de materia prima
<b>Tipo</b>	Sistema.
<b>Excepciones</b>	No se recibe código de materia prima
<b>Salida</b>	-
<b>Pre condición(es)</b>	El proceso de ingreso de materia prima debe estar en curso
<b>Post condición(es)</b>	-
<b>Caso de uso</b>	IngresarMateriaPrima
<b>Nombre del Contrato</b>	Confirmar
<b>Responsabilidades</b>	-Efectuar el ingreso de un nuevo ítem de materia prima en la base de datos
<b>Tipo</b>	Sistema.
<b>Excepciones</b>	Operación cancelada o declinada
<b>Salida</b>	-Se agrega un nuevo ítem de materia prima en la base de datos.
<b>Pre condición(es)</b>	-Recibir la solicitud de ingreso de materia prima
<b>Post condición(es)</b>	-El nuevo ítem ser asignado a un pedido
<b>Caso de uso</b>	ModificarCliente
<b>Nombre del Contrato</b>	seleccionarCliente()

<b>Responsabilidades</b>	Capturar y cargar la información del cliente en el formulario
<b>Tipo</b>	Sistema
<b>Excepciones</b>	No se hayan cargado clientes en la lista
<b>Salida</b>	-
<b>Pre condición(es)</b>	Debe existir cliente creado
<b>Post condición(es)</b>	-
<b>Caso de uso</b>	ModificarCliente
<b>Nombre del Contrato</b>	ingresarCambios()
<b>Responsabilidades</b>	Capturar y enviar los cambios deseados para el registro del cliente
<b>Tipo</b>	Sistema
<b>Excepciones</b>	no se captura un numero de cedula
<b>Salida</b>	-
<b>Pre condición(es)</b>	El proceso de modificación de cliente debe estar llevándose acabo
<b>Post condición(es)</b>	-
<b>Caso de uso</b>	ModificarCliente
<b>Nombre del Contrato</b>	Confirmar()
<b>Responsabilidades</b>	Efectuar el proceso de modificación
<b>Tipo</b>	Sistema
<b>Excepciones</b>	El número de cedula es remplazado por uno existente
<b>Salida</b>	Se efectúan los cambios en un elemento de la tabla cliente
<b>Pre condición(es)</b>	Cliente debe estar creado
<b>Post condición(es)</b>	-El cliente es modificado. -Sus modificaciones se evidencian en todas las listas y relaciones de cliente
<b>Caso de uso</b>	ModificarPersonal
<b>Nombre del Contrato</b>	seleccionarPersonal ()
<b>Responsabilidades</b>	Capturar y cargar la información del Personal en el formulario
<b>Tipo</b>	Sistema
<b>Excepciones</b>	No se haya cargado Personal en la lista
<b>Salida</b>	-
<b>Pre condición(es)</b>	Debe existir Personal creado

<b>Post condición(es)</b>	-
<b>Caso de uso</b>	ModificarPersonal
<b>Nombre del Contrato</b>	ingresarCambios()
<b>Responsabilidades</b>	Capturar y enviar los cambios deseados para el registro del Personal
<b>Tipo</b>	Sistema
<b>Excepciones</b>	no se captura un numero de cedula
<b>Salida</b>	-
<b>Pre condición(es)</b>	El proceso de modificación de Personal debe estar llevándose acabo
<b>Post condición(es)</b>	-
<b>Caso de uso</b>	ModificarPersonal
<b>Nombre del Contrato</b>	Confirmar()
<b>Responsabilidades</b>	Efectuar el proceso de modificación
<b>Tipo</b>	Sistema
<b>Excepciones</b>	El número de cedula es remplazado por uno existente
<b>Salida</b>	Se efectúan los cambios en un elemento de la tabla personal
<b>Pre condición(es)</b>	Personal debe estar creado
<b>Post condición(es)</b>	-El personal es modificado. -Sus modificaciones se evidencian en todas las listas y relaciones de personal
<b>Caso de uso</b>	ModificarEspacioPlanta
<b>Nombre del Contrato</b>	seleccionarEspacioPlanta ()
<b>Responsabilidades</b>	Capturar y cargar la información del EspacioPlanta en el formulario
<b>Tipo</b>	Sistema
<b>Excepciones</b>	No se hayan cargado zonas en la lista
<b>Salida</b>	-
<b>Pre condición(es)</b>	Debe existir zona creada
<b>Post condición(es)</b>	-
<b>Caso de uso</b>	ModificarEspacioPlanta
<b>Nombre del Contrato</b>	ingresarCambios()
<b>Responsabilidades</b>	Capturar y enviar los cambios deseados para el

	registro de EspacioPlanta
<b>Tipo</b>	Sistema
<b>Excepciones</b>	no se captura un numero de zona
<b>Salida</b>	-
<b>Pre condición(es)</b>	El proceso de modificación de EspacioPlanta debe estar llevándose acabo
<b>Post condición(es)</b>	-
<b>Caso de uso</b>	ModificarEspacioPlanta
<b>Nombre del Contrato</b>	Confirmar()
<b>Responsabilidades</b>	Efectuar el proceso de modificación
<b>Tipo</b>	Sistema
<b>Excepciones</b>	El número de zona es remplazado por uno existente
<b>Salida</b>	Se efectúan los cambios en un elemento de la tabla espacio planta
<b>Pre condición(es)</b>	La zona debe estar creada
<b>Post condición(es)</b>	-La zona es modificado. -Sus modificaciones se evidencian en todas las listas y relaciones de espacio planta
<b>Caso de uso</b>	ModificarProceso
<b>Nombre del Contrato</b>	seleccionarProceso ()
<b>Responsabilidades</b>	Capturar y cargar la información del Proceso en el formulario
<b>Tipo</b>	Sistema
<b>Excepciones</b>	No se hayan cargado Procesos en la lista
<b>Salida</b>	-
<b>Pre condición(es)</b>	Debe existir Proceso creado
<b>Post condición(es)</b>	-
<b>Caso de uso</b>	ModificarProceso
<b>Nombre del Contrato</b>	ingresarCambios()
<b>Responsabilidades</b>	Capturar y enviar los cambios deseados para el Proceso
<b>Tipo</b>	Sistema
<b>Excepciones</b>	no se captura un numero de Proceso
<b>Salida</b>	-
<b>Pre condición(es)</b>	El proceso de modificación de Proceso debe estar llevándose acabo

<b>Post condición(es)</b>	-
<b>Caso de uso</b>	ModificarProceso
<b>Nombre del Contrato</b>	Confirmar()
<b>Responsabilidades</b>	Efectuar el proceso de modificación
<b>Tipo</b>	Sistema
<b>Excepciones</b>	El número de Proceso es remplazado por uno existente
<b>Salida</b>	Se efectúan los cambios en un elemento de la tabla Proceso
<b>Pre condición(es)</b>	El Proceso debe estar creada
<b>Post condición(es)</b>	-El Proceso es modificado. -Sus modificaciones se evidencian en todas las listas y relaciones de Proceso
<b>Caso de uso</b>	ModificarProducto
<b>Nombre del Contrato</b>	seleccionarProducto ()
<b>Responsabilidades</b>	Capturar y cargar la información del Producto en el formulario
<b>Tipo</b>	Sistema
<b>Excepciones</b>	No se hayan cargado Productos en la lista
<b>Salida</b>	-
<b>Pre condición(es)</b>	Debe existir Producto creado
<b>Post condición(es)</b>	-
<b>Caso de uso</b>	ModificarProducto
<b>Nombre del Contrato</b>	ingresarCambios()
<b>Responsabilidades</b>	Capturar y enviar los cambios deseados para el Producto
<b>Tipo</b>	Sistema
<b>Excepciones</b>	no se captura un numero de Producto
<b>Salida</b>	-
<b>Pre condición(es)</b>	El Producto de modificación de Producto debe estar llevándose acabo
<b>Post condición(es)</b>	-
<b>Caso de uso</b>	ModificarProducto
<b>Nombre del Contrato</b>	Confirmar()

<b>Responsabilidades</b>	Efectuar el Producto de modificación
<b>Tipo</b>	Sistema
<b>Excepciones</b>	El número de Producto es remplazado por uno existente
<b>Salida</b>	Se efectúan los cambios en un elemento de la tabla Producto
<b>Pre condición(es)</b>	El Producto debe estar creada
<b>Post condición(es)</b>	-El Producto es modificado. -Sus modificaciones se evidencian en todas las listas y relaciones de Producto
<b>Caso de uso</b>	VerificarRecursosDisponibles
<b>Nombre del Contrato</b>	consultarDisponibilidad()
<b>Responsabilidades</b>	Cargar la información de los recursos disponibles
<b>Tipo</b>	Sistema
<b>Excepciones</b>	No hay información de los recursos
<b>Salida</b>	-
<b>Pre condición(es)</b>	Deben existir recursos ingresados
<b>Post condición(es)</b>	-

#### D. Listado de objetos por caso de uso.

<b>Crear cotización</b>		
<b>Control</b>	<b>Borde</b>	<b>Entidad</b>
controladorPrincipal controlCreacionCotizacion	interfaceCreacionCotizacion ventanaConfirmacion ventanaMensaje	cliente cotizacion
<b>Crear contrato</b>		
<b>Control</b>	<b>Borde</b>	<b>Entidad</b>
controladorPrincipal controlConsultaCotizacion controlCreacionContrato	VVentanaValidacionCotizacion VInterfaceCreacionContrato ventanaConfirmacion ventanaMensaje	contrato cotizacion
<b>Consultar cotización</b>		
<b>Control</b>	<b>Borde</b>	<b>Entidad</b>
controladorPrincipal controlConsultaCotizacion	VInterfaceConsultaCotizacion VPantallaMostrarCotizacion	cotizacion
<b>Consultar contrato</b>		
<b>Control</b>	<b>Borde</b>	<b>Entidad</b>
controladorPrincipal	VInterfaceConsultaContrato	contrato

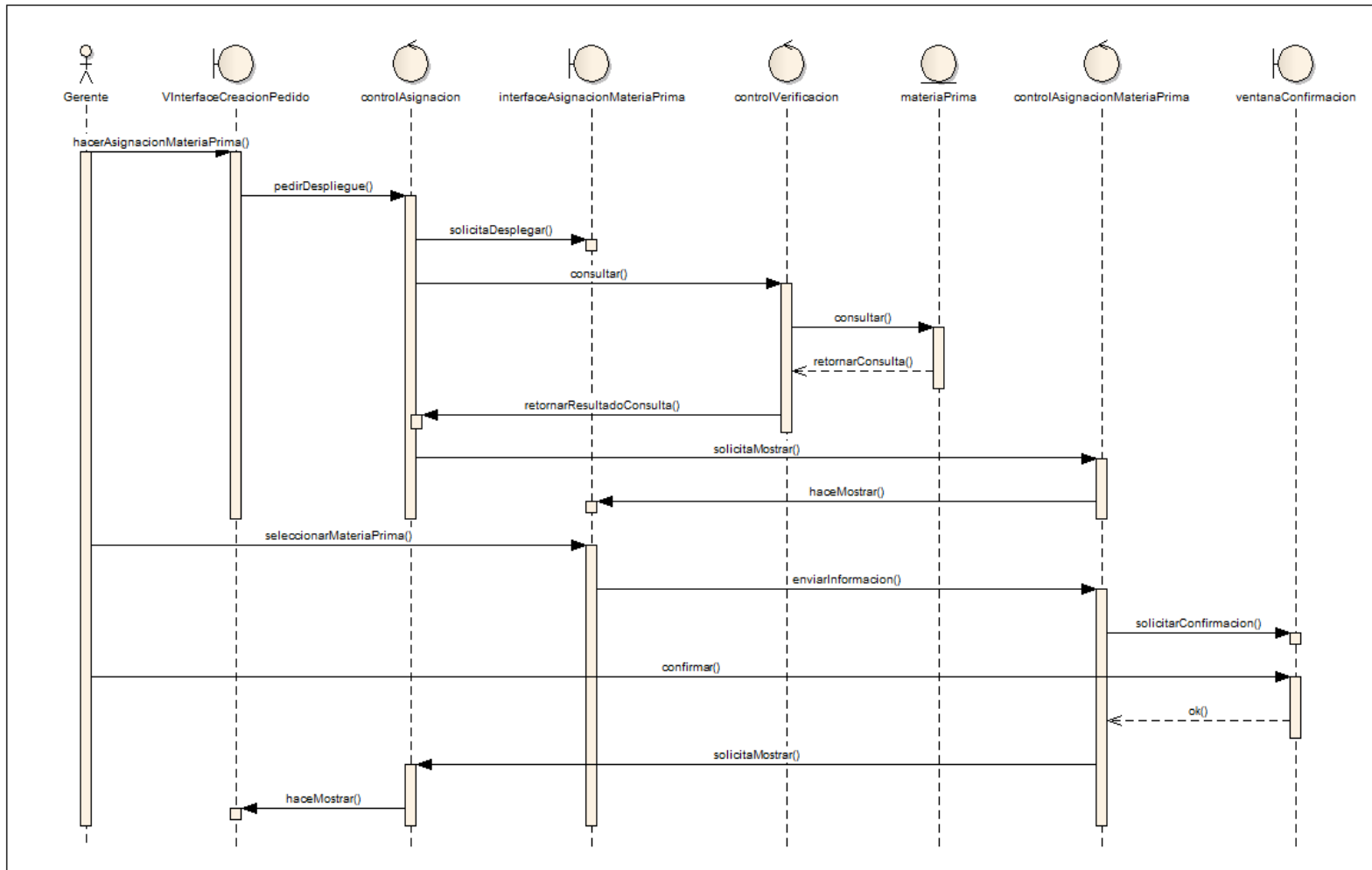
controlConsultaContrato controlConsultaCotizacion	VPantallaMostrarContrato	cotizacion
<b>Consultar pedido</b>		
<b>Control</b>	<b>Borde</b>	<b>Entidad</b>
controladorPrincipal controlConsultaAsignacion controlConsultaPedido	VInterfaceConsultaPedido VPantallaMostrarPedido	espacioPlanta materiaPrima pedido personal
<b>Asignar personal</b>		
<b>Control</b>	<b>Borde</b>	<b>Entidad</b>
controlAsignacion controlAsignacionPersonal controlVerificación	VInterfaceAsignacionPersonal VinterfaceCreacionPedido ventanaConfirmacion	personal
<b>Asignar materia prima</b>		
<b>Control</b>	<b>Borde</b>	<b>Entidad</b>
controlAsignacion controlAsignacionMateriaPri controlVerificacion	VInterfaceAsignacionMateriaPrima VInterfaceCreacionPedido ventanaConfirmacion	materiaPrima
<b>Crear espacio planta</b>		
<b>Control</b>	<b>Borde</b>	<b>Entidad</b>
controlAñadirEspacioPlanta controladorPrincipal	VInterfaceCreacionEspacioPlanta ventanaConfirmacion ventanaMensaje	espacioPlanta
<b>Crear personal</b>		
<b>Control</b>	<b>Borde</b>	<b>Entidad</b>
controladorPrincipal controlAñadirPersonal	VInterfaceCreacionPersonal ventanaConfirmacion ventanaMensaje	personal
<b>Crear proceso</b>		
<b>Control</b>	<b>Borde</b>	<b>Entidad</b>
controladorPrincipal controlCreacionProceso	VInterfaceCreacionProceso ventanaConfirmacion ventanaMensaje	proceso
<b>Eliminar espacio planta</b>		
<b>Control</b>	<b>Borde</b>	<b>Entidad</b>
controlEliminacionEspacioP la	VInterfaceEliminacionEspacioPlant a ventanaConfirmacion ventanaMensaje	espacioPlanta

<b>Eliminar personal</b>		
<b>Control</b>	<b>Borde</b>	<b>Entidad</b>
controleliminacionPersonal	VInterfaceEliminarPersonal ventanaConfirmacion ventanaMensaje	personal
<b>Eliminar proceso</b>		
<b>Control</b>	<b>Borde</b>	<b>Entidad</b>
controlEliminacionProceso	VInterfaceEliminacionProceso ventanaConfirmacion ventanaMensaje	proceso
<b>Modificar espacio planta</b>		
<b>Control</b>	<b>Borde</b>	<b>Entidad</b>
controlModificacionEspacio PI	VInterfaceModificacionEspacioP ventanaConfirmacion ventanaMensaje	espacioPlanta
<b>Modificar personal</b>		
<b>Control</b>	<b>Borde</b>	<b>Entidad</b>
controlModificacionPersonal	VInterfaceModificacionPersonal ventanaConfirmacion ventanaMensaje	personal
<b>Modificar cliente</b>		
<b>Control</b>	<b>Borde</b>	<b>Entidad</b>
controlModificacionCliente	VInterfaceModificacionCliente ventanaConfirmacion ventanaMensaje	cliente
<b>Verificar recursos</b>		
<b>Control</b>	<b>Borde</b>	<b>Entidad</b>
controladorPrincipal controlVerificacion	VInterfaceVerificacionRecursosDis interfaceGerente	materiaPrima personal espacioPlanta
<b>Ingresar materia prima</b>		
<b>Control</b>	<b>Borde</b>	<b>Entidad</b>
controladorPrincipal controlIngresoMateriaPrima	interfazIngresoMateriaPrima ventanaConfirmacion ventanaMensaje	materiaPrima
<b>Entregar materia prima</b>		
<b>Control</b>	<b>Borde</b>	<b>Entidad</b>
controladorPrincipal controlConsultaPedido controlEntregaMaterial	VInterfaceEntregaMateriaPrima VPantallaValidacionCodigoPedido ventanaConfirmacion	materiaPrima pedido personal

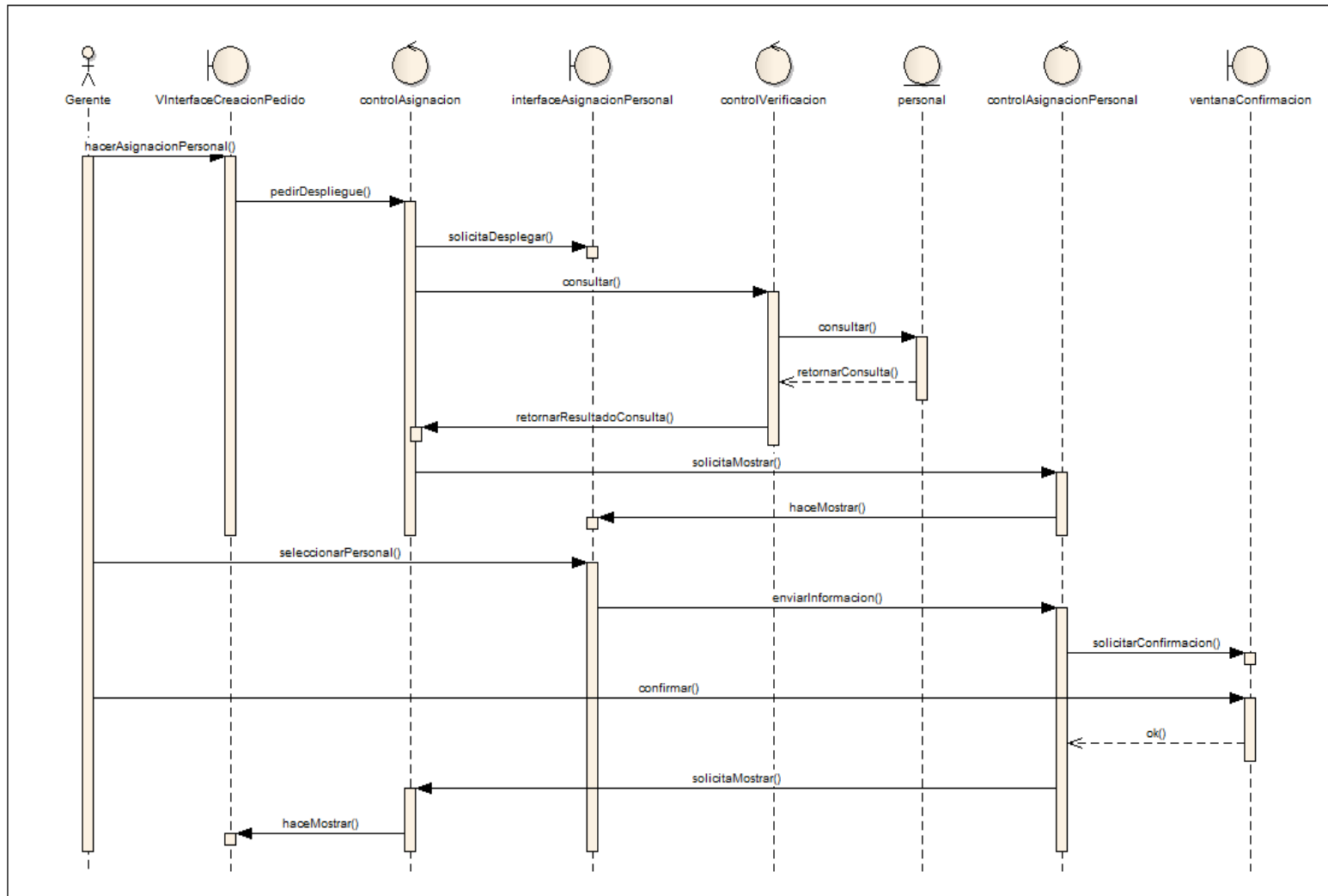
	ventanaMensaje	
<b>Modificar producto</b>		
<b>Control</b>	<b>Borde</b>	<b>Entidad</b>
controlModificacionProducto	VInterfaceModificacionProducto ventanaConfirmacion ventanaMensaje	producto
<b>Modificar proceso</b>		
<b>Control</b>	<b>Borde</b>	<b>Entidad</b>
controlModificacionProceso	VInterfaceModificacionProceso ventanaConfirmacion ventanaMensaje	proceso
<b>Eliminar producto</b>		
<b>Control</b>	<b>Borde</b>	<b>Entidad</b>
controlEliminacionProducto	VInterfaceEliminacionProducto ventanaConfirmacion ventanaMensaje	producto

## E. Diagramas de secuencia con objetos.

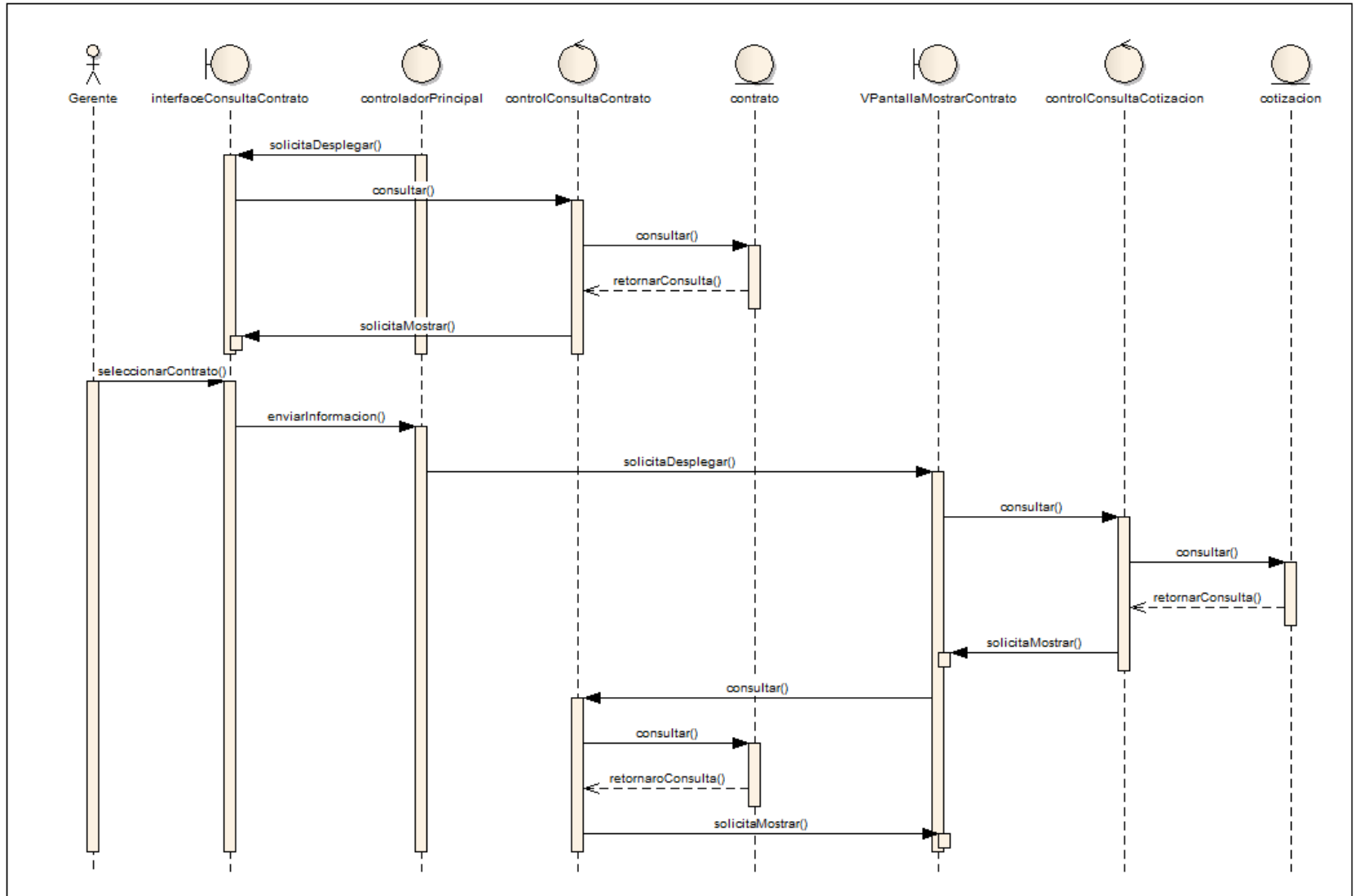
### Asignar materia prima



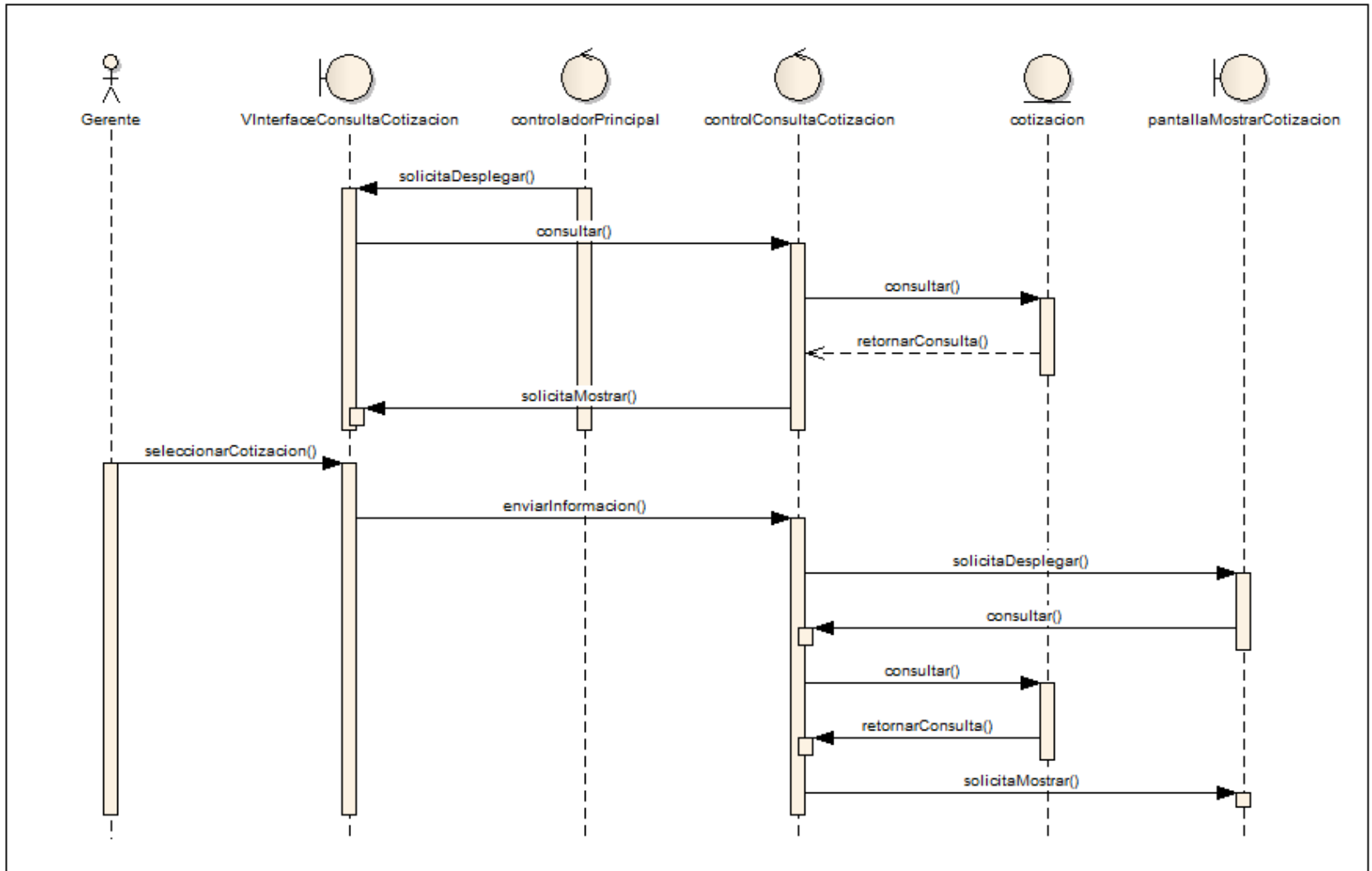
## Asignar Personal



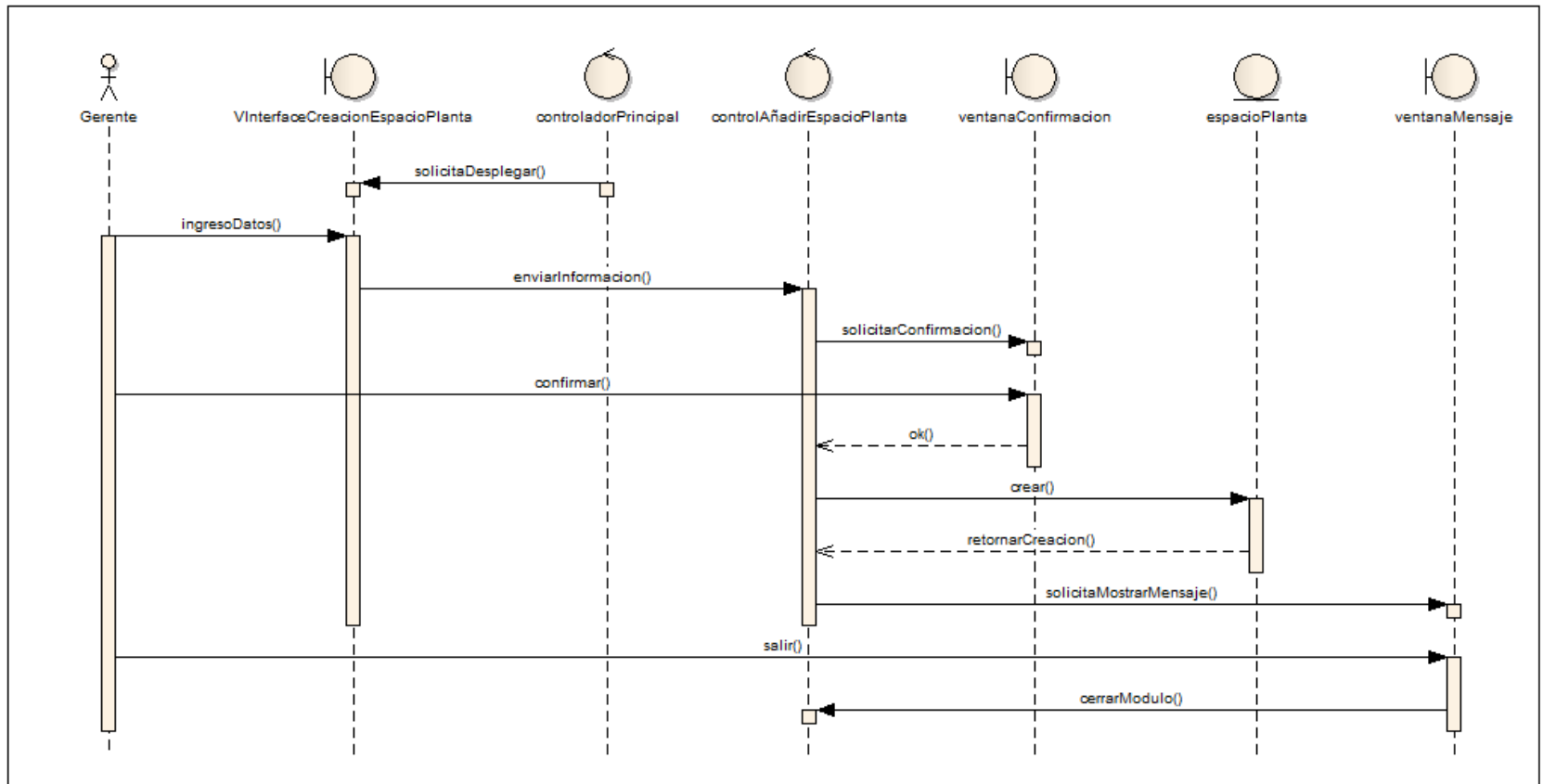
## Consultar Contrato



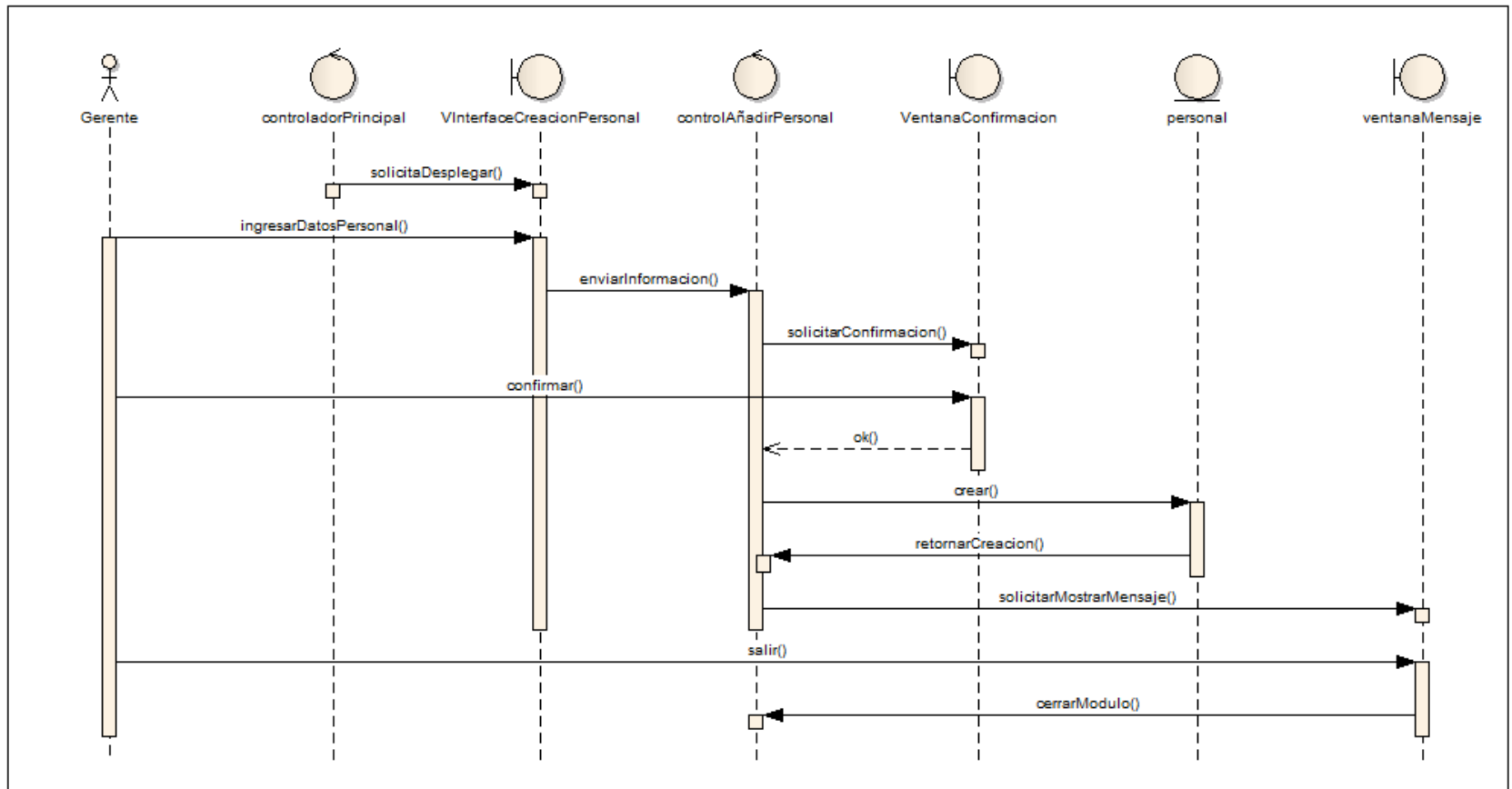
## Consultar cotización



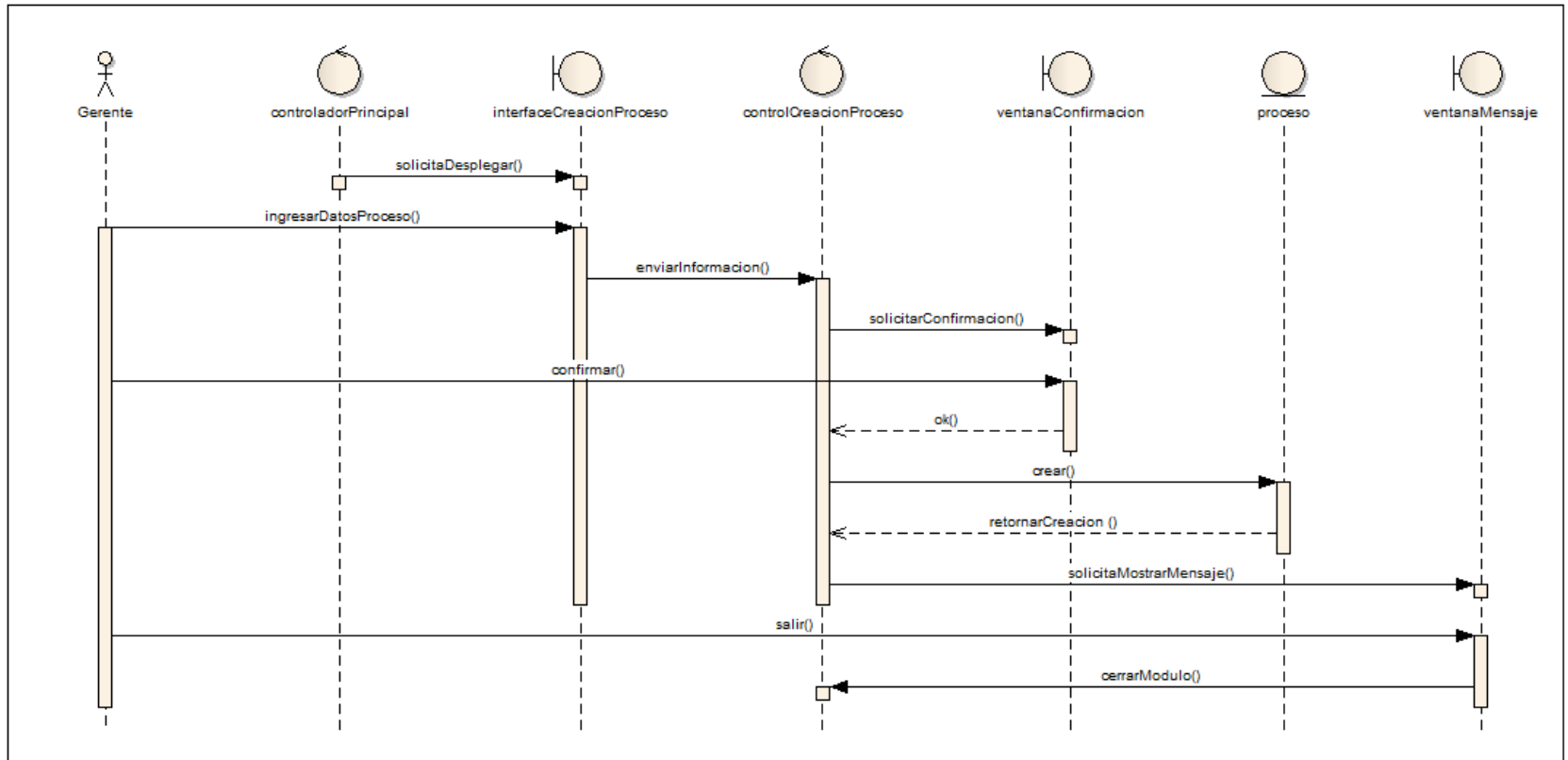
## Crear espacio Planta



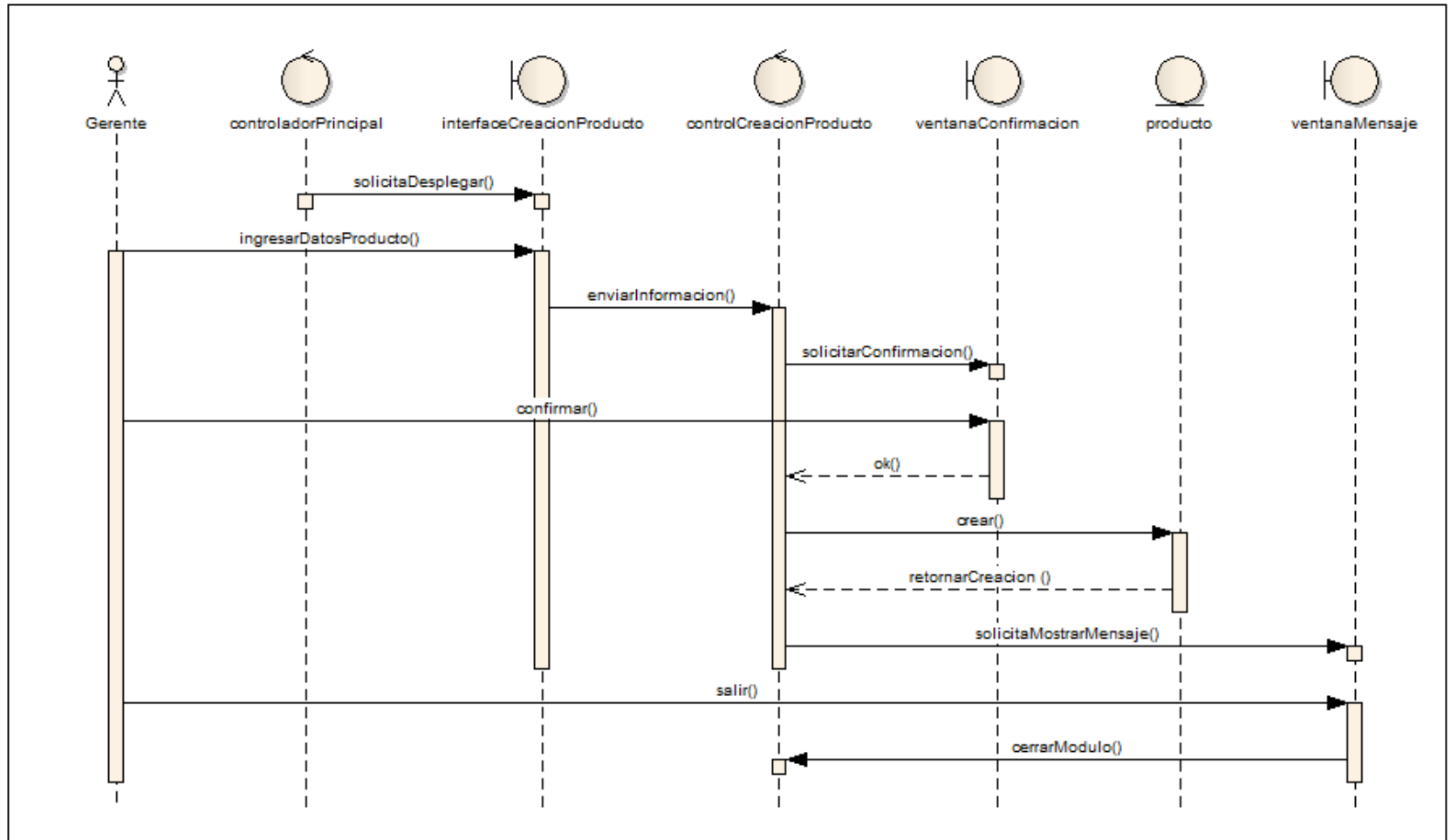
## Crear personal



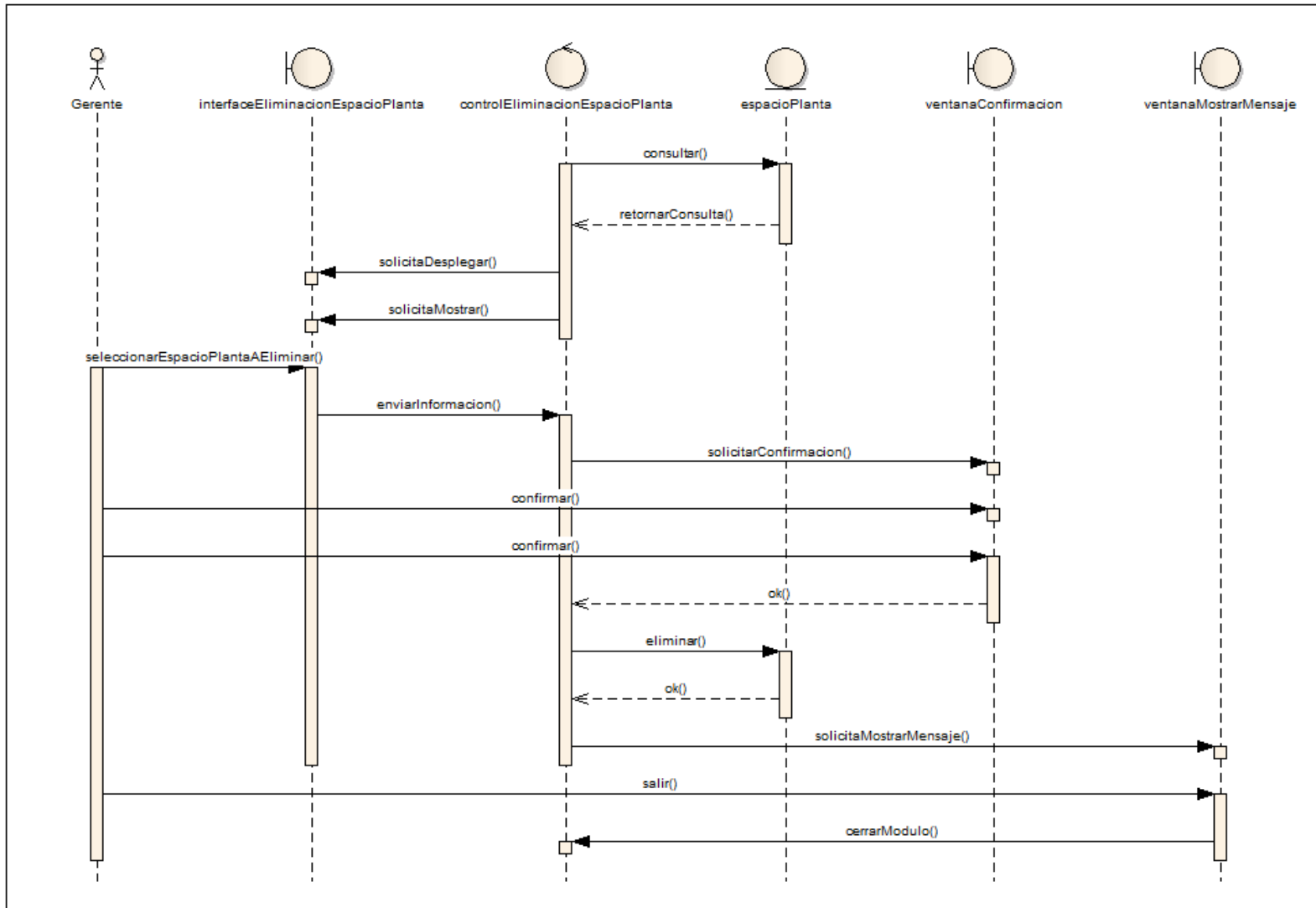
## Crear proceso



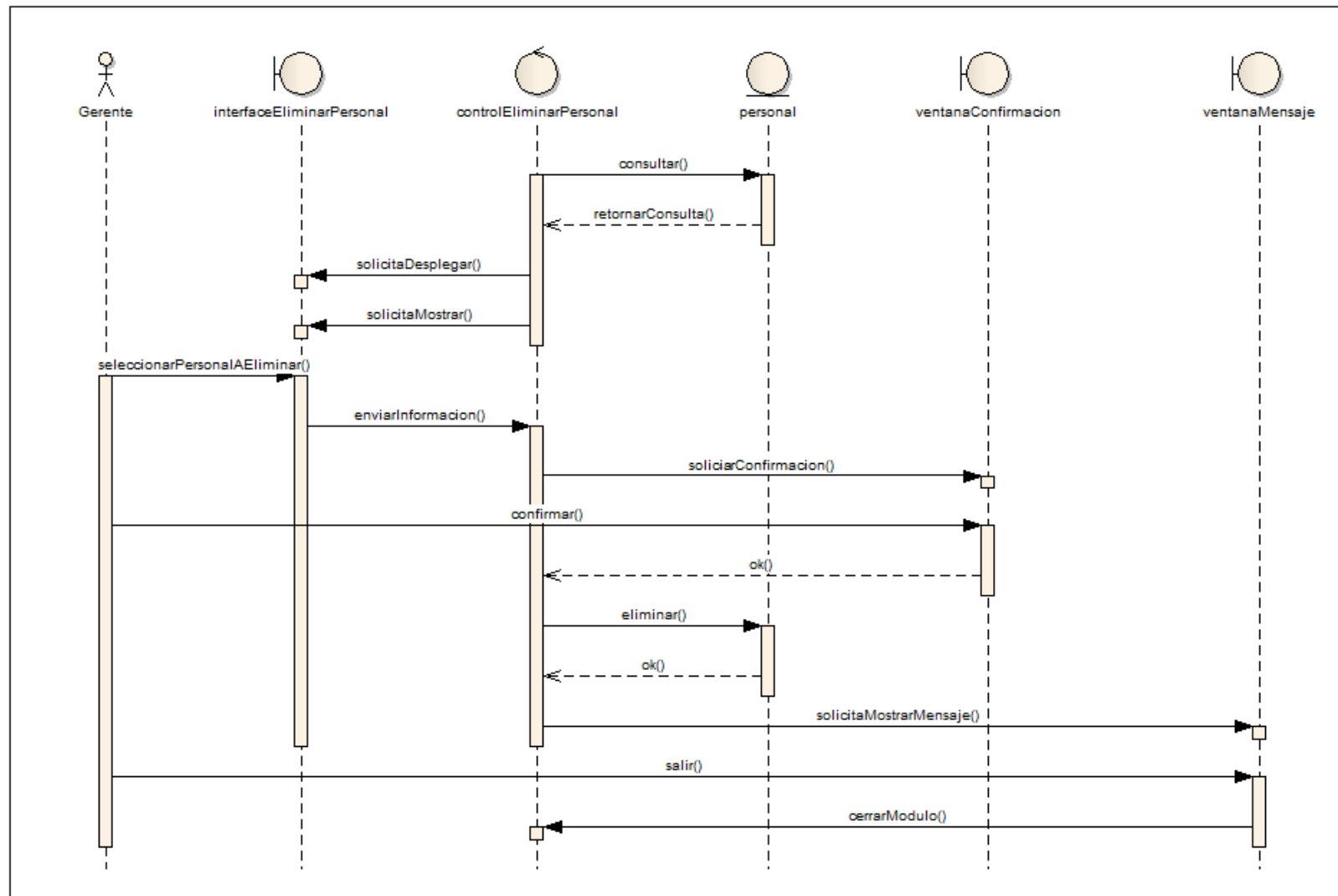
## Crear producto



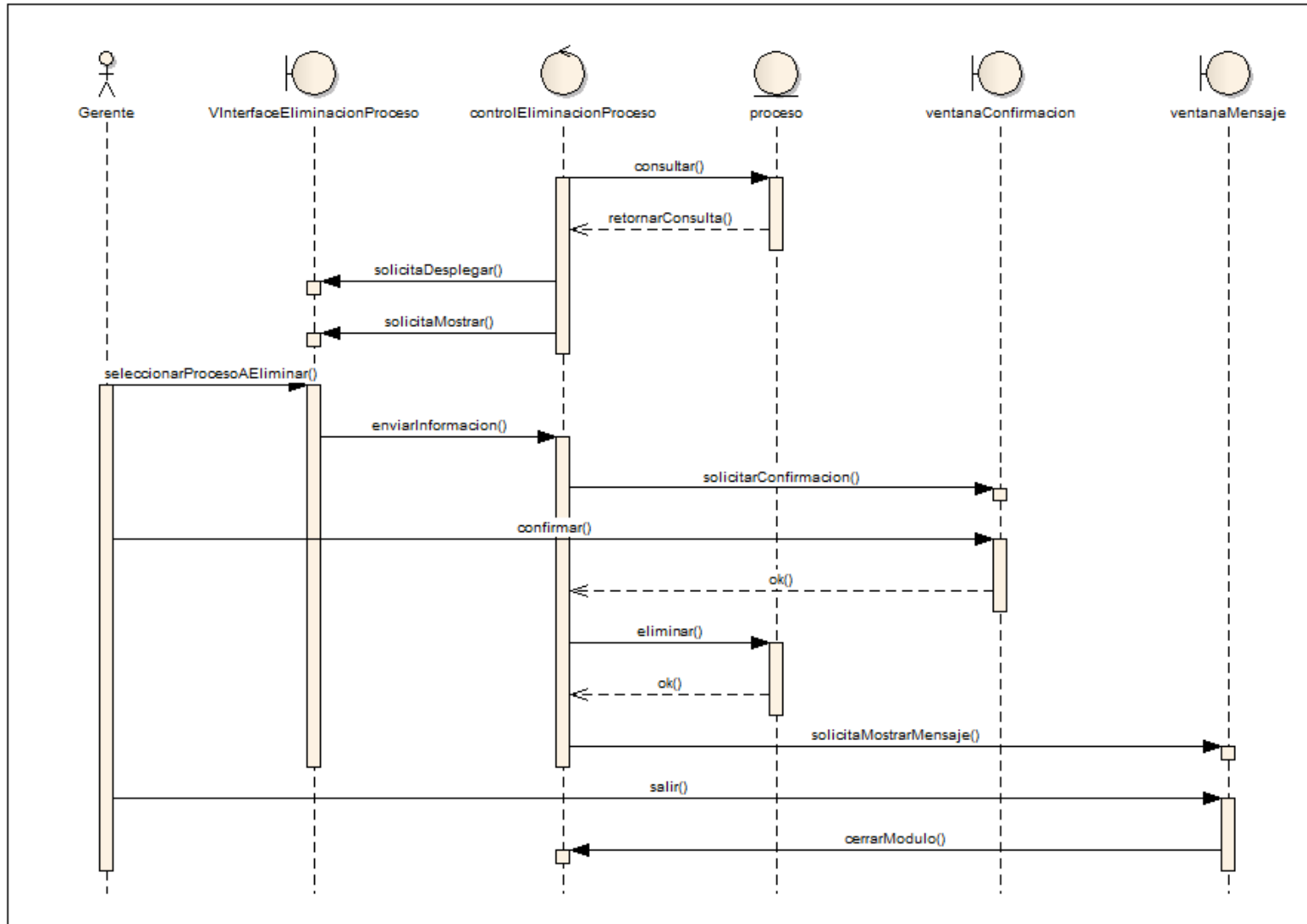
## Eliminar espacio planta



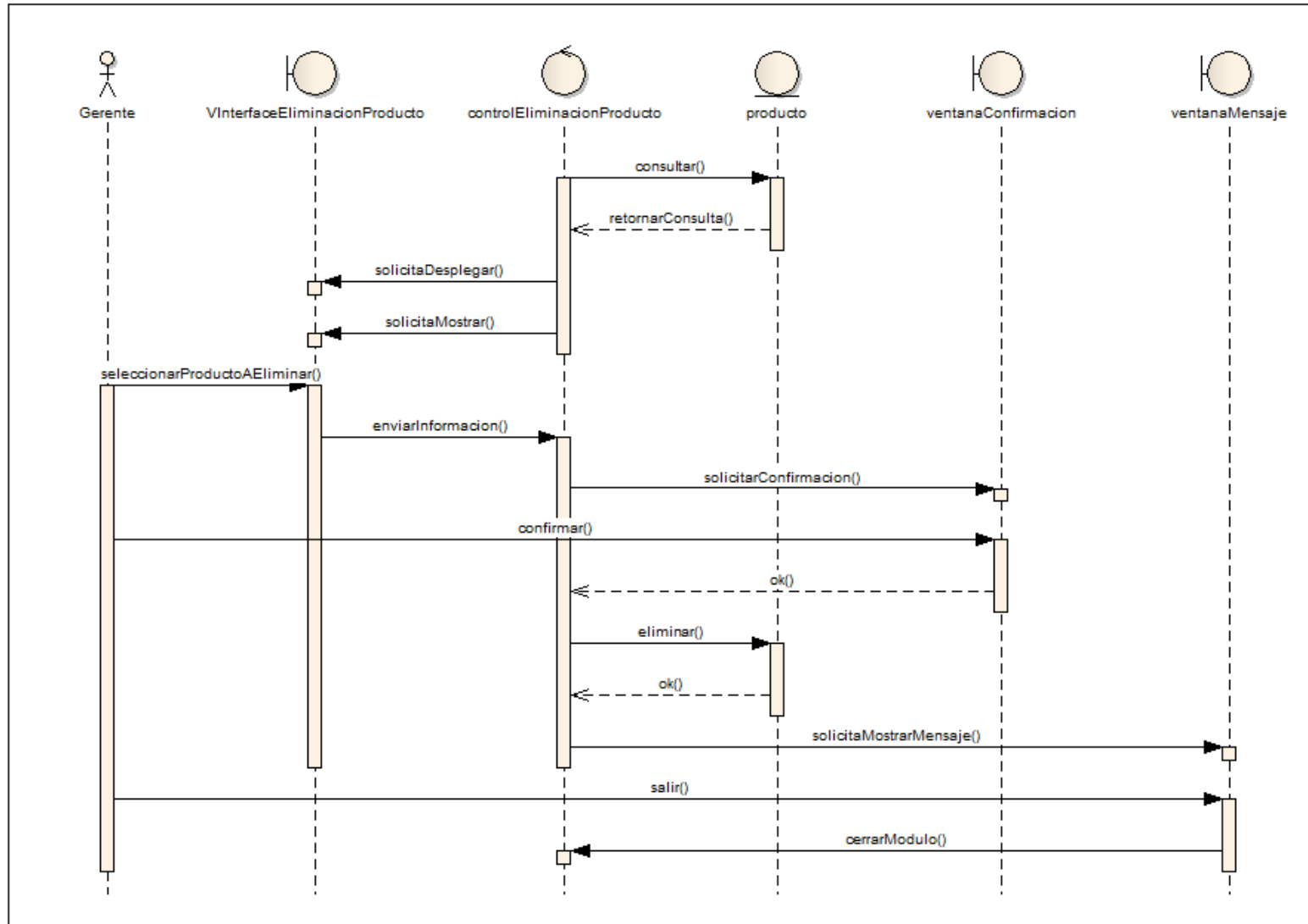
## Eliminar personal



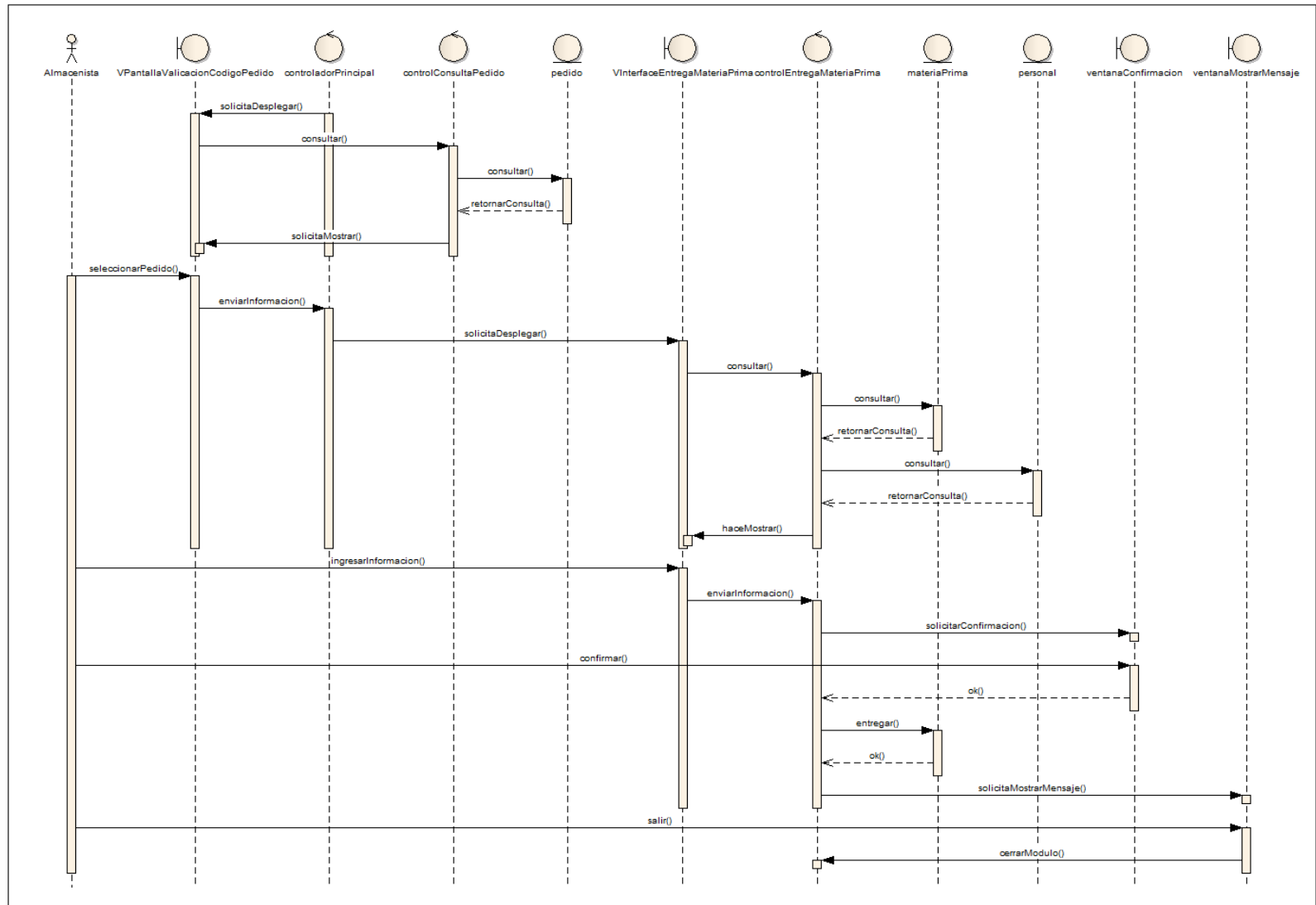
## Eliminar proceso



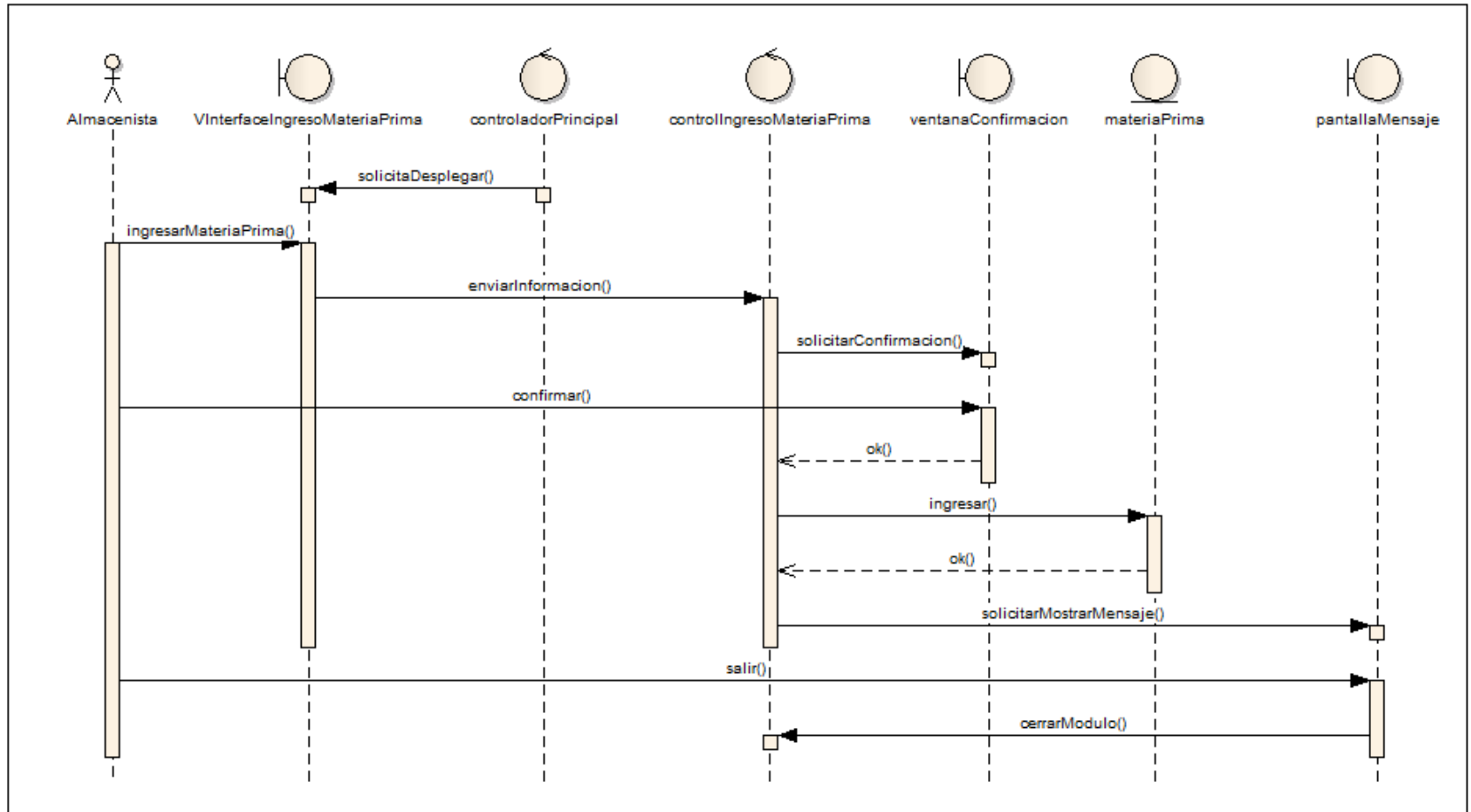
## Eliminar producto



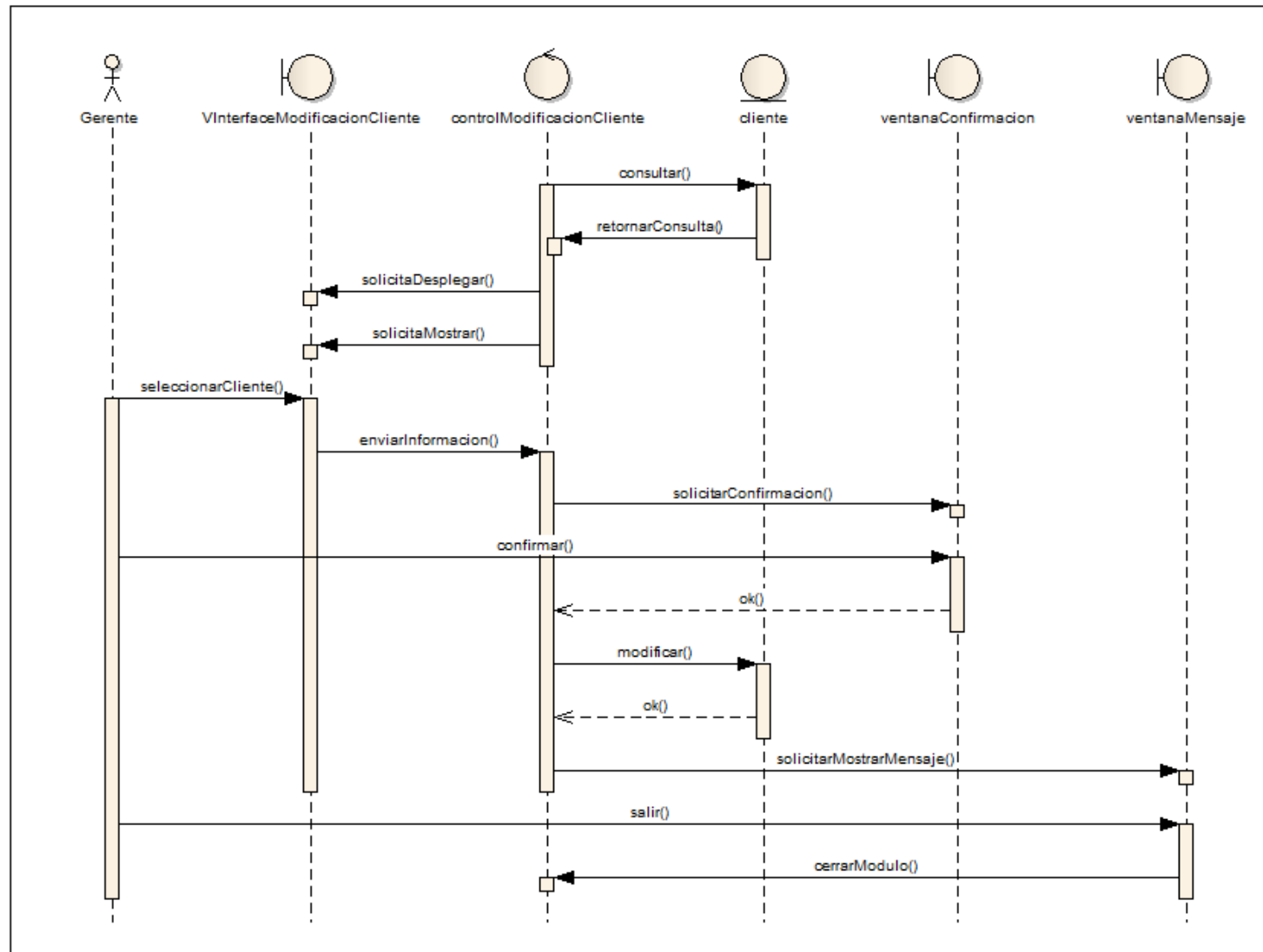
## Entregar materia prima



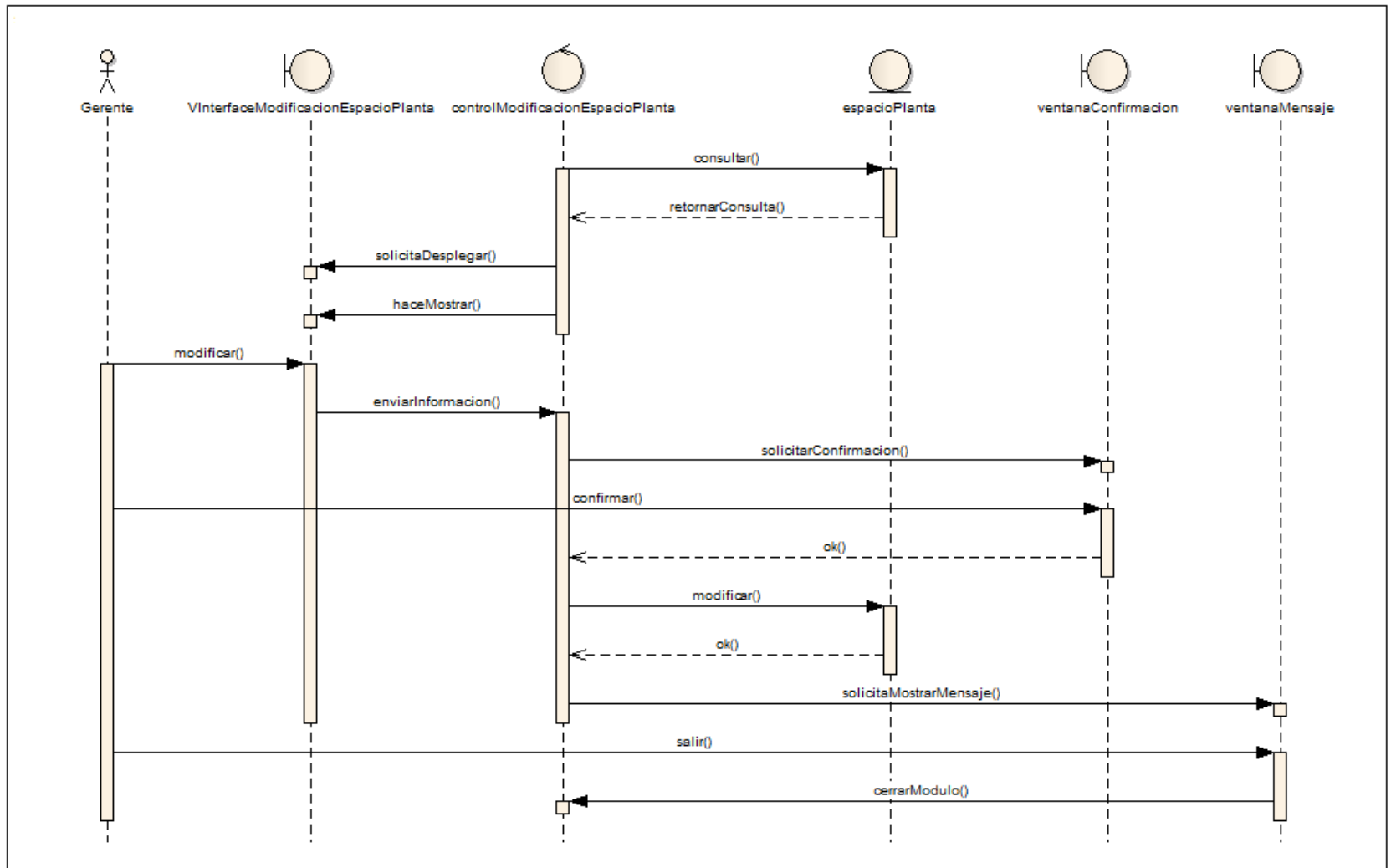
## Ingresar materia prima



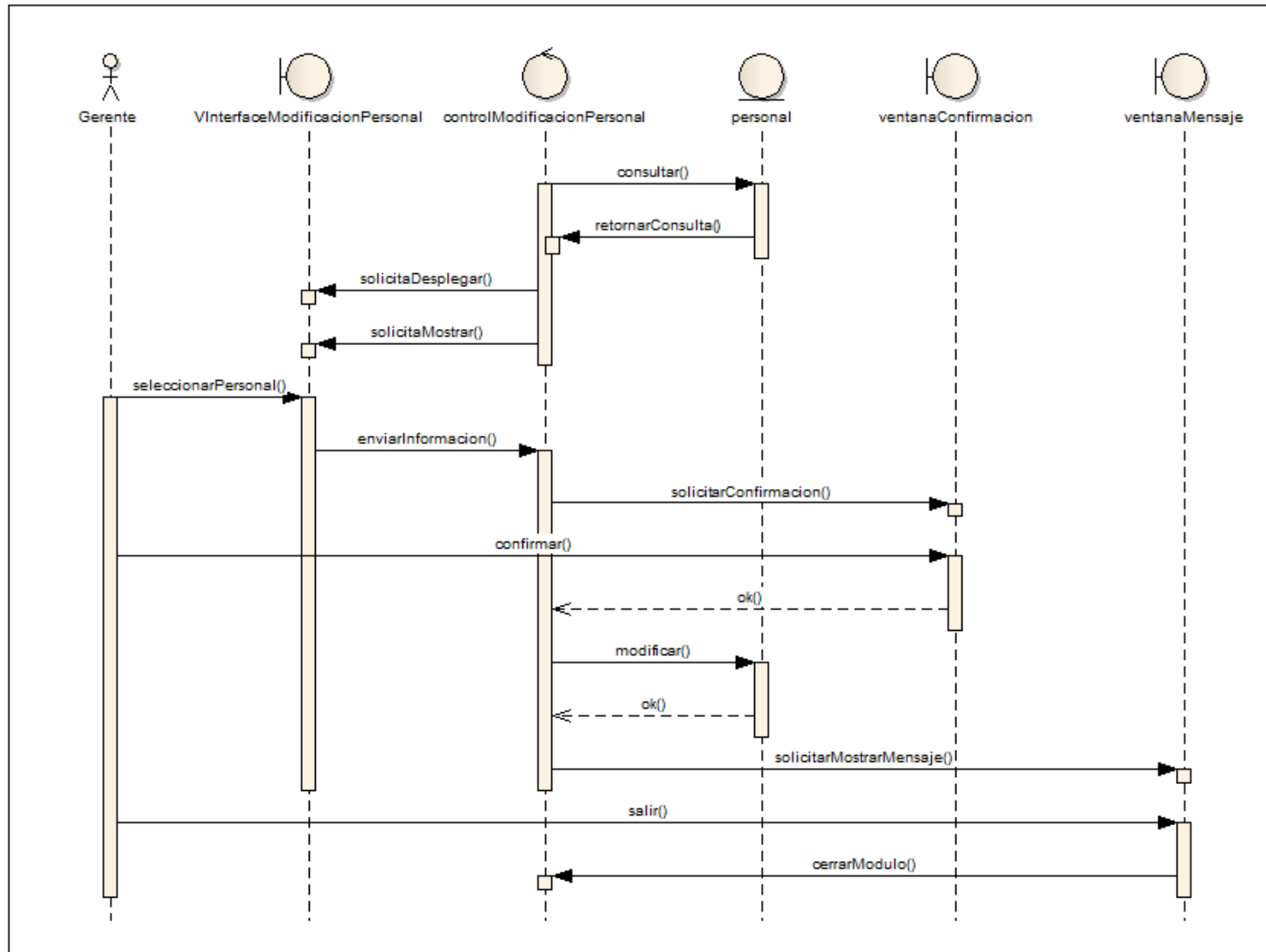
## Modificar cliente



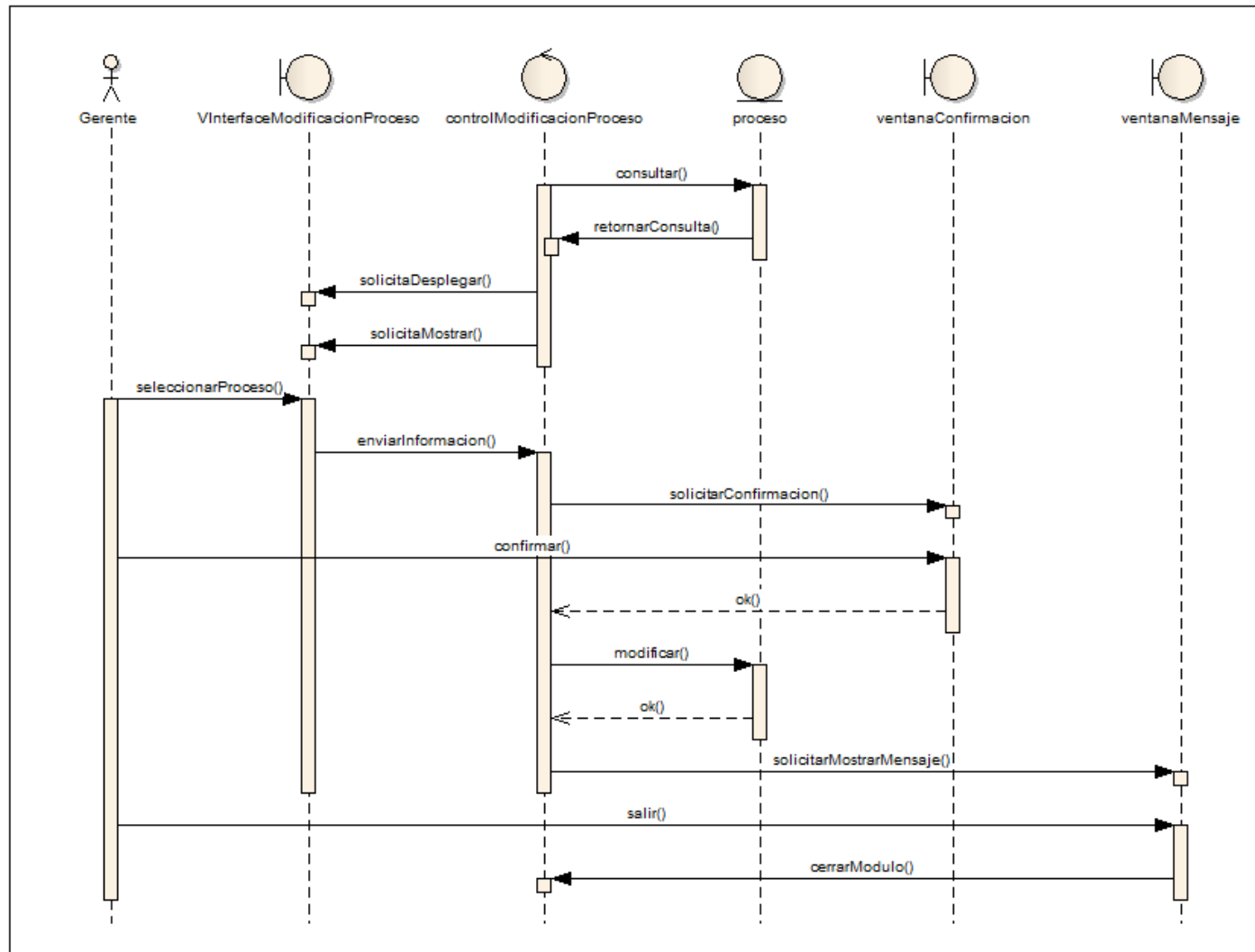
## Modificar espacio planta



## Modificar personal



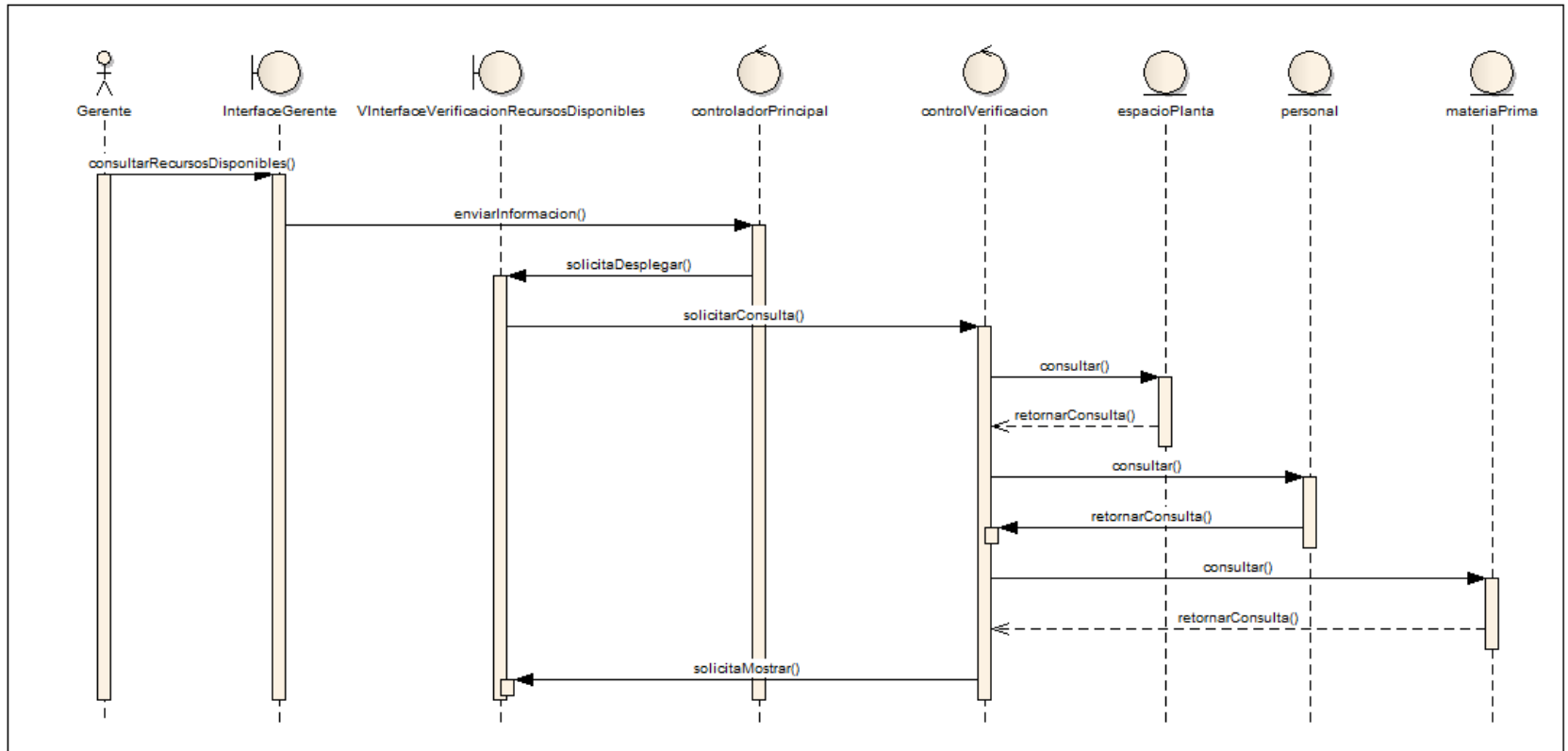
## Modificar proceso



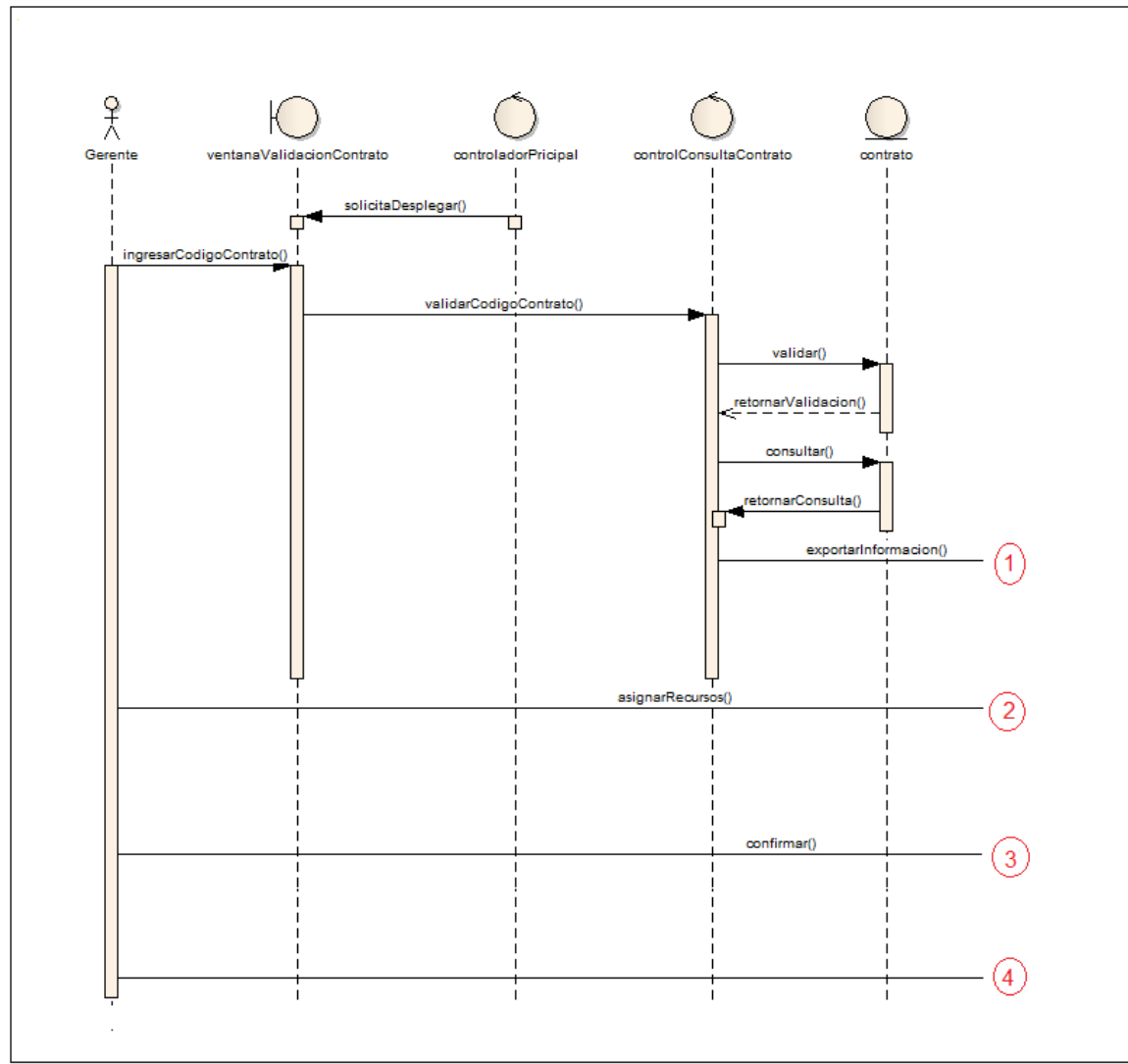
## Modificar producto



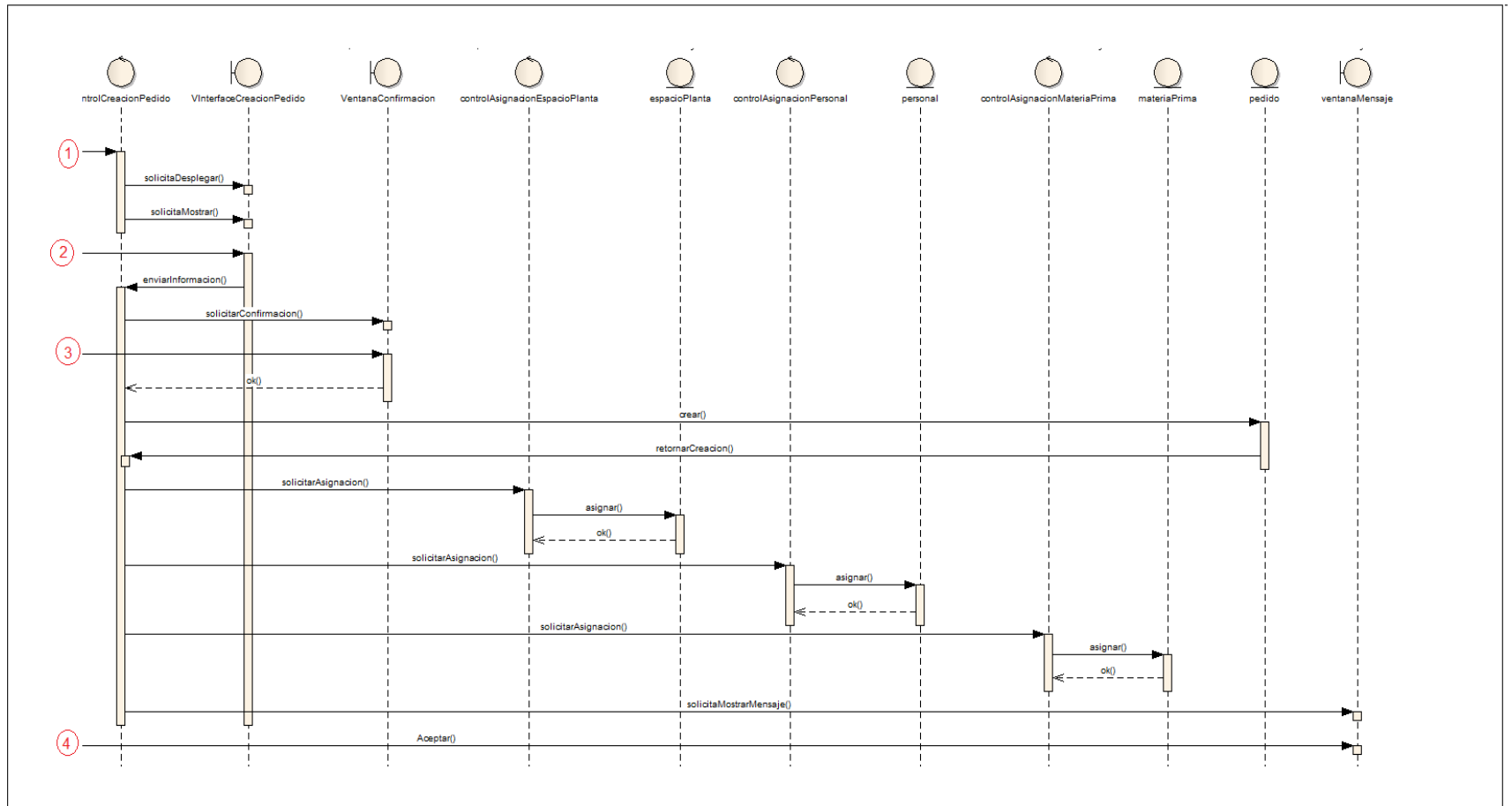
## Verificar recursos



## Crear pedido (Parte 1)



## Crear pedido (Parte 2)



## F. Narrativas de casos de uso con objetos

<b>Caso de uso</b>	Crear cotización
<b>Condición inicial</b>	
<b>Flujo principal</b>	<p>El <u>controladorPrincipal</u> hace <i>generarCodigoCotizacion</i> con <u>controlCreacionCotizacion</u>, este hace <i>consultarCodigoCotizacion</i> sobre la entidad <u>cotizacion</u> y esta devuelve el último código de cotización creado por medio del evento <i>retornarCodigoCotizacion</i>, el código se pasa finalmente al <u>controladorPrincipal</u> quien con esta información realiza el evento <i>solicitaDesplegar</i> sobre la <u>VInterfaceCreacionCotizacion</u>.</p> <p>El <u>gerente</u> hace <i>ingresarDatosCotizacion</i> sobre la <u>VInterfaceCreacionCotizacion</u>, seguido a esto la interface envía la información ingresada al <u>controlCreacionCotizacion</u> por medio del evento <i>enviarInformacion</i>, para que el controlador realice el evento <i>solicitarConfirmacion</i> sobre la <u>ventanaConfirmacion</u>. Una vez el <u>gerente</u> hace <i>confirmar</i> el <u>controlCreacionCotizacion</u> realiza el evento <i>crear</i> sobre la entidad <u>cliente</u> y posteriormente realiza <i>crear</i> sobre la entidad <u>cotizacion</u>. Por último el <u>controlCreacionCotizacion</u> <i>solicitaMostrarMensaje</i> sobre la <u>ventanaMensaje</u> para que el <u>gerente</u> pueda visualizar el la cotización creada por medio del evento <i>aceptar</i>.</p>
<b>Condición de salida</b>	Cotización creada
<b>Caso de uso</b>	Crear contrato
<b>Condición inicial</b>	Debe existir una cotización creada
<b>Flujo principal</b>	<p>El <u>controladorPrincipal</u> <i>solicitaDesplegar</i> la <u>VentataValidacionCotizacion</u>, para que el <u>gerente</u> pueda realizar la acción de <i>ingresarCodigoCotizacion</i>, este código una vez ingresado le permite a la <u>VentanaValidacionCotizacion</u> <i>validarCodigo</i> sobre el <u>controlConsultaCotizacion</u>, quien a realizar el evento <i>validar</i> sobre <u>cotizacion</u> obtendrá de regreso el resultado de la validación con el evento <i>retornarValidacion</i>, lo mismo sucede con el evento <i>consultar</i> y la respuesta de <u>cotizacion</u> con el evento <i>retornarConsulta</i>. Con esta información <u>controlConsultaCotizacion</u> procede a enviar el evento <i>exportarInformacion</i> al <u>controlCreacionContrato</u></p>

	<p>quien realiza los eventos de <i>solicitaDesplegar</i> y <i>solicitaMostrar</i> para mostrar la información resultado de la consulta de la <u>cotizacion</u> en la <u>VInterfaceCreacionContrato</u>, sobre la cual el <u>gerente</u> hará <i>ingresarDatosContrato</i> para que la interface pueda <i>enviarInformacion</i> a <u>controlCreacionContrato</u> y este pueda <i>solicitarConfirmacion</i> sobre la <u>ventanaConfirmacion</u> donde el <u>gerente</u> podrá confirmar la operación. Una vez confirmada la operación, el <u>controlCreacionContrato</u> podrá realizar el evento <i>crear</i> sobre la entidad <u>contrato</u> quien responderá con el evento <i>retornarCreacion</i> para que finalmente el <u>controlCreacionContrato</u> pueda <i>solicitarMostrarMensaje</i> sobre la <u>ventanaMensaje</u> con lo que el <u>gerente</u> pueda visualizar el <u>contrato</u> creado realizando la acción de <i>aceptar</i> sobre la <u>ventanaMensaje</u>.</p>
<b>Condición de salida</b>	Contrato creado
<b>Caso de uso</b>	Consultar cotización
<b>Condición inicial</b>	Deben existir cotizaciones creadas
<b>Flujo principal</b>	<p>El <u>controladorPrincipal</u> <i>solicitaDesplegar</i> la <u>VInterfaceConsultaCotizacion</u> sobre la cual se seleccionará la <u>cotizacion</u> a consultar, una vez desplegada esta genera el evento <i>consultar</i> sobre <u>controlConsultaCotizacion</u> y este realiza el evento <i>consultar</i> sobre la entidad <u>cotizacion</u> que devuelve un listado con todas las cotizaciones creadas, esto por medio del evento <i>retornarConsulta</i>, ya con esta información el <u>controlConsultaCotizacion</u> procede a mostrarla sobre la <u>VInterfaceConsultaCotizacion</u> por medio del evento <i>solicitaMostrar</i>. El gerente realiza <i>seleccionarCotizacion</i> sobre la interface para que esta pueda <i>enviarInformacion</i> de la <u>cotizacion</u> a consultar al <u>controlConsultaCotizacion</u> quien <i>solicitaDesplegar</i> la <u>pantallaMostrarCotizacion</u>. Para mostrar la información de la <u>cotizacion</u> la <u>pantallaMostrarCotizacion</u> realiza <i>consultar</i> sobre <u>controlConsultaCotizacion</u> y este realiza <i>consultar</i> sobre <u>cotizacion</u> quien devuelve la información consultada por medio del evento <i>retornarConsulta</i>, finalmente el <u>controlConsultaCotizacion</u> muestra la información consultada por medio del evento <i>solicitaMostrar</i>.</p>
<b>Condición de salida</b>	

<b>Caso de uso</b>	Consultar contrato
<b>Condición inicial</b>	Deben existir contratos creados
<b>Flujo principal</b>	El <u>controladorPrincipal</u> <i>solicitaDesplegar</i> la <u>VInterfaceConsultaContrato</u> sobre la cual se seleccionará el <u>contrato</u> a consultar, una vez desplegada esta genera el evento <i>consultar</i> sobre <u>controlConsultaContrato</u> y este realiza el evento <i>consultar</i> sobre la entidad <u>contrato</u> que devuelve un listado con todos los contratos creados, esto por medio del evento <i>retornarConsulta</i> , ya con esta información el <u>controlConsultaContrato</u> procede a mostrarla sobre la <u>VInterfaceConsultaContrato</u> por medio del evento <i>solicitaMostrar</i> . El gerente realiza <i>seleccionarContrato</i> sobre la interface para que esta pueda <i>enviarInformacion</i> del <u>contrato</u> al <u>controladorPrincipal</u> , quien <i>solicitaDesplegar</i> la <u>VPantallaMostrarContrato</u> para que esta pueda solicitar <i>consultar</i> al <u>controlConsultaCotizacion</u> y este realiza <i>consultar</i> sobre <u>cotizacion</u> quien devuelve la información consultada por medio del evento <i>retornarConsulta</i> , el <u>controlConsultaCotizacion</u> <i>solicitaMostrar</i> la información consultada sobre la <u>VPantallaMostrarContrato</u> quien realiza <i>consultar</i> sobre <u>controlConsultaContrato</u> para que este obtenga la información restante de <u>contrato</u> por medio del evento <i>consultar</i> . Por último el <u>controlConsultaContrato</u> <i>solicitaMostrar</i> sobre <u>pantallaMostrarContrato</u> la información restante.
<b>Condición de salida</b>	
<b>Caso de uso</b>	Consultar pedido
<b>Condición inicial</b>	Deben existir pedidos creados
<b>Flujo principal</b>	El <u>controladorPrincipal</u> <i>solicitaDesplegar</i> la <u>VInterfaceConsultaPedido</u> sobre la cual se seleccionará el <u>pedido</u> a consultar, una vez desplegada esta genera el evento <i>consultar</i> sobre <u>controlConsultaPedido</u> y este realiza el evento <i>consultar</i> sobre la entidad <u>pedido</u> que devuelve un listado con todos los pedidos creados, esto por medio del evento <i>retornarConsulta</i> , ya con esta información el <u>controlConsultaPedido</u> procede a mostrarla sobre la <u>VInterfaceConsultaPedido</u> por medio del evento <i>solicitaMostrar</i> . El gerente realiza <i>seleccionarPedido</i> sobre

	<p>la interface para que esta pueda <i>enviarInformacion</i> del <u>pedido</u> al <u>controladorPrincipal</u>, quien <i>solicitaDesplegar</i> la <u>VPantallaMostrarPedido</u>. Esta hace el evento <i>consultar</i> sobre el <u>controlConsultaAsignacion</u> para que este realice <i>consultar</i> sobre <u>espacioPlanta</u>, <u>materiaPrima</u> y <u>personal</u>, quienes retornarán información sobre su respectiva asignación para que el <u>controlConsultaAsignacion</u> pueda hacer el evento <i>solicitaMostrar</i> sobre <u>VPantallaMostrarPedido</u>. Por último esta pantalla realiza el evento <i>consultar</i> sobre el <u>controlConsultaPedido</u> para obtener la información restante del <u>pedido</u> haciendo el evento <i>consultar</i> sobre esta entidad, para que finalmente el <u>controlConsultaPedido</u> <i>solicitaMostrar</i> la información en <u>VPantallaMostrarPedido</u>.</p>
<b>Condición de salida</b>	
<b>Caso de uso</b>	Asignar personal
<b>Condición inicial</b>	Tener disponibilidad de personal y estar creando un pedido
<b>Flujo principal</b>	<p>El flujo comienza cuando un nuevo pedido se está creando y la <u>VInterfaceCreacionPedido</u> está desplegada y el <u>gerente</u> realiza <i>hacerAsignacionPersonal</i> para que la interface haga <i>solicitarDespliegue</i> sobre el <u>controlAsignacion</u>. Este control <i>solicitaDesplegar</i> la <u>VInterfaceAsignacionPersonal</u> y posteriormente realiza el evento <i>consultar</i> sobre <u>controlVerificacion</u> para que este consulte el listado de operarios sobre <u>personal</u> por medio del evento <i>consultar</i>, una vez ha recibido la información realiza el evento <i>retornarResultadoConsulta</i> a <u>controlAsignacion</u> para que este haga el evento <i>solicitaMostrar</i> a <u>controlAsignacionPersonal</u> quien finalmente muestra el resultado de la consulta sobre la <u>VInterfaceAsignacionPersonal</u> por medio del evento <i>haceMostrar</i>. El <u>gerente</u> hace la selección de operarios a asignar por medio del evento <i>seleccionarPersonal</i> sobre la <u>VInterfaceAsignacionPersonal</u>, esta información es enviada al <u>controlAsignacionPersonal</u> por el evento <i>enviarInformacion</i> para que este realice <i>solicitarConfirmacion</i> sobre <u>ventanaConfirmacion</u>. Por último el <u>gerente</u> hace <i>confirmar</i> sobre <u>ventanaConfirmacion</u>, con esto <u>controlAsignacionPersonal</u> envía el evento <i>solicitaMostrar</i> a <u>controlAsignacion</u> y esta muestra la selección de operarios sobre la</p>

	<u>VInterfaceCreacionPedido</u> con el evento <i>haceMostrar</i> .
<b>Condición de salida</b>	Personar seleccionado para asignar
<b>Caso de uso</b>	Asignar materia prima
<b>Condición inicial</b>	Tener disponibilidad de materia prima y estar creando un pedido
<b>Flujo principal</b>	<p>El flujo comienza cuando un nuevo pedido se está creando y la <u>VInterfaceCreacionPedido</u> está desplegada y el <u>gerente</u> realiza <i>hacerAsignacionMateriaPrima</i> para que la interface haga <i>solicitarDespliegue</i> sobre el <u>controlAsignacion</u>. Este control <i>solicitaDesplegar</i> la <u>VInterfaceAsignacionMateriaPrima</u> y posteriormente realiza el evento <i>consultar</i> sobre <u>controlVerificacion</u> para que este consulte el listado de materia prima sobre <u>materiaPrima</u> por medio del evento <i>consultar</i>, una vez ha recibido la información realiza el evento <i>retornarResultadoConsulta</i> a <u>controlAsignacion</u> para que este haga el evento <i>solicitaMostrar</i> a <u>controlAsignacionMateriaPrima</u> quien finalmente muestra el resultado de la consulta sobre la <u>VInterfaceAsignacionMateriaPrima</u> por medio del evento <i>haceMostrar</i>. El <u>gerente</u> hace la selección de <u>materiaPrima</u> a asignar por medio del evento <i>seleccionarMateriaPrima</i> sobre la <u>VInterfaceAsignacionMateriaPrima</u>, esta información es enviada al <u>controlAsignacionMateriaPrima</u> por el evento <i>enviarInformacion</i> para que este realice <i>solicitarConfirmacion</i> sobre <u>ventanaConfirmacion</u>. Por último el <u>gerente</u> hace <i>confirmar</i> sobre <u>ventanaConfirmacion</u>, con esto <u>controlAsignacionMateriaPrima</u> envía el evento <i>solicitaMostrar</i> a <u>controlAsignacion</u> y esta muestra la selección de materia prima sobre la <u>VInterfaceCreacionPedido</u> con el evento <i>haceMostrar</i>.</p>
<b>Condición de salida</b>	Materia prima seleccionado para asignar
<b>Caso de uso</b>	Verificar recursos disponibles
<b>Condición inicial</b>	
<b>Flujo principal</b>	Una vez desplegada la <u>interfaceGerente</u> , el <u>gerente</u> solicita <i>consultarRecursosDisponibles</i> sobre dicha

	interfaz, una vez hecho esto realiza <i>enviarInformacion</i> al <u>controladorPrincipal</u> quien <i>solicitaDesplegar</i> la <u>VInterfaceVerificacionRecursosDisponibles</u> , la cual hace <i>solicitarConsulta</i> sobre <u>controlVerificacion</u> el cual es el encargado de traer la información de disponibilidad de <u>espacioPlanta</u> , <u>personal</u> y <u>materiaPrima</u> por medio del evento <i>consultar</i> . Una vez obtenida la información el <u>controlVerificacion</u> se encarga de <i>solicitarMostrar</i> sobre la <u>VInterfaceVerificacionRecursosDisponibles</u> , la cual muestra el resultado de la consulta.
<b>Condición de salida</b>	
<b>Caso de uso</b>	Crear espacio planta
<b>Condición inicial</b>	
<b>Flujo principal</b>	El <u>controladorPrincipal</u> <i>solicitaDesplegar</i> la <u>VInterfaceCreacionEspacioPlanta</u> , sobre la cual se hará el ingreso de los datos de la nueva zona por parte del <u>gerente</u> con el evento <i>ingresarDatos</i> , una vez hecho esto, la interfaz realiza <i>enviarInformacion</i> a <u>controlAñadirEspacioPlanta</u> , este controlador es el encargado de <i>solicitarConfirmacion</i> a la <u>ventanaConfirmacion</u> sobre la cual el <u>gerente</u> hará <i>confirmar</i> . Una vez confirmada la operación, el <u>controlAñadirEspacioPlanta</u> solicita <i>crear</i> sobre <u>espacioPlanta</u> quien devuelve <i>retornarCreacion</i> indicando que la creación se ha hecho de manera satisfactoria, esta confirmación es mostrada en la <u>ventanaMensaje</u> en respuesta al evento <i>solicitaMostrarMensaje</i> del <u>controlAñadirEspacioPlanta</u> . Por último el <u>gerente</u> realiza <i>salir</i> sobre <u>ventanaMensaje</u> quien le indica al <u>controlAñadirEspacioPlanta</u> la acción de <i>cerrarModulo</i> .
<b>Condición de salida</b>	Espacio en planta ha sido creado
<b>Caso de uso</b>	Crear espacio planta
<b>Condición inicial</b>	
<b>Flujo principal</b>	El <u>controladorPrincipal</u> <i>solicitaDesplegar</i> la <u>VInterfaceCreacionEspacioPlanta</u> , sobre la cual se hará el ingreso de los datos de la nueva zona por parte del <u>gerente</u> con el evento <i>ingresarDatos</i> , una vez hecho esto, la interfaz realiza <i>enviarInformacion</i> a

	<p><u>controlAñadirEspacioPlanta</u>, este controlador es el encargado de <i>solicitarConfirmacion</i> a la <u>ventanaConfirmacion</u> sobre la cual el <u>gerente</u> hará <i>confirmar</i>. Una vez confirmada la operación, el <u>controlAñadirEspacioPlanta</u> solicita <i>crear</i> sobre <u>espacioPlanta</u> quien devuelve <i>retornarCreacion</i> indicando que la creación se ha hecho de manera satisfactoria, esta confirmación es mostrada en la <u>ventanaMensaje</u> en respuesta al evento <i>solicitaMostrarMensaje</i> del <u>controlAñadirEspacioPlanta</u>. Por último el <u>gerente</u> realiza <i>salir</i> sobre <u>ventanaMensaje</u> quien le indica al <u>controlAñadirEspacioPlanta</u> la acción de <i>cerrarModulo</i>.</p>
<b>Condición de salida</b>	Espacio en planta ha sido creado
<b>Caso de uso</b>	Crear personal
<b>Condición inicial</b>	
<b>Flujo principal</b>	<p>El <u>controladorPrincipal</u> <i>solicitaDesplegar</i> la <u>VInterfaceCreacionPersonal</u>, sobre la cual se hará el ingreso de los datos del nuevo empleado por parte del <u>gerente</u> con el evento <i>ingresarDatosPersonal</i>, una vez hecho esto, la interfaz realiza <i>enviarInformacion</i> a <u>controlAñadirPersonal</u>, este controlador es el encargado de <i>solicitarConfirmacion</i> a la <u>ventanaConfirmacion</u> sobre la cual el <u>gerente</u> hará <i>confirmar</i>. Una vez confirmada la operación, el <u>controlAñadirPersonal</u> solicita <i>crear</i> sobre <u>personal</u> quien devuelve <i>retornarCreacion</i> indicando que la creación se ha hecho de manera satisfactoria, esta confirmación es mostrada en la <u>ventanaMensaje</u> en respuesta al evento <i>solicitaMostrarMensaje</i> del <u>controlAñadirPersonal</u>. Por último el <u>gerente</u> realiza <i>salir</i> sobre <u>ventanaMensaje</u> quien le indica al <u>controlAñadirPersonal</u> la acción de <i>cerrarModulo</i>.</p>
<b>Condición de salida</b>	Personal ha sido creado
<b>Caso de uso</b>	Crear proceso
<b>Condición inicial</b>	
<b>Flujo principal</b>	<p>El <u>controladorPrincipal</u> <i>solicitaDesplegar</i> la <u>VInterfaceCreacionProceso</u>, sobre la cual se hará el ingreso de los datos del nuevo proceso por parte del <u>gerente</u> con el evento <i>ingresarDatosProceso</i>, una vez</p>

	hecho esto, la interfaz realiza <i>enviarInformacion</i> a <u>controlCreacionProceso</u> , este controlador es el encargado de <i>solicitarConfirmacion</i> a la <u>ventanaConfirmacion</u> sobre la cual el <u>gerente</u> hará <i>confirmar</i> . Una vez confirmada la operación, el <u>controlCreacionProceso</u> solicita <i>crear</i> sobre <u>proceso</u> quien devuelve <i>retornarCreacion</i> indicando que la creación se ha hecho de manera satisfactoria, esta confirmación es mostrada en la <u>ventanaMensaje</u> en respuesta al evento <i>solicitaMostrarMensaje</i> del <u>controlCreacionProceso</u> . Por último el <u>gerente</u> realiza <i>salir</i> sobre <u>ventanaMensaje</u> quien le indica al <u>controlCreacionProceso</u> la acción de <i>cerrarModulo</i> .
<b>Condición de salida</b>	Proceso ha sido creado
<b>Caso de uso</b>	Crear proceso
<b>Condición inicial</b>	
<b>Flujo principal</b>	El <u>controladorPrincipal</u> <i>solicitaDesplegar</i> la <u>VInterfaceCreacionProducto</u> , sobre la cual se hará el ingreso de los datos del nuevo producto por parte del <u>gerente</u> con el evento <i>ingresarDatosProducto</i> , una vez hecho esto, la interfaz realiza <i>enviarInformacion</i> a <u>controlCreacionProducto</u> , este controlador es el encargado de <i>solicitarConfirmacion</i> a la <u>ventanaConfirmacion</u> sobre la cual el <u>gerente</u> hará <i>confirmar</i> . Una vez confirmada la operación, el <u>controlCreacionProducto</u> solicita <i>crear</i> sobre <u>producto</u> quien devuelve <i>retornarCreacion</i> indicando que la creación se ha hecho de manera satisfactoria, esta confirmación es mostrada en la <u>ventanaMensaje</u> en respuesta al evento <i>solicitaMostrarMensaje</i> del <u>controlCreacionProducto</u> . Por último el <u>gerente</u> realiza <i>salir</i> sobre <u>ventanaMensaje</u> quien le indica al <u>controlCreacionProducto</u> la acción de <i>cerrarModulo</i> .
<b>Condición de salida</b>	Producto ha sido creado
<b>Caso de uso</b>	Modificar cliente
<b>Condición inicial</b>	Deben existir clientes almacenados en la base de datos
<b>Flujo principal</b>	El <u>controlModificacionCliente</u> realiza <i>consultar</i> sobre <u>cliente</u> que devuelve el evento <i>retornarConsulta</i> para pasarle al control el listado de todos los clientes en el

	<p>sistema, una vez esta información es consultada, el <u>controlModificacionCliente</u> <i>solicitaDesplegar</i> la <u>VInterfaceModificacionCliente</u> sobre la cual mostrará la información por medio del evento <i>solicitaMostrar</i>. El <u>gerente</u> realiza <i>seleccionarCliente</i> sobre la interfaz y esta por medio de <i>enviarInformacion</i> la pasa al <u>controlModificacionCliente</u> quien realiza <i>solicitarConfirmacion</i> a <u>ventanaConfirmacion</u> para que el <u>gerente</u> pueda <i>confirmar</i> la operación. Una vez hecho esto el <u>controlModificacionCliente</u> realiza el evento <i>modificar</i> sobre <u>cliente</u> quien confirma la modificación por medio del evento <i>ok</i>, finalmente el <u>controlModificacionCliente</u> hace <i>solicitarMostarMensaje</i> sobre <u>ventanaMensaje</u> para que el <u>gerente</u> haga <i>salir</i> y se le informe al <u>controlModificacionCliente</u> que puede <i>cerrarModulo</i>.</p>
<b>Condición de salida</b>	Cliente ha sido modificado
<b>Caso de uso</b>	Modificar personal
<b>Condición inicial</b>	Deben existir personal almacenado en la base de datos
<b>Flujo principal</b>	<p>El <u>controlModificacionPersonal</u> realiza <i>consultar</i> sobre <u>personal</u> que devuelve el evento <i>retornarConsulta</i> para pasarle al control el listado de todos los empleados en el sistema, una vez esta información es consultada, el <u>controlModificacionPersonal</u> <i>solicitaDesplegar</i> la <u>VInterfaceModificacionPersonal</u> sobre la cual mostrará la información por medio del evento <i>solicitaMostrar</i>. El <u>gerente</u> realiza <i>seleccionarPersonal</i> sobre la interfaz y esta por medio de <i>enviarInformacion</i> la pasa al <u>controlModificacionPersonal</u> quien realiza <i>solicitarConfirmacion</i> a <u>ventanaConfirmacion</u> para que el <u>gerente</u> pueda <i>confirmar</i> la operación. Una vez hecho esto el <u>controlModificacionPersonal</u> realiza el evento <i>modificar</i> sobre <u>personal</u> quien confirma la modificación por medio del evento <i>ok</i>, finalmente el <u>controlModificacionPersonal</u> hace <i>solicitarMostarMensaje</i> sobre <u>ventanaMensaje</u> para que el <u>gerente</u> haga <i>salir</i> y se le informe al <u>controlModificacionPersonal</u> que puede <i>cerrarModulo</i>.</p>
<b>Condición de salida</b>	Personal ha sido modificado
<b>Caso de uso</b>	Modificar proceso
<b>Condición</b>	Deben existir procesos almacenados en la base de datos

<b>inicial</b>	
<b>Flujo principal</b>	El <u>controlModificacionProceso</u> realiza <i>consultar</i> sobre <u>proceso</u> que devuelve el evento <i>retornarConsulta</i> para pasarle al control el listado de todos los procesos en el sistema, una vez esta información es consultada, el <u>controlModificacionProceso</u> <i>solicitaDesplegar</i> la <u>VInterfaceModificacionProceso</u> sobre la cual mostrará la información por medio del evento <i>solicitaMostrar</i> . El <u>gerente</u> realiza <i>seleccionarProceso</i> sobre la interfaz y esta por medio de <i>enviarInformacion</i> la pasa al <u>controlModificacionProceso</u> quien realiza <i>solicitarConfirmacion</i> a <u>ventanaConfirmacion</u> para que el <u>gerente</u> pueda <i>confirmar</i> la operación. Una vez hecho esto el <u>controlModificacionProceso</u> realiza el evento <i>modificar</i> sobre <u>proceso</u> quien confirma la modificación por medio del evento <i>ok</i> , finalmente el <u>controlModificacionProceso</u> hace <i>solicitarMostarMensaje</i> sobre <u>ventanaMensaje</u> para que el <u>gerente</u> haga <i>salir</i> y se le informe al <u>controlModificacionProceso</u> que puede <i>cerrarModulo</i> .
<b>Condición de salida</b>	Proceso ha sido modificado
<b>Caso de uso</b>	Modificar producto
<b>Condición inicial</b>	Deben existir productos almacenados en la base de datos
<b>Flujo principal</b>	El <u>controlModificacionProducto</u> realiza <i>consultar</i> sobre <u>producto</u> que devuelve el evento <i>retornarConsulta</i> para pasarle al control el listado de todos los productos en el sistema, una vez esta información es consultada, el <u>controlModificacionProducto</u> <i>solicitaDesplegar</i> la <u>VInterfaceModificacionProducto</u> sobre la cual mostrará la información por medio del evento <i>solicitaMostrar</i> . El <u>gerente</u> realiza <i>seleccionarProducto</i> sobre la interfaz y esta por medio de <i>enviarInformacion</i> la pasa al <u>controlModificacionProducto</u> quien realiza <i>solicitarConfirmacion</i> a <u>ventanaConfirmacion</u> para que el <u>gerente</u> pueda <i>confirmar</i> la operación. Una vez hecho esto el <u>controlModificacionProducto</u> realiza el evento <i>modificar</i> sobre <u>producto</u> quien confirma la modificación por medio del evento <i>ok</i> , finalmente el <u>controlModificacionProducto</u> hace <i>solicitarMostarMensaje</i> sobre <u>ventanaMensaje</u> para que el <u>gerente</u> haga <i>salir</i> y se le informe al <u>controlModificacionProducto</u> que puede <i>cerrarModulo</i> .
<b>Condición de</b>	Proceso ha sido modificado

<b>salida</b>	
<b>Caso de uso</b>	Modificar espacio planta
<b>Condición inicial</b>	Deben existir zonas almacenadas en la base de datos
<b>Flujo principal</b>	El <u>controlModificacionEspacioPlanta</u> realiza <i>consultar</i> sobre <u>espacioPlanta</u> que devuelve el evento <i>retornarConsulta</i> para pasarle al control el listado de todas las zonas en el sistema, una vez esta información es consultada, el <u>controlModificacionEspacioPlanta</u> <i>solicitaDesplegar</i> la <u>VInterfaceModificacionEspacioPlanta</u> sobre la cual mostrará la información por medio del evento <i>solicitaMostrar</i> . El <u>gerente</u> realiza <i>modificar</i> sobre la interfaz y esta por medio de <i>enviarInformacion</i> la pasa al <u>controlModificacionEspacioPlanta</u> quien realiza <i>solicitarConfirmacion</i> a <u>ventanaConfirmacion</u> para que el <u>gerente</u> pueda <i>confirmar</i> la operación. Una vez hecho esto el <u>controlModificacionEspacioPlanta</u> realiza el evento <i>modificar</i> sobre <u>espacioPlanta</u> quien confirma la modificación por medio del evento <i>ok</i> , finalmente el <u>controlModificacionEspacioPlanta</u> hace <i>solicitarMostarMensaje</i> sobre <u>ventanaMensaje</u> para que el <u>gerente</u> haga <i>salir</i> y se le informe al <u>controlModificacionEspacioPlanta</u> que puede <i>cerrarModulo</i> .
<b>Condición de salida</b>	Espacio planta ha sido modificado
<b>Caso de uso</b>	Eliminar espacio planta
<b>Condición inicial</b>	Debe existir zonas almacenadas en el sistema
<b>Flujo principal</b>	El <u>controlEliminacionEspacioPlanta</u> realiza <i>consultar</i> sobre <u>espacioPlanta</u> que devuelve el evento <i>retornarConsulta</i> para pasarle al control el listado de todas las zonas en el sistema, una vez esta información es consultada, el <u>controlEliminacionEspacioPlanta</u> <i>solicitaDesplegar</i> la <u>VInterfaceEliminacionEspacioPlanta</u> sobre la cual mostrará la información por medio del evento <i>solicitaMostrar</i> . El <u>gerente</u> realiza <i>seleccionarEspacioPlantaAEliminar</i> sobre la interfaz y esta por medio de <i>enviarInformacion</i> la pasa al <u>controlEliminacionEspacioPlanta</u> quien realiza <i>solicitarConfirmacion</i> a <u>ventanaConfirmacion</u> para que el <u>gerente</u> pueda <i>confirmar</i> la operación. Una vez hecho esto

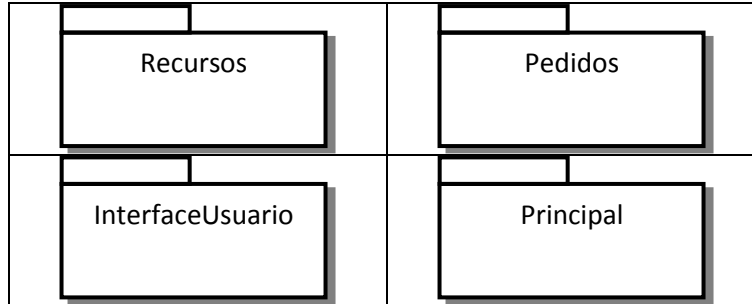
	el <u>controlEliminacionEspacioPlanta</u> realiza el evento <i>eliminar</i> sobre <u>espacioPlanta</u> quien confirma la eliminación por medio del evento <i>ok</i> , finalmente el <u>controlEliminacionEspacioPlanta</u> hace <i>solicitarMostarMensaje</i> sobre <u>ventanaMensaje</u> para que el <u>gerente</u> haga <i>salir</i> y se le informe al <u>controlEliminacionEspacioPlanta</u> que puede <i>cerrarModulo</i> .
<b>Condición de salida</b>	Espacio en planta ha sido eliminado
<b>Caso de uso</b>	Eliminar personal
<b>Condición inicial</b>	Debe existir empleados almacenados en el sistema
<b>Flujo principal</b>	El <u>controlEliminacionPersonal</u> realiza <i>consultar</i> sobre <u>personal</u> que devuelve el evento <i>retornarConsulta</i> para pasarle al control el listado de todas los empleados en el sistema, una vez esta información es consultada, el <u>controlEliminacionPersonal</u> <i>solicitaDesplegar</i> la <u>VInterfaceEliminacionPersonal</u> sobre la cual mostrará la información por medio del evento <i>solicitaMostrar</i> . El <u>gerente</u> realiza <i>seleccionarPersonalAEliminar</i> sobre la interfaz y esta por medio de <i>enviarInformacion</i> la pasa al <u>controlEliminacionPersonal</u> quien realiza <i>solicitarConfirmacion</i> a <u>ventanaConfirmacion</u> para que el <u>gerente</u> pueda <i>confirmar</i> la operación. Una vez hecho esto el <u>controlEliminacionPersonal</u> realiza el evento <i>eliminar</i> sobre <u>personal</u> quien confirma la eliminación por medio del evento <i>ok</i> , finalmente el <u>controlEliminacionPersonal</u> hace <i>solicitarMostarMensaje</i> sobre <u>ventanaMensaje</u> para que el <u>gerente</u> haga <i>salir</i> y se le informe al <u>controlEliminacionPersonal</u> que puede <i>cerrarModulo</i> .
<b>Condición de salida</b>	Personal ha sido eliminado
<b>Caso de uso</b>	Eliminar proceso
<b>Condición inicial</b>	Debe existir proceso almacenados en el sistema
<b>Flujo principal</b>	El <u>controlEliminacionProceso</u> realiza <i>consultar</i> sobre <u>proceso</u> que devuelve el evento <i>retornarConsulta</i> para pasarle al control el listado de todas los procesos en el sistema, una vez esta información es consultada, el <u>controlEliminacionProceso</u> <i>solicitaDesplegar</i> la <u>VInterfaceEliminacionProceso</u> sobre la cual mostrará la información por medio del evento <i>solicitaMostrar</i> . El

	<p><u>gerente</u> realiza <i>seleccionarProcesoAEliminar</i> sobre la interfaz y esta por medio de <i>enviarInformacion</i> la pasa al <u>controlEliminacionProceso</u> quien realiza <i>solicitarConfirmacion</i> a <u>ventanaConfirmacion</u> para que el <u>gerente</u> pueda <i>confirmar</i> la operación. Una vez hecho esto el <u>controlEliminacionProceso</u> realiza el evento <i>eliminar</i> sobre <u>proceso</u> quien confirma la eliminación por medio del evento <i>ok</i>, finalmente el <u>controlEliminacionProceso</u> hace <i>solicitarMostarMensaje</i> sobre <u>ventanaMensaje</u> para que el <u>gerente</u> haga <i>salir</i> y se le informe al <u>controlEliminacionProceso</u> que puede <i>cerrarModulo</i>.</p>
<b>Condición de salida</b>	Proceso ha sido eliminado
<b>Caso de uso</b>	Eliminar producto
<b>Condición inicial</b>	Debe existir productos almacenados en el sistema
<b>Flujo principal</b>	<p>El <u>controlEliminacionProducto</u> realiza <i>consultar</i> sobre <u>producto</u> que devuelve el evento <i>retornarConsulta</i> para pasarle al control el listado de todas los productos en el sistema, una vez esta información es consultada, el <u>controlEliminacionProducto</u> <i>solicitaDesplegar</i> la <u>VInterfaceEliminacionProducto</u> sobre la cual mostrará la información por medio del evento <i>solicitaMostrar</i>. El <u>gerente</u> realiza <i>seleccionarProductoAEliminar</i> sobre la interfaz y esta por medio de <i>enviarInformacion</i> la pasa al <u>controlEliminacionProducto</u> quien realiza <i>solicitarConfirmacion</i> a <u>ventanaConfirmacion</u> para que el <u>gerente</u> pueda <i>confirmar</i> la operación. Una vez hecho esto el <u>controlEliminacionProducto</u> realiza el evento <i>eliminar</i> sobre <u>producto</u> quien confirma la eliminación por medio del evento <i>ok</i>, finalmente el <u>controlEliminacionProducto</u> hace <i>solicitarMostarMensaje</i> sobre <u>ventanaMensaje</u> para que el <u>gerente</u> haga <i>salir</i> y se le informe al <u>controlEliminacionProducto</u> que puede <i>cerrarModulo</i>.</p>
<b>Condición de salida</b>	Producto ha sido eliminado
<b>Caso de uso</b>	Ingresar materia prima
<b>Condición inicial</b>	
<b>Flujo principal</b>	<p>El <u>controladorPrincipal</u> <i>solicitaDesplegar</i> la <u>VInterfaceIngresoMateriaPrima</u>, sobre la cual se hará el ingreso de los datos de la nueva materia prima por parte</p>

	<p>del <u>almacenista</u> con el evento <i>ingresarMateriaPrima</i>, una vez hecho esto, la interfaz realiza <i>enviarInformacion</i> a <u>controlIngresoMateriaPrima</u>, este controlador es el encargado de <i>solicitarConfirmacion</i> a la <u>ventanaConfirmacion</u> sobre la cual el <u>almacenista</u> hará <i>confirmar</i>. Una vez confirmada la operación, el <u>controlIngresoMateriaPrima</u> solicita <i>ingresar</i> sobre <u>materiaPrima</u> quien devuelve <i>ok</i> indicando que la creación se ha hecho de manera satisfactoria, esta confirmación es mostrada en la <u>ventanaMensaje</u> en respuesta al evento <i>solicitaMostrarMensaje</i> del <u>controlIngresoMateriaPrima</u>. Por último el <u>almacenista</u> realiza <i>salir</i> sobre <u>ventanaMensaje</u> quien le indica al <u>controlIngresoMateriaPrima</u> la acción de <i>cerrarModulo</i>.</p>
<b>Condición de salida</b>	Materia prima ha sido ingresada
<b>Caso de uso</b>	Entregar materia prima
<b>Condición inicial</b>	La materia prima y el personal deben haber sido asignados
<b>Flujo principal</b>	<p>El <u>controladorPrincipal</u> <i>solicitDesplegar</i> la <u>VPantallaValidacionCodigoPedido</u> para que esta pueda hacer el evento <i>consultar</i> sobre el <u>controlConsultaPedido</u>, quien es el encargado de consultar sobre <u>pedido</u> el código de todos los pedidos con asignaciones, una vez esta consulta se ha hecho, el <u>controlConsultaPedido</u> <i>solicitaMostar</i> esa información sobre <u>VPantallaValidacionCodigoPedido</u>, sobre la cual el <u>almacenista</u> podrá <i>seleccionarPedido</i> y permitirá a la <u>VPantallaValidacionCodigoPedido</u> <i>enviarInformacion</i> a <u>controladorPrincipal</u>. Con esta información, el <u>controladorPrincipal</u> <i>solicitaDesplegar</i> la <u>VInterfaceEntregaMateriaPrima</u>, la cual con el código de <u>pedido</u> hará <i>consultar</i> sobre el <u>controlEntregaMateriaPrima</u>, quien es el encargado de <i>consultar</i> la información de <u>materiaPrima</u> y <u>personal</u> asociada al código de pedido ingresado para que se lleve a cabo el evento <i>haceMostrar</i> con la información proveniente de la consulta. El <u>almacenista</u> hará <i>ingresarInformacion</i> sobre la interface y esta realizará <i>enviarInformacion</i> sobre el <u>controlEntregaMateriaPrima</u> quien podrá <i>solicitarConfirmacion</i> por medio de la <u>ventanaConfirmacion</u> en la cual el <u>almacenista</u> hará <i>confirmar</i> con lo que se completará la entrega de</p>

	<u>materiaPrima</u> . Para finalizar el caso de uso, el <u>controlEntregaMateriaPrima</u> realiza <u>solicitaMostrarMensaje</u> sobre la <u>ventanaMostrarMensaje</u> para que el <u>almacenista</u> pueda realizar <i>salir</i> y el controlador pueda hacer <i>cerrarModulo</i> .
<b>Condición de salida</b>	La materia prima se ha entregado

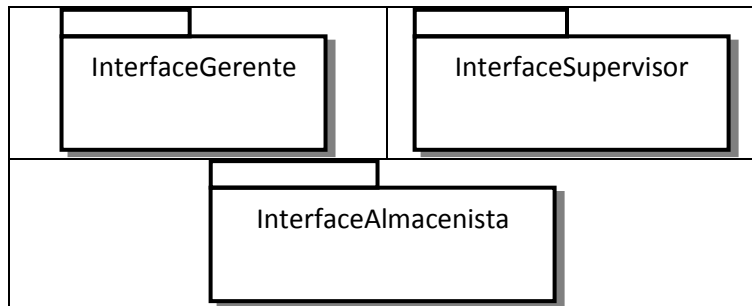
**G. Diccionario de clases agrupado por módulos.**



**Principal**

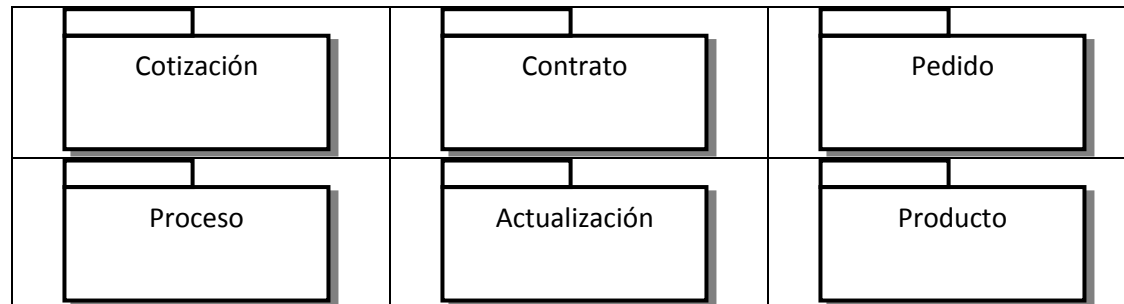
Módulo	Descripción
InterfazPrincipal	Clase de Borde. Los usuarios interactúan con el sistema a través de la interfaz principal
Manejador Principal	Clase Control. Es el encargado de desplegar la interfaz principal, en la cual el usuario hace el ingreso al sistema

**Interfaz  
Usuario**



Módulo	Descripción
InterfazGerente	Clase de Borde. Se despliegan las diferentes acciones que puede ejecutar el Gerente
InterfazSupervisor	Clase de Borde. Se despliegan las diferentes acciones que puede ejecutar el Supervisor
InterfazAlmacenista	Clase de Borde. Se despliega las diferentes acciones que puede ejecutar el Almacenista
ManejadorGerente	Clase de Control. Encargada de desplegar la interfaz del Gerente
ManejadorSupervisor	Clase de Control. Encargada de desplegar la interfaz del Supervisor
ManejadorAlmacenista	Clase de Control. Encargada de desplegar la interfaz del Almacenista

### PEDIDOS

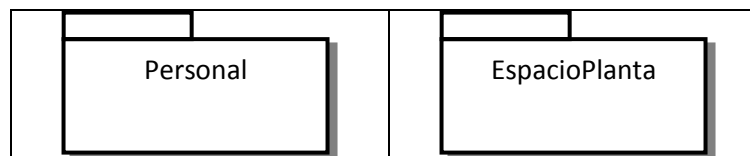


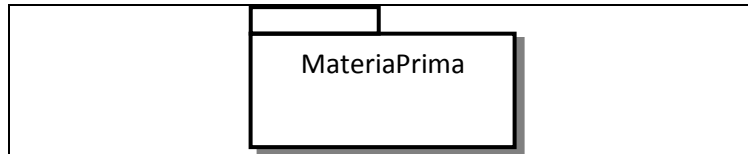
Sub-Modulo	Módulo	Descripción
Cotizacion	InterfaceCrearCotizacion	Clase de borde. Se despliega el formato para la creación de la cotización en la cual se genera un nuevo número de cotización para poder continuar con el llenado de los campos de la misma.
	ManejadorCrearCotización	Clase de Control. Encargada de habilitar los campos de la cotización, una vez se genere el número del contrato

Contrato	InterfaceCreacionContrato	Clase de Borde. Se despliega el formato para la creación del contrato en el cual, luego de validar el número de la cotización es posible continuar con el llenado de los campos del mismo.
	ManejadorCreacionContrato	Clase de Control. Encargada de habilitar los campos del contrato, una vez se valide el número de la cotización, como también de preestablecer en la interfaceCreacionContrato los valores traídos desde la cotización correspondiente.
Pedido	InterfaceCreacionPedido	Clase de Borde. Se despliega el formato para la creación del pedido en el cual después de validar el número del contrato, se habilitan los campos para ingresar las características del nuevo pedido.
	ManejadorCrearPedido	Clase de Control. Encargada de habilitar los campos del pedido, una vez validado el número del contrato
	InterfaceActualizacionPedido	Clase de Borde. Se despliega la interface de actualización de pedido, donde una vez validado el número del pedido, se muestran todas las actualizaciones disponibles, donde las que ya han sido ingresadas aparecen deshabilitadas, y aparecen habilitadas las que se pueden ingresar.
	ManejadorIngresoActualizaciones	Clase de Control. Encargada de habilitar las actualizaciones disponibles a ingresar una vez validado en número del pedido.

Actualizacion	InterfaceConsultarActualizacionesPedido	Clase de Borde. Se despliega la interfaz de consulta del pedido, donde una vez validado el número del pedido, se muestran las actualizaciones correspondientes al pedido validado.
	ManejadorConsultarActualizacionesPedido	Clase de Control. Encargada de mostrar las actualizaciones del pedido, una vez validado el número del pedido.
Proceso	ManejadorProceso	Clase de Control. Encargada de desplegar la información relacionada con cada una de las etapas en la elaboración de un pedido.
	ManejadorCreacionProceso	Clase de Control. Encargada de capturar la información ingresada en la interface creación proceso para la creación de un nuevo proceso
	InterfaceCreaciónProceso	Clase de Borde. Se despliega el formato para la creación de un proceso.
Producto	ManejadorCreaciónProducto	Clase de Control. Encargada de capturar la información ingresada en la interface creación producto para la creación de un nuevo tipo de producto
	InterfaceCreaciónProducto	Clase de Borde. Se despliega el formato para la creación de un tipo de producto.

### RECURSOS



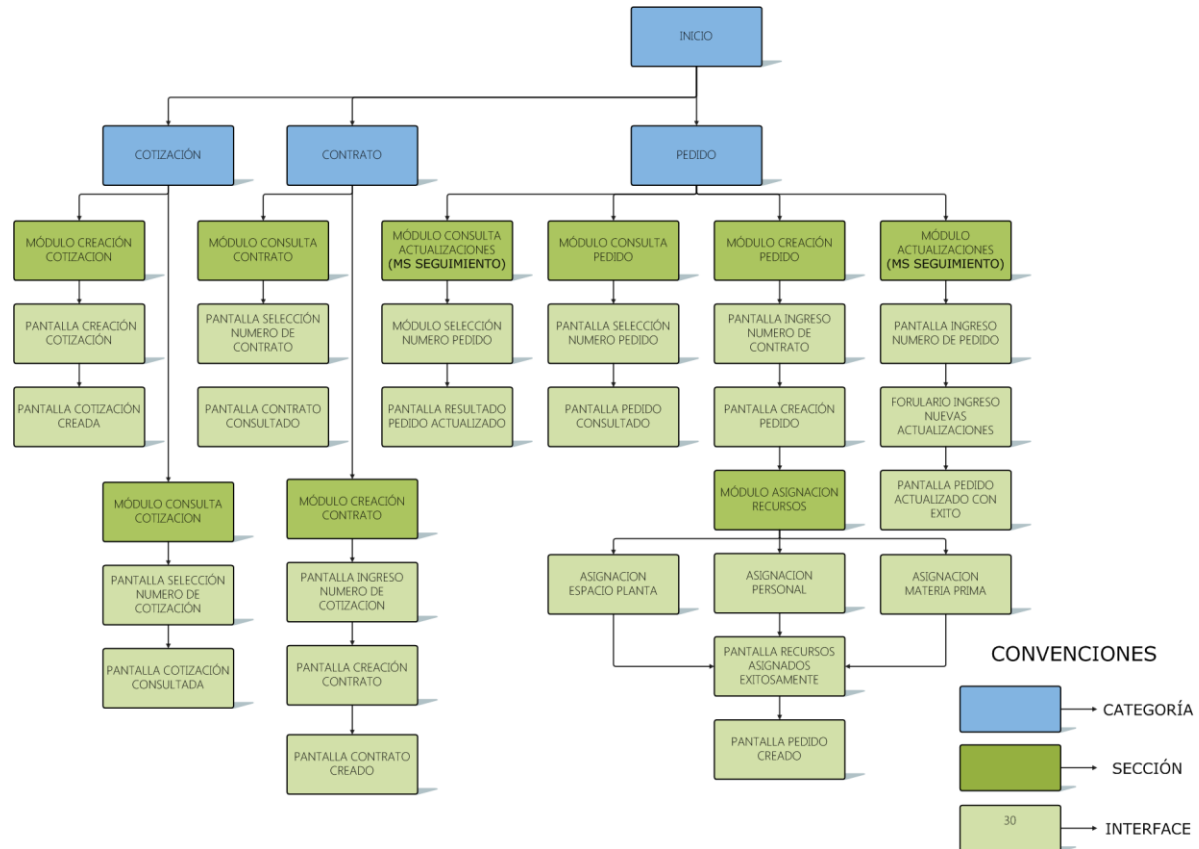


Sub-Modulo	Módulo	Descripción
Personal	InterfaceAsignacionPersonal	Clase de Borde. Se despliega el formato de asignación de personal luego haber validado el número del pedido en la interfaceCreacionPedido, se habilitan los campos para que el gerente pueda generar una nueva asignación de personal.
	ManejadorAsignacionPersonal	Clase de Control. Encargada de desplegar la interfaceAsignaciónPersonal, una vez validada el tipo de asignación elegida por el gerente.
	InterfaceVerificacionPersonal	Clase de Borde. Se despliega la interfaceVerificacionPersonal con toda la información del personal.
	ManejadorVerificacionPersonal	Clase de Control. Encargada de desplegar la interfaceVerificaciónPersonal, una vez validado el tipo de verificación elegida por el gerente.
Espacio Planta	InterfaceAsignacionEspacioPlanta	Clase de Borde. Se despliega el formato de asignación de espacio en planta luego haber validado el número del pedido en la interfaceCreacionPedido, se habilitan los campos para que el gerente pueda generar una nueva asignación de espacio en planta.
	ManejadorAsignacionEspacioPlanta	Clase de Control. Encargada de desplegar la interfaceAsignaciónEspacioPlanta, una vez validada el tipo de asignación elegida por el gerente.
	InterfaceVerificacionEspacioPlanta	Clase de Borde. Se despliega la interfaceVerificacionEspacioPlanta con toda la información del Espacio en Planta

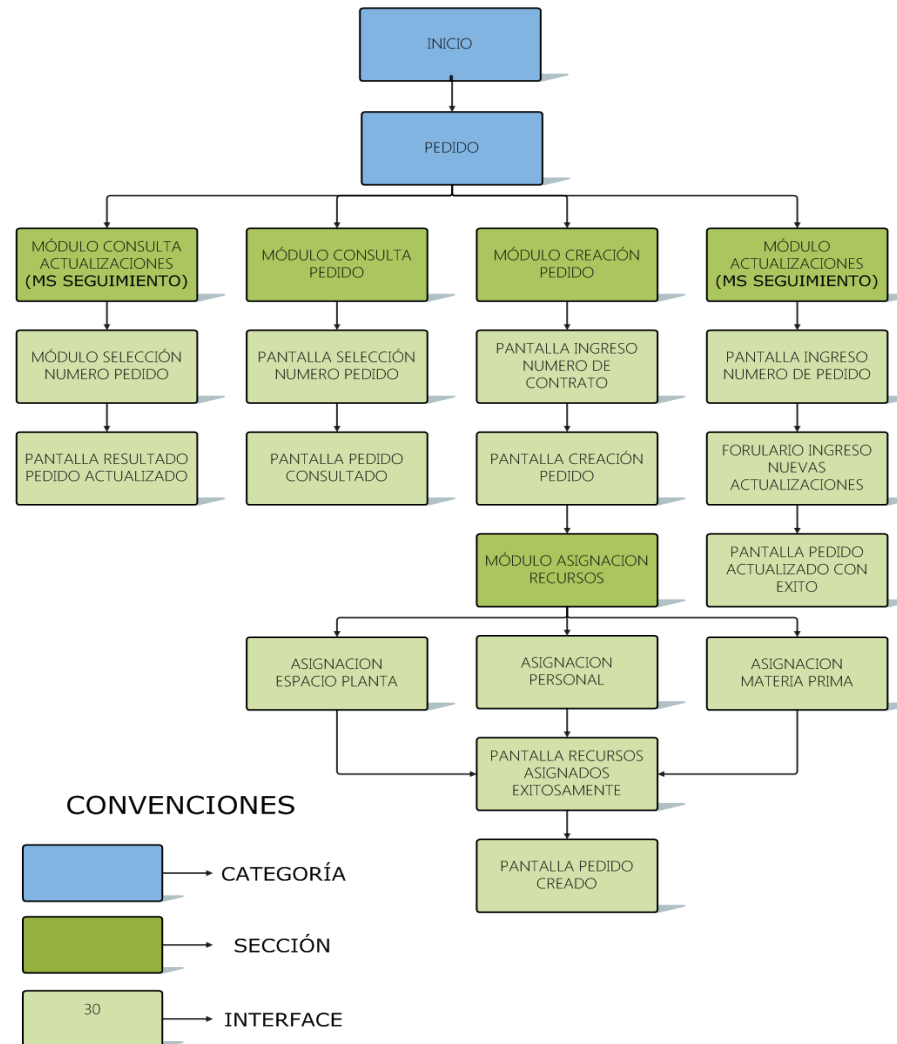
	ManejadorVerificacionEspacioPlanta	Clase de Control. Encargada de desplegar la interfaceVerificaciónEspacioPlanta, una vez validado el tipo de verificación elegida por el gerente.
Materia Prima	InterfaceAsignacionMateriaPrima	Clase de Borde. Se despliega el formato de asignación de Materia Prima luego haber validado el número del pedido en la interfaceCreacionPedido, se habilitan los campos para que el gerente pueda generar una nueva asignación de Materia Prima.
	ManejadorAsignarMateriaPrima	Clase de Control. Encargada de desplegar la interfaceAsignaciónMateriaPrima, una vez validada el tipo de asignación elegida por el gerente.
	InterfaceVerificacionMateriaPrima	Clase de Borde. Se despliega la interfaceVerificacionMateriaPrima con toda la información del personal.
	ManejadorVerificacionMateriaPrima	Clase de Control. Encargada de desplegar la interfaceVerificaciónMateriaPrima, una vez validado el tipo de verificación elegida por el gerente.
	PantallaRecepcionMateriaPrima	Clase de Borde. Se despliega el formato de recepción de materia prima, donde se ingresa la información de la materia prima que se quiere ingresar
	ManejadorRecepcionMateriaPrima	Clase de Control. Encargada de habilitar el formato de recepción de materia prima
	PantallaEntregaMateriaPrima	Clase de Borde. Se despliega un módulo en el cual será consultada la asignación de materia prima realizada según el número de pedido, el cual deberá ser ingresado
	ManejadorEntregaMateriaPrima	Clase de Control. Encargada de gestionar el estado la la asignación de materia prima una vez inicie el evento de entrega de la asignación, así mismo deberá validar la entrega realizada.

## H. Diagramas de navegación de ventanas.

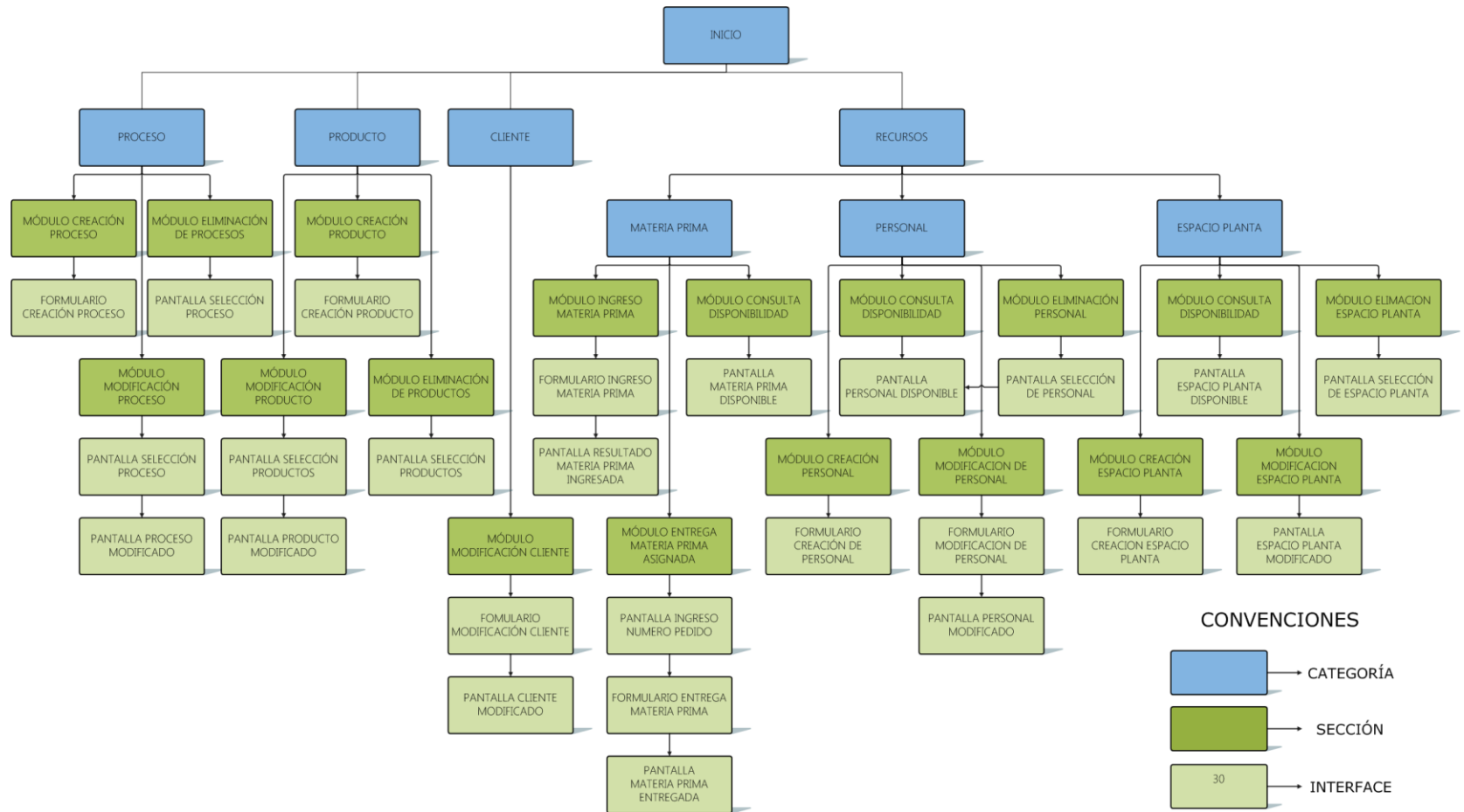
### Módulo negocios.



## Módulo pedido.



## Módulo recursos.



I. Casos de uso de condiciones de frontera.

Casos de Uso		
<b>Caso de Uso</b>	Guardar Componentes de creación	
<b>Actor(es)</b>	<i>controlCotización, Cotización</i>	
<b>Propósito</b>	<i>Salvar información de la cotización creada para su posterior uso por el componente contrato.</i>	
<b>Descripción</b>	<i>Cuando el proceso de creación está por finalizar controlContrato toma la información ingresada al módulo y la guarda en una base de datos relacional que es la que garantizará la persistencia de esta información para futuros accesos</i>	
<b>Precondición(es)</b>		
<b>Flujo Principal</b>	<b>Acciones de Actor(es)</b>	<b>Respuestas del Sistema</b>
	<ol style="list-style-type: none"> <li>1. controlCotización valida la información ingresada.</li> <li>2. Inicia la comunicación con el subsistema de almacenamiento y comunicación con la base de datos y guarda la cotización</li> <li>4. controlCotización recibe la confirmación de operación de BD exitosa y despliega el contrato creado en pantalla.</li> </ol>	<ol style="list-style-type: none"> <li>3. La base de datos confirma un proceso de almacenamiento exitoso</li> </ol>
<b>Sub-flujos</b>	3. La comunicación con la base de datos falla. Vuelve al paso 1.	
<b>Poscondición(es)</b>	Cotización guardada y persistente.	

<b>Caso de Uso</b>	Guardar contrato	
<b>Actor(es)</b>	<i>controlContrato, contrato</i>	
<b>Propósito</b>	<i>Salvar información del contrato creado para su posterior uso por el componente pedido y por el componente entrega pedido.</i>	
<b>Descripción</b>	<i>Cuando el caso de uso creación de contrato está por finalizar controlContrato toma la información ingresada al módulo y la guarda en una base de datos relacional que es la que garantizará la persistencia de esta información para futuros accesos</i>	
<b>Precondición(es)</b>		
<b>Flujo Principal</b>	<b>Acciones de Actor(es)</b>	<b>Respuestas del Sistema</b>
	<ol style="list-style-type: none"> <li>1. controlContrato valida la información ingresada.</li> <li>2. Inicia la comunicación con el subsistema de almacenamiento y comunicación con la base de datos y guarda el contrato</li> <li>4. controlContrato recibe la confirmación de operación de BD exitosa y despliega el contrato creado en pantalla.</li> </ol>	<ol style="list-style-type: none"> <li>3. La base de datos confirma un proceso de almacenamiento exitoso</li> </ol>
<b>Sub-flujos</b>	3. La comunicación con la base de datos falla. Vuelve al paso 1.	
<b>Poscondición(es)</b>	Contrato guardado y persistente.	
<b>Caso de Uso</b>	Guardar pedido	
<b>Actor(es)</b>	<i>controlPedido, Pedido</i>	
<b>Propósito</b>	<i>Salvar información creada para su posterior uso en consultas</i>	

	<i>y por el componente de entrega de pedido.</i>	
<b>Descripción</b>	<i>Cuando el proceso de creación de pedido está por finalizar, controlPedido toma la información importada de contrato y la obtenida al solicitar cada una de más asignaciones y la guarda en una base de datos relacional que es la que garantizará la persistencia de esta información para futuros accesos</i>	
<b>Precondición(es)</b>		
<b>Flujo Principal</b>	<b>Acciones de Actor(es)</b> 1. controlPedido valida la información importada desde contrato, como también las asignaciones realizadas por el gerente.  2. controlPedido Inicia la comunicación con el subsistema de almacenamiento y comunicación con la base de datos y guarda el pedido.  4. controlPedido recibe la confirmación de operación de BD exitosa.	<b>Respuestas del Sistema</b>  3. La base de datos confirma un proceso de almacenamiento exitoso
<b>Sub-flujos</b>	3. La comunicación con la base de datos falla. Vuelve al paso 1.	
<b>Poscondición(es)</b>	Pedido guardado y persistente.	
<b>Caso de Uso</b>	guardarMateriaPrima	
<b>Actor(es)</b>	controlIngreso, materiaPrima	

<b>Propósito</b>	<i>Hacer persistente cada objeto que confirme el ingreso de la materia prima pues esta puede ser consultada, asignada o entregada</i>	
<b>Descripción</b>	<i>Inicia cuando se está haciendo un nuevo ingreso de materia prima a través del módulo respectivo, allí se ingresa la información pertinente y se almacena en el subsistema de base de datos.</i>	
<b>Precondición(es)</b>		
<b>Flujo Principal</b>	<b>Acciones de Actor(es)</b>	<b>Respuestas del Sistema</b>
	1. controlIngreso inicia la comunicación con el subsistema de base de datos para guardar campo por campo cada elemento de materia prima que vaya a ser ingresado.	2. El subsistema de base de datos almacena la información y envía una respuesta de operación exitosa
<b>Sub-flujos</b>	Falla 2. Vuelve al paso 1	
<b>Poscondición(es)</b>	Materia prima ingresada y persistente	
<b>Caso de Uso</b>	crearPersonal	
<b>Actor(es)</b>	<i>Gerente, Personal, controlCreacionPersonal,</i>	
<b>Propósito</b>	<i>Crear un registro de todo el personal presente en la organización</i>	
<b>Descripción</b>	<i>El gerente desea agregar nuevo personal al sistema para que su información pueda ser utilizada en la aplicación.</i>	
<b>Precondición(es)</b>		
<b>Flujo Principal</b>	<b>Acciones de Actor(es)</b>	<b>Respuestas del Sistema</b>
	1. El gerente ingresa en el módulo de creación de personal la información del empleado y envía dicha información.	2. El sistema valida la información y solicita confirmación

	<p>3. El gerente confirma.</p> <p>4. controlCreaciónPerson al inicia la comunicación con el subsistema encargado de almacenar la información de la persona y hace la inserción.</p> <p>6. El gerente cierra el modulo</p>	<p>5. El subsistema de base de datos regresa una ventana advirtiendo una operación exitosa.</p>
<b>Sub-flujos</b>	<p>Paso: 2, falla la validación vuelve a 1.  Paso: 5, La operación con la base de datos ha fallado, regresa al paso 4.</p>	
<b>Poscondición(es)</b>	<p>Personal Creado y persistente</p>	
<b>Caso de Uso</b>	<p>crearEspacioPlanta</p>	
<b>Actor(es)</b>	<p><i>Gerente, espacioPlanta, controlCreacionEspacioPlanta,</i></p>	
<b>Propósito</b>	<p><i>Crear un registro de todo el espacioPlanta en la organización</i></p>	
<b>Descripción</b>	<p><i>El gerente desea agregar nuevo espacioPlanta al sistema para que su información pueda ser utilizada en la aplicación.</i></p>	
<b>Precondición(es)</b>		
<b>Flujo Principal</b>	<b>Acciones de Actor(es)</b>	<b>Respuestas del Sistema</b>
	<p>1. El gerente ingresa en el módulo de creación de espacioPlanta la información de la zona y envía dicha información.</p> <p>3. El gerente confirma.</p>	<p>2. El sistema válida la información y solicita confirmación.</p> <p>5. El subsistema de base de datos regresa una ventana advirtiendo una operación exitosa.</p>

	<p>4. controlCreaciónEspacioPlanta inicia la comunicación con el subsistema encargado de almacenar la información del espacioPlanta y hace la inserción.</p> <p>6. El gerente cierra el modulo</p>	
<b>Sub-flujos</b>	<p>Paso: 2, falla la validación vuelve a 1.  Paso: 5, La operación con la base de datos ha fallado, regresa al paso 4.</p>	
<b>Poscondición(es)</b>	EspacioPlanta creado y persistente	
<b>Caso de Uso</b>	crearMateriaPrima	
<b>Actor(es)</b>	<i>Almacenista, materiaPrima, controlCreacionMateriaPrima,</i>	
<b>Propósito</b>	<i>Crear un registro de toda la materia prima en la organización</i>	
<b>Descripción</b>	<i>El gerente desea agregar nueva materiaPrima al sistema para que su información pueda ser utilizada en la aplicación.</i>	
<b>Precondición(es)</b>		
<b>Flujo Principal</b>	<b>Acciones de Actor(es)</b>	<b>Respuestas del Sistema</b>
	<p>1. El Almacenista ingresa en el módulo de creación de materiaPrima la información de cada material y envía dicha información.</p>	<p>2. El sistema valida la información</p> <p>3. El sistema solicita confirmación para la creación del nuevo registro de</p>

	<p>4. El gerente confirma.</p> <p>7. El gerente cierra el modulo</p>	<p>materiaPrima</p> <p>5. controlCreaciónMateria Prima inicia la comunicación con el subsistema en cargado de almacenar la información de la materiaPrima y hace la inserción.</p> <p>6. El subsistema de base de datos regresa una ventana advirtiendo una operación exitosa.</p>
<b>Sub-flujos</b>	<p>Paso: 2, falla la validación vuelve a 1.  Paso: 3, La operación de inserción con la base de datos ha fallado, retorna a 5</p>	
<b>Poscondición(es)</b>	<p>Materia Prima creada y persistente</p>	
<b>Caso de Uso</b>	<p>Desplegar ventana de error</p>	
<b>Actor(es)</b>	<p><i>Usuarios del sistema que interactúan con formularios.</i></p>	
<b>Propósito</b>	<p><i>Desplegar una ventana en la que se indique que la información suministrada en ciertos campos es inválida.</i></p>	
<b>Descripción</b>	<p><i>En el proceso de validación de los formularios presentes a la hora de crear una cotización, un contrato, un pedido; al consultar el estado de un pedido o entregar un pedido terminado, pueden existir campos que se han diligenciado con datos inválidos o que no se han ingresado la información solicitada, si esto sucede el sistema muestra en pantalla una ventana informando del error.</i></p>	
<b>Flujo Principal</b>	<b>Acciones de Actor(es)</b>	<b>Respuestas del Sistema</b>
	<p>1. El usuario ingresa la</p>	<p>2. El sistema valida la</p>

	información en el formulario y la envía para su validación.	información y encuentra información errónea o sin ingresar. 3. El sistema despliega una ventana indicando el error en los campos solicitando proporcionar la información correctamente.
	4. El usuario acepta el mensaje	5. El sistema despliega el formulario nuevamente para el correcto ingreso de la información.
<b>Sub-flujos</b>		
<b>Poscondición(es)</b>		
<b>Caso de Uso</b>	Crear Proceso	
<b>Actor(es)</b>	<i>Gerente, ControlCreacionProceso, pedido</i>	
<b>Propósito</b>	<i>Crear todo el listado de procesos para relacionarlo con cada tipo de producto</i>	
<b>Descripción</b>	<i>El gerente hará uso de este caso de uso para ingresar los diferentes procesos y sus características para hacerle seguimiento a los mismos durante la fabricación.</i>	
<b>Precondición(es)</b>	Debe estar creado un tipo de producto para realizar la asociación con su respectivo listado de procesos	
<b>Flujo Principal</b>	<b>Acciones de Actor(es)</b>	<b>Respuestas del Sistema</b>
	1. El Gerente ingresa en el módulo de creación Proceso, e ingresa la información de cada operación y la envía.	2. El sistema valida la información
	4. El gerente confirma.	3. El sistema solicita confirmación para la creación del nuevo proceso
		5. ControlCreaciónProceso

		<p>inicia la comunicación con el subsistema en cargado de almacenar la información del proceso y hace la inserción.</p> <p>6. El subsistema de base de datos regresa una ventana advirtiendo una operación exitosa.</p> <p>7. El gerente cierra el modulo.</p>
<b>Sub-flujos</b>	<p>Paso: 2, falla la validación vuelve a 1.  Paso: 3, La operación de inserción con la base de datos ha fallado, retorna a 5</p>	
<b>Poscondición(es)</b>	Proceso creado y asociado a un tipo de producto	
<b>Caso de Uso</b>	CrearProducto	
<b>Actor(es)</b>	<i>Gerente, ControlCreacionProducto, pedido</i>	
<b>Propósito</b>	<i>Crear todo el listado de productos para relacionarlo a cada pedido en firme y asignarle una lista de recursos</i>	
<b>Descripción</b>	<i>El gerente hará uso de este caso de uso para ingresar los diferentes tipos de productos y sus características para hacerle seguimiento a los mismos durante la fabricación.</i>	
<b>Precondición(es)</b>		
<b>Flujo Principal</b>	<b>Acciones de Actor(es)</b>	<b>Respuestas del Sistema</b>
	<p>1. El Gerente ingresa en el módulo de creación Producto, e ingresa la información de cada operación y la envía.</p> <p>4. El gerente confirma.</p>	<p>2. El sistema valida los campos</p> <p>3. El sistema solicita confirmación para la creación del nuevo producto</p>

	7. El gerente cierra el modulo.	5. ControlCreaciónProducto inicia la comunicación con el subsistema en cargado de almacenar la información del proceso y hace la inserción.  6. El subsistema de base de datos regresa una ventana advirtiendo una operación exitosa.
<b>Sub-flujos</b>	Paso: 2, falla la validación vuelve a 1. Paso: 3, La operación de inserción con la base de datos ha fallado, retorna a 5	
<b>Poscondición(es)</b>	Proceso creado y asociado a un tipo de producto	

## J. Tarjetas CRC

### Objetos Borde

Tarjetas CRC	
<b>Objeto:</b>	Pantalla
<b>Descripción:</b>	Super clase principal de los objetos de tipo borde; hereda para todas ellas el método desplegar, necesario para que cada VInterface de despliegue para la interacción con el usuario.
<b>Módulo:</b>	
<b>Estereotipo:</b>	Borde
<b>Propiedades:</b>	Abstracta
<b>Superclases:</b>	
<b>Subclases:</b>	PantallaMostrar, VInterfaceGerente, VInterfaceSuper, VInterfaceAlmacenista, VInterfaceCreacion, VInterfaceEliminacion, VInterfaceModificacion, VPedido, VVInterfaceAdministracion
<b>Atributos:</b>	
<b>1. Desplegar Pantalla</b>	
Desplegar() devuelve void	

<b>Tarjetas CRC</b>	
<b>Objeto:</b>	pantallaMostrar
<b>Descripción:</b>	Encargada en heredar el método mostrar, cada vez que se requiera mostrar la información de una consulta en un módulo o pantalla durante la ejecución de la aplicación
<b>Módulo:</b>	
<b>Estereotipo:</b>	Borde
<b>Propiedades:</b>	Abstracta
<b>Superclases:</b>	pantalla
<b>Subclases:</b>	pantallaEnvio, pantallaCerrar, pantallaConsulta, VPantallaMostrarContrato, VPantallaMostrarCotizacion
<b>Atributos:</b>	
<b>2. Mostrar Información</b>	
haceMostrar(ArrayList datos, int opcion) devuelve void	
<b>Tarjetas CRC</b>	
<b>Objeto:</b>	pantallaEnvio
<b>Descripción:</b>	Es la clase encargada de agrupar los métodos que se encargan de tomar la información ingresada en cada uno de los formularios e VInterfaces y enviarlas a los diferentes objetos control para su respectivo manejo.
<b>Módulo:</b>	
<b>Estereotipo:</b>	Borde
<b>Propiedades:</b>	Abstracta
<b>Superclases:</b>	pantallaMostrar, Pantalla
<b>Subclases:</b>	pantallaValidacion, VInterfaceCreacionProceso, VVInterfaceCreacionProducto, VInterfaceAsignacionPersonal, VInterfaceAsignacionEspacioPlanta, VInterfaceAsignacionMateriaPrima, VInterfaceEliminacionProceso, VInterfaceEliminacionProducto, VInterfaceCreacionPersonal, VInterfaceCreacionEspacioPlanta, VInterfaceEliminarPersonal, VInterfaceEliminarEspacioPlanta, VInterfaceModificacionPersonal, VInterfaceModificacionEspacioPlanta,

	VInterfaceModificacionCliente, VInterfaceModificacionProducto, VInterfaceModificacionProceso,
<b>Atributos:</b>	
<b>3. Solicitar confirmación</b>	
enviarInformación(controlConfirmacion , Array_Info[ ] ) devuelve void	controlConfirmacion (3)
<b>Tarjetas CRC</b>	
<b>Objeto:</b>	pantallaCerrar
<b>Descripción:</b>	Encargada en heredar el método cerrar, cada vez que se requiera el cierre de un módulo o pantalla durante la ejecución de la aplicación.
<b>Módulo:</b>	
<b>Estereotipo:</b>	Borde
<b>Propiedades:</b>	Abstracta
<b>Superclases:</b>	pantallaMostrar, pantalla
<b>Subclases:</b>	pantallaMostrarMensaje, pantallaValidacionCodigo, VPantallaResultadoBusqueda, VInterfaceCreacionPedido, VPantallaMostrarPedido
<b>Atributos:</b>	
<b>4. Cerrar módulo</b>	
enviaCerrar(controlCierre) devuelve void	controlCierre(4)
<b>Tarjetas CRC</b>	
<b>Objeto:</b>	pantallaValidacionCodigo
<b>Descripción:</b>	Agrupada todas las pantallas que deben solicitar la validación de un código a un controlador
<b>Módulo:</b>	
<b>Estereotipo:</b>	Borde
<b>Propiedades:</b>	Abstracta
<b>Superclases:</b>	pantallaCerrar
<b>Subclases:</b>	VVentanaValidacionContrato, VVentanaValidacionCotizacion, VInterfaceCreacionContrato, VInterfaceConsultaCotizacion, VInterfaceCreacionCotizacion, VInterfaceEntregaMateriaPrima, VInterfaceActualizacionPedido, VInterfaceConsultaContrato,

	VPantallaValidacionCodigoPedido
<b>Atributos:</b>	
<b>5. Validar código</b>	
validarCodigo(String, controlValidacion) devuelve void	controlValidacion (5)
<b>Tarjetas CRC</b>	
<b>Objeto:</b>	pantallaValidacion
<b>Descripción:</b>	Método invocado para hacer validación de los campos ingresados en un formulario pantalla o VInterface
<b>Módulo:</b>	
<b>Estereotipo:</b>	borde
<b>Propiedades:</b>	Abstracta
<b>Superclases:</b>	pantallaEnvio, pantalla mostrar, pantalla
<b>Subclases:</b>	VInterfaceCreacionContrato VInterfaceEntregaMateriaPrima VInterfaceIngresoMatPrima VformularioIngresoActualizacion VInterfaceCreacionCotizacion VInterfaceConsultaContrato VInterfaceConsultaCotizacion
<b>Atributos:</b>	
<b>9. Validar campos</b>	
validarCampos() devuelve boolean	
<b>Tarjetas CRC</b>	
<b>Objeto:</b>	pantallaConsulta
<b>Descripción:</b>	Esta clase tiene los métodos necesarios para hacer llamados de consulta a diferentes entidades para hacer validaciones de información
<b>Módulo:</b>	
<b>Estereotipo:</b>	borde
<b>Propiedades:</b>	Abstracta
<b>Superclases:</b>	pantallaMostrar, pantalla
<b>Subclases:</b>	VInterfaceConsultaActualizacionPedido VInterfaceVerificacionRecursosDisponibles (sólo implementaría método de Pantalla), VInterfaceConsultaPedido

<b>Atributos:</b>	
<b>7. Consultar</b>	
solicitarConsulta(controlConsulta, String) devuelve void	controlConsulta (7)
<b>Tarjetas CRC</b>	
<b>Objeto:</b>	VInterfaceCreacionContrato
<b>Descripción:</b>	Encargada de mostrar la información procedente de cotización y recibir los datos restantes para la creación de un contrato
<b>Módulo:</b>	CreacionContrato
<b>Estereotipo:</b>	VInterface
<b>Propiedades:</b>	Concreta
<b>Superclases:</b>	pantallaValidacion, pantallaValidacionCodigo
<b>Subclases:</b>	-
<b>Atributos:</b>	Datos: ArrayList, objCrea: ControlCreacionContrato, objConCot: controlConsultaCotización
<b>1. Desplegar pantalla</b>	
Desplegar() Devuelve void	
<b>5. Validar Código</b>	
validarCodigo()Devuelve void	controlConsultaCotizacion
<b>9. Validar Campos</b>	
validarCampos()Devuelve boolean	
<b>2. Mostrar Información</b>	
haceMostrar(ArrayList datos, int opcion) Devuelve void	
<b>3. Solicitar Confirmacion</b>	
enviarInformación()Devuelve void	controlCreacionContrato
<b>4. Cerrar Módulo</b>	
Interface: enviaCerrar()Devuelve void	
<b>Tarjetas CRC</b>	
<b>Objeto:</b>	VInterfaceConsultaActualizacionPedido
<b>Descripción:</b>	Pantalla que permite ver y seleccionar el pedido al que se le consultarán las actualizaciones
<b>Módulo:</b>	
<b>Estereotipo:</b>	Borde
<b>Propiedades:</b>	Concreta
<b>Superclases:</b>	PantallaConsulta
<b>Subclases:</b>	-
<b>Atributos:</b>	objCtrlConsulta : ControlConsultaActualizacionPedido
<b>1. Desplegar Pantalla</b>	

Desplegar() devuelve void		
<b>2. Mostrar Información</b>		
haceMostrar(ArrayList datos, int opcion) devuelve void		
<b>7. Consultar</b>		
solicitarConsulta()Devuelve void		controlConsultaActualizacionPedido
Limpiar() devuelve void		
<b>Tarjetas CRC</b>		
<b>Objeto:</b>	VPantallaResultadoBusqueda	
<b>Descripción:</b>	Encargado de mostrar el resultado de las consultas de actualización para el pedido seleccionado	
<b>Módulo:</b>	-	
<b>Estereotipo:</b>	Borde	
<b>Propiedades:</b>	concreta	
<b>Superclases:</b>	pantallaCerrar	
<b>Subclases:</b>	-	
<b>Atributos:</b>		
<b>1. Desplegar Pantalla</b>		
Desplegar() devuelve void		
<b>2. Mostrar Información</b>		
haceMostrar(ArrayList datos, int opcion) devuelve void		
<b>4. Cerrar Módulo</b>		
enviaCerrar() devuelve void		controlConsultaActualizacionPedido
cerrarInterface() devuelve void		
calculoProgreso(ArrayList Busqueda) devuelve void		
getObjPantallaBusqueda() devuelve VPantallaResultadoBusqueda		
<b>Tarjetas CRC</b>		
<b>Objeto:</b>	VInterfaceEntregaMateriaPrima	
<b>Descripción:</b>	Encargada de recibir el número de pedido, mostrar su asignación de materia prima y realizar su entrega	
<b>Módulo:</b>		
<b>Estereotipo:</b>	Borde	
<b>Propiedades:</b>	Concreta	
<b>Superclases:</b>	pantallaValidaciónCodigo, pantallaValidacion	
<b>Subclases:</b>	-	
<b>Atributos:</b>	-	

<b>1. Desplegar Pantalla</b>		
Desplegar() devuelve void		
<b>5. Validar Código</b>		
validarCodigo() devuelve boolean		controlConsultaPedido
<b>2. Mostrar Información</b>		
haceMostrar(ArrayList datos, int opcion) devuelve void		
<b>9. Validar Campos</b>		
validarCampos() devuelve boolean		
<b>3. Solicitar Confirmación</b>		
enviarInformación() devuelve void		controlEntregaMaterial
setCodPedido(int codPedido) devuelve void		
getObjIntEntr() devuelve VInterfaceEntregaMateriaPRima		
<b>Tarjetas CRC</b>		
<b>Objeto:</b>	VPantallaValidacionCodigoPedido	
<b>Descripción:</b>	Encargada de recibir el número de pedido, para dar paso a su entrega	
<b>Módulo:</b>	-	
<b>Estereotipo:</b>	Borde	
<b>Propiedades:</b>	Concreta	
<b>Superclases:</b>	pantallaValidaciónCodigo	
<b>Subclases:</b>	-	
<b>Atributos:</b>	-	
<b>1. Desplegar Pantalla</b>		
Desplegar() devuelve void		
<b>5. Validar Código</b>		
validarCodigo() devuelve void		controlConsultaPedido
<b>2. Mostrar Información</b>		
haceMostrar(ArrayList datos, int opcion) devuelve void		
<b>4. Cerrar Módulo</b>		
enviaCerrar() devuelve void		
getObjIntPantVal() devuelve VPantallaValidacionPedido		
<b>Tarjetas CRC</b>		
<b>Objeto:</b>	VInterfaceIngresoMatPrima	
<b>Descripción:</b>	Permite recibir los datos necesarios para ingresar materia prima en el sistema	
<b>Módulo:</b>	-	
<b>Estereotipo:</b>	Borde	

<b>Propiedades:</b>	Concreta
<b>Superclases:</b>	pantallaValidacion
<b>Subclases:</b>	-
<b>Atributos:</b>	objContP : ControladorPrincipal
<b>1. Desplegar Pantalla</b>	
Desplegar() devuelve void	
<b>9. Validar Campos</b>	
validarCampos() devuelve boolean	
<b>3. Solicitar Confirmación</b>	
enviarInformacion () devuelve void	controlIngresoMateriaPrima
getObjIntMatPrima() devuelve VInterfaceIngresoMatPrima	
<b>2. Mostrar Información</b>	
Interface: haceMostrar(ArrayList datos, int opcion) devuelve void	
<b>Tarjetas CRC</b>	
<b>Objeto:</b>	VInterfaceAsignacionPersonal
<b>Descripción:</b>	Muestra el personal disponible y permite seleccionarlo para su posterior asignación
<b>Módulo:</b>	-
<b>Estereotipo:</b>	Borde
<b>Propiedades:</b>	Concreta
<b>Superclases:</b>	pantallaEnvio
<b>Subclases:</b>	-
<b>Atributos:</b>	-
<b>1. Desplegar Pantalla</b>	
Desplegar() devuelve void	
<b>2. Mostrar Información</b>	
haceMostrar(ArrayList datos, int opcion) devuelve void	
<b>3. Solicitar Confirmación</b>	
enviarInformación() devuelve void	controlAsignacionPersonal
getObjIntAsPer() devuelve VInterfaceAsignacionPersonal	
setCodigoPedido(int codPedido) devuelve void	
<b>Tarjetas CRC</b>	
<b>Objeto:</b>	VInterfaceAsignacionEspacioPlanta
<b>Descripción:</b>	Muestra el espacio en planta disponible y permite seleccionarlo para su posterior asignación
<b>Módulo:</b>	-
<b>Estereotipo:</b>	Borde

<b>Propiedades:</b>	Concreta
<b>Superclases:</b>	pantallaEnvio
<b>Subclases:</b>	-
<b>Atributos:</b>	-
<b>1. Desplegar Pantalla</b>	
Desplegar() devuelve void	
<b>2. Mostrar Información</b>	
haceMostrar(ArrayList datos, int opcion) devuelve void	
<b>3. Solicitar Confirmación</b>	
enviarInformación() devuelve void	controlAsignacionEspacioPlanta
getObjIntEspPI() devuelve VInterfaceAsignacionEspacioPlanta	
setCodPedido(int codPedido) devuelve void	
<b>Tarjetas CRC</b>	
<b>Objeto:</b>	VInterfaceAsignacionMateriaPrima
<b>Descripción:</b>	Muestra la materia prima disponible y permite seleccionarla para su posterior asignación
<b>Módulo:</b>	-
<b>Estereotipo:</b>	Borde
<b>Propiedades:</b>	Concreta
<b>Superclases:</b>	pantallaEnvio
<b>Subclases:</b>	-
<b>Atributos:</b>	-
<b>1. Desplegar Pantalla</b>	
Desplegar() devuelve void	
<b>2. Mostrar Información</b>	
haceMostrar(ArrayList datos, int opcion) devuelve void	
<b>3. Solicitar Confirmación</b>	
enviarInformación() devuelve void	controlAsignacionMateriaPrima
getCantidadAsignada() devuelve ArrayList	
getObjAsMat() devuelve VInterfaceAsignacionMateriaPrima	
setCodigoPedido(int codPedido) devuelve void	
<b>Tarjetas CRC</b>	
<b>Objeto:</b>	VPedido
<b>Descripción:</b>	Permite seleccionar el módulo al cual se desea ingresar. Solo implementaría método de pantalla

<b>Módulo:</b>	-
<b>Estereotipo:</b>	Borde
<b>Propiedades:</b>	Concreta
<b>Superclases:</b>	Pantalla
<b>Subclases:</b>	-
<b>Atributos:</b>	objContP : ControladorPrincipal
<b>1. Desplegar Pantalla</b>	
Desplegar()	devuelve void
<b>Tarjetas CRC</b>	
<b>Objeto:</b>	VInterfaceCreacion
<b>Descripción:</b>	Permite seleccionar el módulo al cual se desea ingresar. Solo impementaría método de pantalla
<b>Módulo:</b>	
<b>Estereotipo:</b>	Borde
<b>Propiedades:</b>	Concreta
<b>Superclases:</b>	Pantalla
<b>Subclases:</b>	
<b>Atributos:</b>	
<b>1. Desplegar Pantalla</b>	
Desplegar()	
btnCreaProcesoActionPerformed(java.a wt.event.ActionEvent evt)	controladorPrincipal Devuelve void
btnCreaEspacioActionPerformed (java.awt.event)	controladorPrincipal Devuelve void
btnCreaPersonaActionPerformed (java.awt.event)	controladorPrincipal Devuelve void
btnCreaPersonaActionPerformed (java.awt.event)	controladorPrincipal Devuelve void
getObjIntCreacion()	devuelve VInterfaceCreacion
<b>Tarjetas CRC</b>	
<b>Objeto:</b>	VInterfaceEliminacion
<b>Descripción:</b>	Permite seleccionar el módulo al cual se desea ingresar. Solo impementaría método de pantalla
<b>Módulo:</b>	-
<b>Estereotipo:</b>	Borde
<b>Propiedades:</b>	Concreta
<b>Superclases:</b>	Pantalla
<b>Subclases:</b>	-
<b>Atributos:</b>	-

<b>1. Desplegar Pantalla</b>		
Desplegar() Devuelve void		
btnEliminaProcesoActionPerformed(java .event) devuelve void		controladorPrincipal
btnEliminaPersonaActionPerformed(java .event) devuelve void		controladorPrincipal
btnEliminaespacioActionPerformed(java .event) devuelve void		controladorPrincipal
btnRegresaActionPerformed(java.event) devuelve void		controladorPrincipal
getObjIntEliminacion() devuelve VInterfaceEliminacion		
<b>Tarjetas CRC</b>		
<b>Objeto:</b>	VInterfaceModificacion	
<b>Descripción:</b>	Permite seleccionar el módulo al cual se desea ingresar. Solo impementaría método de pantalla	
<b>Módulo:</b>	-	
<b>Estereotipo:</b>	Borde	
<b>Propiedades:</b>	Concreta	
<b>Superclases:</b>	Pantalla	
<b>Subclases:</b>	-	
<b>Atributos:</b>	-	
<b>1. Desplegar Pantalla</b>		
Desplegar() Devuelve void		
btnRegresaActionPerformed (java.event) Devuelve void		controladorPrincipal
btnModificaEspacioActionPerformed (java.event) Devuelve void		controladorPrincipal
btnModificaPersonalActionPerformed (java.event) Devuelve void		controladorPrincipal
btnModificaClienteActionPerformed (java.event) Devuelve void		controladorPrincipal
getObjIntModificacion() devuelve VInterfaceModificacion		
<b>Tarjetas CRC</b>		
<b>Objeto:</b>	VVInterfaceAdministracion	
<b>Descripción:</b>	Permite seleccionar el módulo al cual se desea ingresar. Solo impementaría método de pantalla	
<b>Módulo:</b>	-	
<b>Estereotipo:</b>	Borde	
<b>Propiedades:</b>	Concreta	
<b>Superclases:</b>	Pantalla	

<b>Subclases:</b>	-
<b>Atributos:</b>	-
<b>1. Desplegar Pantalla</b>	
Desplegar() Devuelve void	
btnRegresarActionPerformed(java.event ) Devuelve void	controladorPrincipal
btnEliminarActionPerformed(java.event) Devuelve void	controladorPrincipal
btnCrearActionPerformed(java.event) Devuelve void	controladorPrincipal
btnModificarActionPerformed(java.event ) Devuelve void	controladorPrincipal
objIntAdmin() devuelve VInterfaceAdministracion	
<b>Tarjetas CRC</b>	
<b>Objeto:</b>	VInterfaceConsultaCotizacion
<b>Descripción:</b>	Permite ingresar el código de la cotización a consultar así como ver el resultado de la consulta
<b>Módulo:</b>	-
<b>Estereotipo:</b>	Borde
<b>Propiedades:</b>	Concreta
<b>Superclases:</b>	pantallaValidacion, pantallaValidacionCodigo
<b>Subclases:</b>	-
<b>Atributos:</b>	Datos : ArrayList
<b>1. Desplegar Pantalla</b>	
Desplegar() Devuelve void	
<b>5. Validar Código</b>	
validarCodigo() Devuelve void	controlConsultaCotizacion
<b>2. Mostrar Información</b>	
haceMostrar(ArrayList datos, int opcion) Devuelve void	
<b>4. Cerrar Módulo</b>	
enviaCerrar()Devuelve void	controlConsultaCotizacion
getObjIntConsCot() Devuelve VInterfaceConsultaCotizacion	
<b>Tarjetas CRC</b>	
<b>Objeto:</b>	VInterfaceConsultaContrato
<b>Descripción:</b>	Permite ingresar el código del contrato a consultar así como ver el resultado de la consulta
<b>Módulo:</b>	-
<b>Estereotipo:</b>	Borde
<b>Propiedades:</b>	Concreta

<b>Superclases:</b>	PantallaValidacion , pantallaValidacionCodigo	
<b>Subclases:</b>	-	
<b>Atributos:</b>	Datos : ArrayList	
<b>1. Desplegar Pantalla</b>		
	Desplegar() Devuelve void	
<b>5. Validar Código</b>		
	validarCodigo() Devuelve void	controlConsultaContrato
<b>2. Mostrar Información</b>		
	haceMostrar(ArrayList datos, int opcion) Devuelve void	
<b>4. Cerrar Módulo</b>		
	enviaCerrar() Devuelve void	controlConsultaContrato
	getObjIntConsCot() Devuelve VInterfaceConsultaContrato	
<b>Tarjetas CRC</b>		
<b>Objeto:</b>	VInterfaceConsultaPedido	
<b>Descripción:</b>	Permite ingresar el código del pedido a consultar así como ver el resultado de la consulta	
<b>Módulo:</b>	-	
<b>Estereotipo:</b>	Borde	
<b>Propiedades:</b>	Concreta	
<b>Superclases:</b>	pantallaCerrar, pantallaValidacionCodigo, pantallaConsulta	
<b>Subclases:</b>	-	
<b>Atributos:</b>	objContConPe : ControlConsultaPedido	
<b>1. Desplegar Pantalla</b>		
	Desplegar() devuelve void	
<b>5. Validar Código</b>		
	validarCodigo() devuelve void	controlConsultaPedido
<b>2. Mostrar Información</b>		
	haceMostrar(ArrayList datos, int opcion) devuelve void	
<b>4. Cerrar Módulo</b>		
	enviaCerrar() devuelve void	controlConsultaPedido
	getObjIntConsPedido() Devuelve VInterfaceConsultaPedido	
	solicitaConsulta() devuelve void	controlConsultaPedido
<b>Tarjetas CRC</b>		
<b>Objeto:</b>	VPantallaMostrarPedido	
<b>Descripción:</b>	Pantalla encargada de mostrar el resultado del pedido consultado y su respectiva asignación de materiales	
<b>Módulo:</b>	-	
<b>Estereotipo:</b>	Borde	

<b>Propiedades:</b>	Concreta
<b>Superclases:</b>	Pantalla, PantallaMostrar, PantallaCerrar
<b>Subclases:</b>	-
<b>Atributos:</b>	Datos : ArrayList, datosRecursos : ArrayList
<b>1. Desplegar</b>	
Desplegar() devuelve void	
<b>2. Mostrar Información</b>	
ControlConsultaPedido	
haceMostrar(ArrayList datos, int opcion) devuelve void	
setCodigo(int Codigo) devuelve void	
getObjPantMostP() Devuelve VPantallaMostrarPedido	
<b>4. Cerrar Módulo</b>	
enviaCerrar() devuelve void	controlConsultaEntregaPedido
<b>Tarjetas CRC</b>	
<b>Objeto:</b>	VInterfaceCreacionProceso
<b>Descripción:</b>	Recibe la información del nuevo proceso a crear
<b>Módulo:</b>	-
<b>Estereotipo:</b>	Borde
<b>Propiedades:</b>	Concreta
<b>Superclases:</b>	pantallaEnvio
<b>Subclases:</b>	-
<b>Atributos:</b>	-
<b>1. Desplegar Pantalla</b>	
Desplegar()	
<b>3. Solicitar confirmación</b>	
enviarInformación()	controlCreacionProceso
Limpiar() devuelve void	
cargarCombo() Devuelve void	ControlCreacionProceso
Cerrar()Devuelve void	
<b>2. Mostrar Informacion</b>	
haceMostrar(ArrayList datos, int opcion) devuelve void	
getObjIntCreaProceso() Devuelve VInterfaceCreacionProceso	
<b>Tarjetas CRC</b>	
<b>Objeto:</b>	VInterfaceEliminacionProceso
<b>Descripción:</b>	Permite seleccionar el proceso a eliminar
<b>Módulo:</b>	-
<b>Estereotipo:</b>	Borde
<b>Propiedades:</b>	Concreta
<b>Superclases:</b>	pantallaEnvio

<b>Subclases:</b>	-
<b>Atributos:</b>	DatosProceso : ArrayList, objCtrlEliminaProceso : ControlEliminacionProceso
<b>1. Desplegar Pantalla</b>	
Desplegar() devuelve void	
<b>2. Mostrar Información</b>	
haceMostrar(ArrayList datos, int opcion) devuelve void	
<b>3. Solicitar Confirmación</b>	
enviarInformación() devuelve void	controlEliminacionProceso
getObjIntEliminaProceso() devuelve VInterfaceEliminacionProceso	
<b>Tarjetas CRC</b>	
<b>Objeto:</b>	VInterfaceEliminacionProducto
<b>Descripción:</b>	Permite seleccionar el producto a eliminar
<b>Módulo:</b>	-
<b>Estereotipo:</b>	Borde
<b>Propiedades:</b>	Concreta
<b>Superclases:</b>	pantallaEnvio
<b>Subclases:</b>	-
<b>Atributos:</b>	DatosProducto : ArrayList, objCtrlEliminaProproducto : ControlEliminacionProducto
<b>1. Desplegar Pantalla</b>	
Desplegar() devuelve void	
<b>2. Mostrar Información</b>	
haceMostrar(ArrayList datos, int opcion) devuelve void	
<b>3. Solicitar Confirmación</b>	
enviarInformación() devuelve void	controlEliminacionProducto
getObjEliminaProducto() devuelve VInterfaceEliminacionProducto	
<b>Tarjetas CRC</b>	
<b>Objeto:</b>	VInterfaceVerificacionRecursosDisponibles
<b>Descripción:</b>	Permite generar la orden de verificación del recurso seleccionado
<b>Módulo:</b>	-
<b>Estereotipo:</b>	Borde
<b>Propiedades:</b>	Concreta
<b>Superclases:</b>	pantallaConsulta
<b>Subclases:</b>	-
<b>Atributos:</b>	objContVer : ControlVerificacion

<b>1. Desplegar Pantalla</b>	
Desplegar() devuelve void	
<b>7. Consultar</b>	
solicitarConsulta() devuelve void	controlVerificación
<b>2. Mostrar Información</b>	
haceMostrar(ArrayList datos, int opcion) devuelve void	
getObjIntVerRec() devuelve VInterfaceVerificacionRecursosDisponib les	
<b>Tarjetas CRC</b>	
<b>Objeto:</b>	VInterfaceActualizacionPedido
<b>Descripción:</b>	Recibe el código del pedido a actualizar
<b>Módulo:</b>	-
<b>Estereotipo:</b>	Borde
<b>Propiedades:</b>	Concreta
<b>Superclases:</b>	pantallaValidacionCodigo
<b>Subclases:</b>	-
<b>Atributos:</b>	-
<b>1. Desplegar Pantalla</b>	
Desplegar()	
<b>5. Validar código</b>	
validarCodigo()	controlConsultaPedido
getObjIntActualizaPedido() Devuelve VInterfaceActualizacionPedido	
<b>4. Cerrar Módulo</b>	
enviaCerrar() devuelve void	
getOpcion() devuelve int	
<b>Tarjetas CRC</b>	
<b>Objeto:</b>	VFormularioIngresoActualizacion
<b>Descripción:</b>	Recibe la actualización del pedido seleccionado
<b>Módulo:</b>	-
<b>Estereotipo:</b>	Borde
<b>Propiedades:</b>	Concreta
<b>Superclases:</b>	pantallaValidacion
<b>Subclases:</b>	-
<b>Atributos:</b>	CheckBoxes : ArrayList<JCheckBox>, Labels : ArrayList<JLabel>, Labels3 : ArrayList<JLabel> l = int, jcheck : JCheckBox[], objCtrlActualizaPedido : ControlActualizaciónPedido
<b>1. Desplegar Pantalla</b>	

Desplegar() devuelve void		
<b>9. Validar Campos</b>		
validarCampos() devuelve void		
<b>3. Solicitar Confirmación</b>		
enviarInformación() devuelve void		controlActualizacionPedido
<b>4. Cerrar Módulo</b>		
cerrarInterface() devuelve void		
getObjFormActualización() devuelve VFormularioIngresoActualización		
<b>Tarjetas CRC</b>		
<b>Objeto:</b>	VInterfaceCreacionCotizacion	
<b>Descripción:</b>	Recibe la información necesaria para la creación de una nueva cotización	
<b>Módulo:</b>	-	
<b>Estereotipo:</b>	Borde	
<b>Propiedades:</b>	Concreta	
<b>Superclases:</b>	pantallaValidacionCodigo, pantallaValidacion	
<b>Subclases:</b>	-	
<b>Atributos:</b>	Datos : ArrayList	
<b>1. Desplegar Pantalla</b>		
Desplegar() devuelve void		
<b>9. Validar Campos</b>		
validarCampos() devuelve boolean		
<b>3. Solicitar Confirmación</b>		
enviarInformación()		controlCreaciónCotización
<b>4. CerrarModulo</b>		
enviaCerrar() devuelve void		
setCodigoCod(int codigo) devuelve void		
getObjVIntCreCot() devuelve VInterfaceCreacionCotizacion		
<b>Tarjetas CRC</b>		
<b>Objeto:</b>	VVentanaValidacionCotizacion	
<b>Descripción:</b>	Recibe el código de la cotización a validar	
<b>Módulo:</b>	-	
<b>Estereotipo:</b>	Borde	
<b>Propiedades:</b>	Concreta	
<b>Superclases:</b>	pantallaValidacionCodigo	
<b>Subclases:</b>	-	
<b>Atributos:</b>	-	
<b>1. Desplegar Pantalla</b>		
Desplegar() devuelve void		
<b>5. Validar Código</b>		

validarCodigo() devuelve void	controlConsultaCotizacion
<b>2. Mostrar Información</b>	
haceMostrar(ArrayList datos, int opcion) devuelve void	
getObjVenValCon() devuelve VVentanaValidacionCotizacion	
<b>4. CerrarModulo</b>	
enviaCerrar() devuelve void	
<b>Tarjetas CRC</b>	
<b>Objeto:</b>	VVentanaValidacionContrato
<b>Descripción:</b>	Recibe el código de contrato a validar
<b>Módulo:</b>	-
<b>Estereotipo:</b>	Borde
<b>Propiedades:</b>	Concreta
<b>Superclases:</b>	pantallaValidacionCodigo
<b>Subclases:</b>	-
<b>Atributos:</b>	-
<b>1. Desplegar Pantalla</b>	
Desplegar() devuelve void	
<b>5. Validar Código</b>	
validarCodigo() devuelve void	controlConsultaContrato
<b>2. Mostrar Información</b>	
haceMostrar(ArrayList datos, int opcion) devuelve void	
getObjVentCont() devuelve VVentanaValidacionContrato	
<b>4. CerrarModulo</b>	
enviaCerrar() devuelve void	
<b>Tarjetas CRC</b>	
<b>Objeto:</b>	VInterfaceCreacionPedido
<b>Descripción:</b>	Recibe y muestra toda la información necesaria para la creación de un nuevo pedido
<b>Módulo:</b>	-
<b>Estereotipo:</b>	Borde
<b>Propiedades:</b>	Concreta
<b>Superclases:</b>	pantallaCerrar
<b>Subclases:</b>	
<b>Atributos:</b>	materiaPRima : ArrayList, objcontAsi : ControlAsignacion, objContConsCont : controlConsultaContrato, objIntAsMaterial : VInterfaceAsignacionMateriaPrima, personal : ArrayList, codMatPrima : ArrayList, codPersonal : ArrayList, espPlanta : ArrayList

<b>1. Desplegar Pantalla</b>		
Desplegar() devuelve void		
<b>2. Mostrar Información</b>		
haceMostrar(ArrayList datos, int opcion) devuelve void		
<b>3. Solicitar Confirmación</b>		
Interface: enviarInformacion() devuelve void		
<b>4. Cerrar Módulo</b>		
enviaCerrar() devuelve void		controlCreaciónPedido
getEspPlanta() devuelve ArrayList		
getMateriaPrima() devuelve ArrayList		
getPersonal() devuelve ArrayList		
<b>Tarjetas CRC</b>		
<b>Objeto:</b>	VInterfacelInicio	
<b>Descripción:</b>	Recibe los datos de inicio de sesión de un usuario	
<b>Módulo:</b>		
<b>Estereotipo:</b>	Borde	
<b>Propiedades:</b>	Concreta	
<b>Superclases:</b>	pantallaConsulta	
<b>Subclases:</b>		
<b>Atributos:</b>		
<b>1. Desplegar Pantalla</b>		
Desplegar() devuelve void		
<b>7. Consultar</b>		
solicitarConsulta() devuelve void		controladorPrincipal
<b>Tarjetas CRC</b>		
<b>Objeto:</b>	VInterfaceGerente	
<b>Descripción:</b>	Se encarga de dirigir al gerente al módulo que desee ir dependiendo la tarea que vaya a realizar, es el menú principal para este usuario	
<b>Módulo:</b>	-	
<b>Estereotipo:</b>	Borde	
<b>Propiedades:</b>	Concreta	
<b>Superclases:</b>	Pantalla	
<b>Subclases:</b>	-	
<b>Atributos:</b>	-	
<b>1. Desplegar Pantalla</b>		
Desplegar() devuelve void		

btnPedidoActionPerformed(java.Event) devuelve void		
btnAdministracionActionPerformed(java.Event) devuelve void		ControladorPrincipal
btnSeguimientoActionPerformed(java.Event) devuelve void		ControladorPrincipal
btnRecursosActionPerformed(java.Event) devuelve void		ControladorPrincipal
btnCerrarActionPerformed(java.Event) devuelve void		ControladorPrincipal
getObjIntGer() Devuelve VInterfaceGerente		
<b>Tarjetas CRC</b>		
<b>Objeto:</b>	VInterfaceAlmacenista	
<b>Descripción:</b>	Se encarga de dirigir al almacenista al módulo que desee ir dependiendo la tarea que vaya a realizar, es el menú principal para este usuario	
<b>Módulo:</b>	-	
<b>Estereotipo:</b>	Borde	
<b>Propiedades:</b>	Concreta	
<b>Superclases:</b>	Pantalla	
<b>Subclases:</b>	-	
<b>Atributos:</b>	-	
<b>1. Desplegar Pantalla</b>		
Desplegar()devuelve void		
btnIngresoActionPerformed(java.event)		ControladorPrincipal
btnSalirActionPerformed(java.event)		ControladorPrincipal
getObjIntAlm() Devuelve VInterfaceAlmacenista		
<b>Tarjetas CRC</b>		
<b>Objeto:</b>	VInterfaceSuper	
<b>Descripción:</b>	Se encarga de dirigir al Supervisor al módulo que desee ir dependiendo la tarea que vaya a realizar, es el menú principal para este usuario	
<b>Módulo:</b>	-	
<b>Estereotipo:</b>	Borde	
<b>Propiedades:</b>	Concreta	
<b>Superclases:</b>	Pantalla	
<b>Subclases:</b>	-	
<b>Atributos:</b>	-	
<b>1. Desplegar Pantalla</b>		
Desplegar()devuelve void		

btnSeguimientoActionPerformed(java.event)	ControladorPrincipal
btnCerrarActionPerformed(java.event)	ControladorPrincipal
getObjIntSuper()	Devuelve VInterfaceSuper
<b>Tarjetas CRC</b>	
<b>Objeto:</b>	VInterfaceCreacionPersonal
<b>Descripción:</b>	Recibe la información necesaria para añadir un nuevo empleado en el sistema
<b>Módulo:</b>	-
<b>Estereotipo:</b>	Borde
<b>Propiedades:</b>	Concreta
<b>Superclases:</b>	pantallaEnvio
<b>Subclases:</b>	-
<b>Atributos:</b>	-
<b>1. Desplegar Pantalla</b>	
Desplegar()	devuelve void
<b>3. Solicitar confirmación</b>	
enviarInformación()	devuelve void controlAñadirPersonal
Limpiar()	devuelve void
Cerrar()	devuelve void
getObjCreacionPersonal()	Devuelve VInterfaceCreacionPersonal
<b>Tarjetas CRC</b>	
<b>Objeto:</b>	VInterfaceCreacionEspacioPlanta
<b>Descripción:</b>	Recibe la información necesaria para añadir una nueva zona
<b>Módulo:</b>	-
<b>Estereotipo:</b>	Borde
<b>Propiedades:</b>	Concreta
<b>Superclases:</b>	pantallaEnvio
<b>Subclases:</b>	-
<b>Atributos:</b>	-
<b>1. Desplegar Pantalla</b>	
Desplegar()	devuelve void
<b>3. Solicitar Confirmación</b>	
enviarInformación()	devuelve void controlAñadirEspacioPlanta
Limpiar()	devuelve void
Cerrar()	devuelve void
<b>2. Mostrar Información</b>	

haceMostrar(ArrayList datos, int opcion) devuelve void	
getObjIntAñadirEspacio() Devuelve VInterfaceCreacionEspacioPlata	
<b>Tarjetas CRC</b>	
<b>Objeto:</b>	VInterfaceModificacionPersonal
<b>Descripción:</b>	Permite seleccionar un empleado para modificar su información
<b>Módulo:</b>	-
<b>Estereotipo:</b>	Borde
<b>Propiedades:</b>	Concreta
<b>Superclases:</b>	pantallaConsulta
<b>Subclases:</b>	-
<b>Atributos:</b>	-
<b>1. Desplegar Pantalla</b>	
Desplegar() devuelve void	
<b>3. Solicitar Confirmación</b>	
enviarInformacion() devuelve void	ControlModificacionPErsonal
<b>2. Mostrar Información</b>	
haceMostrar(ArrayList datos, int opcion) devuelve void	
Limpiar() devuelve void	
getObjIntModificaPersona() Devuelve VInterfaceModificacionPersonal	
<b>Tarjetas CRC</b>	
<b>Objeto:</b>	VInterfaceModificacionEspacioPlanta
<b>Descripción:</b>	Permite seleccionar una zona para realizar la modificación de información
<b>Módulo:</b>	-
<b>Estereotipo:</b>	Borde
<b>Propiedades:</b>	Concreta
<b>Superclases:</b>	pantallaConsulta
<b>Subclases:</b>	-
<b>Atributos:</b>	-
<b>1. Desplegar Pantalla</b>	
Desplegar() Devuelve void	
<b>3. Solicitar Confirmación</b>	
enviarInformacion() devuelve void	controlModificaciónEspacioPlanta
Limpiar() devuelve void	
<b>2. Mostrar Información</b>	
haceMostrar(ArrayList datos, int opcion) devuelve void	

getObjIntModificaEspacio() Devuelve VInterfaceModificacionEspacioPlanta		
<b>Tarjetas CRC</b>		
<b>Objeto:</b>	VInterfaceModificacionCliente	
<b>Descripción:</b>	Permite seleccionar el cliente al cual se le modificará su información	
<b>Módulo:</b>	-	
<b>Estereotipo:</b>	Borde	
<b>Propiedades:</b>	Concreta	
<b>Superclases:</b>	pantallaEnvio	
<b>Subclases:</b>	-	
<b>Atributos:</b>	-	
<b>1. Desplegar Pantalla</b>		
Desplegar() devuelve void		
<b>3. Solicitar Confirmación</b>		
enviarInformación() devuelve void		controlModificaciónCliente
<b>2. Mostrar Información</b>		
haceMostrar(ArrayList datos, int opcion) devuelve void		
getObjIntModificaPersonal() Devuelve VInterfaceModificacionCliente		
<b>Tarjetas CRC</b>		
<b>Objeto:</b>	VInterfaceEliminarPersonal	
<b>Descripción:</b>	Permite eliminar un empleado del sistema	
<b>Módulo:</b>		
<b>Estereotipo:</b>	Borde	
<b>Propiedades:</b>	Concreta	
<b>Superclases:</b>	pantallaEnvio	
<b>Subclases:</b>		
<b>Atributos:</b>		
<b>1. Desplegar Pantalla</b>		
Desplegar() devuelve void		
<b>2. Mostrar Información</b>		
haceMostrar(ArrayList datos, int opcion) devuelve void		
<b>3. Solicitar Confirmación</b>		
enviarInformación() devuelve void		controleliminarPersonal
getObjIntEliminarPersona() Devuelve VInterfaceEliminacionPersonal		
<b>Tarjetas CRC</b>		
<b>Objeto:</b>	VInterfaceEliminarEspacioPlanta	

<b>Descripción:</b>	Permite eliminar una zona del sistema
<b>Módulo:</b>	
<b>Estereotipo:</b>	Borde
<b>Propiedades:</b>	Concreta
<b>Superclases:</b>	pantallaEnvio
<b>Subclases:</b>	
<b>Atributos:</b>	
<b>1. Desplegar Pantalla</b>	
Desplegar() devuelve void	
<b>2. Mostrar Información</b>	
haceMostrar(ArrayList datos, int opcion) devuelve void	
<b>3. Solicitar Confirmación</b>	
enviarInformación() devuelve void	controlEliminarEspacioPlanta
getObjIntEliminarEspacio() devuelve VInterfaceEliminarEspacioPlanta	
<b>Tarjetas CRC</b>	
<b>Objeto:</b>	VInterfaceCreacionProducto
<b>Descripción:</b>	Recibe la información del nuevo producto a crear
<b>Módulo:</b>	-
<b>Estereotipo:</b>	Borde
<b>Propiedades:</b>	Concreta
<b>Superclases:</b>	pantallaEnvio
<b>Subclases:</b>	-
<b>Atributos:</b>	-
<b>1. Desplegar Pantalla</b>	
Desplegar() devuelve void	
<b>3. Solicitar confirmación</b>	
enviarInformación() devuelve void	<u>controlCreacionProducto</u>
Limpiar() devuelve void	
cargarCombo() Devuelve void	ControlCreacionProducto
Cerrar() Devuelve void	
<b>2. Mostrar Informacion</b>	
haceMostrar(ArrayList datos, int opcion) devuelve void	
getObjIntCreaProd() devuelve VInterfaceCreacionProducto	
<b>Tarjetas CRC</b>	
<b>Objeto:</b>	VInterfaceModificacionProceso
<b>Descripción:</b>	Permite seleccionar un proceso para modificar su información
<b>Módulo:</b>	-

<b>Estereotipo:</b>	Borde
<b>Propiedades:</b>	concreta
<b>Superclases:</b>	pantallaConsulta
<b>Subclases:</b>	
<b>Atributos:</b>	
<b>1. Desplegar Pantalla</b>	
Desplegar() devuelve void	
<b>3. Solicitar Confirmación</b>	
enviarInformacion() devuelve void	ControlModificacionProceso
<b>2. Mostrar Información</b>	
haceMostrar(ArrayList datos, int opcion) devuelve void	
Limpiar() devuelve void	
getObjIntModProc() devuelve VInterfaceModificacionProceso	
<b>Tarjetas CRC</b>	
<b>Objeto:</b>	VInterfaceModificacionProducto
<b>Descripción:</b>	Permite seleccionar un producto para modificar su información
<b>Módulo:</b>	-
<b>Estereotipo:</b>	Borde
<b>Propiedades:</b>	concreta
<b>Superclases:</b>	pantallaConsulta
<b>Subclases:</b>	
<b>Atributos:</b>	
<b>1. Desplegar Pantalla</b>	
Desplegar() devuelve void	
<b>3. Solicitar Confirmación</b>	
enviarInformacion() devuelve void	ControlModificacionProducto
<b>2. Mostrar Información</b>	
haceMostrar(ArrayList datos, int opcion) devuelve void	
Limpiar() devuelve void	
getObjIntModProd() devuelve VInterfaceModificacionProducto	

### Objetos Control.

<b>Tarjetas CRC</b>	
<b>Objeto:</b>	controlCierre
<b>Descripción:</b>	Objeto heredado por todos los controles que requieren

	realizar el cierre de un módulo	
<b>Módulo:</b>		
<b>Estereotipo:</b>	Control	
<b>Propiedades:</b>	Abstracta	
<b>Superclases:</b>		
<b>Subclases:</b>	controlMostrar, controlConfirmacion	
<b>Atributos:</b>		
<b>4. Cerrar módulo</b>		
	cerrarModulo() devuelve void	
<b>Tarjetas CRC</b>		
<b>Objeto:</b>	controlMostrar	
<b>Descripción:</b>	Objeto heredado por todos los controles que requieren solicitar mostrar sobre una interfaz	
<b>Módulo:</b>		
<b>Estereotipo:</b>	Control	
<b>Propiedades:</b>	Abstracta	
<b>Superclases:</b>	controlCierre	
<b>Subclases:</b>	controlDespliegue, controlCreacionPedido, controlCreacionContrato, controlAsignacionPersonal, controlAsignacionEspacioPlanta, controlAsignacionMateriaPrima	
<b>Atributos:</b>		
<b>2. Mostrar información</b>		
	solicitaMostrar (ArrayList datos) devuelve void	pantallaMostrar (2)
<b>Tarjetas CRC</b>		
<b>Objeto:</b>	controlDespliegue	
<b>Descripción:</b>	Objeto heredado por todos los controladores que requieren realizar el despliegue de una interfaz	
<b>Módulo:</b>		
<b>Estereotipo:</b>	Control	
<b>Propiedades:</b>	Abstracto	
<b>Superclases:</b>	controlMostrar	
<b>Subclases:</b>	controlConsulta, controlActualizacionPedido, controlEliminacionProceso, controlAsignacion, controlAñadirPersonal, controlAñadirEspacioPlanta,	

	controlIngresoMateriaPrima, controlCreacionProceso, controlEliminacionProducto, controladorPrincipal	
<b>Atributos:</b>		
<b>1. Desplegar Pantalla</b>		
solicitaDesplegar(Pantalla p) devuelve void	Pantalla (1)	
<b>Tarjetas CRC</b>		
<b>Objeto:</b>	controlConsulta	
<b>Descripción:</b>	Objeto encargado de realizar consultas a los objetos entidad	
<b>Módulo:</b>		
<b>Estereotipo:</b>	Control	
<b>Propiedades:</b>	Abstracta	
<b>Superclases:</b>	controlDespliegue	
<b>Subclases:</b>	controlValidacion, controlEliminarEspacioPlanta, controlVerificacion, controlConsultaActualizacionPedido, controlConsultaAsignacion, controlEliminacionPersonal, controlEntregaMaterial, controlModificacionPersonal, controlModificacionProceso, controlModificacionProducto	
<b>Atributos:</b>		
<b>7. Consultar</b>		
Consultar(int codigo) devuelve ArrayList	Datos	
<b>Tarjetas CRC</b>		
<b>Objeto:</b>	controlValidacion	
<b>Descripción:</b>	Objeto heredado por todos los controladores que requieren realizar la validación de un código sobre un objeto entidad	
<b>Módulo:</b>		
<b>Estereotipo:</b>	Control	
<b>Propiedades:</b>	Abstracta	
<b>Superclases:</b>	controlConsulta	
<b>Subclases:</b>	controlConsultaContrato, controlConsultaCotizacion, controlConsultaPedido, controlCreacionCotizacion, controlModificacionCliente, controlModificacionEspacioPlanta	
<b>Atributos:</b>		
<b>5. Validar código</b>		
Validar(int codigo) devuelve void	Datos	

<b>Tarjetas CRC</b>	
<b>Objeto:</b>	controlConfirmacion
<b>Descripción:</b>	Objeto heredado por todos los controladores que requieren solicitar confirmación sobre una interface
<b>Módulo:</b>	
<b>Estereotipo:</b>	Control
<b>Propiedades:</b>	Abstracta
<b>Superclases:</b>	controlCierre
<b>Subclases:</b>	controlMostrarMensaje, controlAsignacionPersonal, controlAsignacionMateriaPrima, controlAsignacionEspacioPlanta, controlEliminarEspacioPlanta
<b>Atributos:</b>	
<b>3. Solicitar confirmación</b>	
solicitarConfirmacion(ArrayList datos) devuelve void	ventanaConfirmacion
<b>Tarjetas CRC</b>	
<b>Objeto:</b>	controlMostrarMensaje
<b>Descripción:</b>	Objeto heredado por todos los controladores que requieren solicitar mostrar mensaje sobre una interface
<b>Módulo:</b>	
<b>Estereotipo:</b>	Control
<b>Propiedades:</b>	Abstracta
<b>Superclases:</b>	controlConfirmacion
<b>Subclases:</b>	controlCreacion, controlActualizacionPedido, controlEliminacionProceso, controlEliminacionPersonal, controlModificacionCliente, controlModificacionEspacioPlanta, controlModificacionPersonal, controlEliminacionProducto, controlModificacionProceso, controlModificacionProducto, controlEliminarPersonal
<b>Atributos:</b>	
<b>6. Mostrar Mensaje</b>	
solicitaMostrarMensaje() devuelve void	pantallaMostrarMensaje (6)

<b>Tarjetas CRC</b>	
<b>Objeto:</b>	controlCreacion
<b>Descripción:</b>	Objeto heredado por todos los controladores que generan nueva información a ser almacenada
<b>Módulo:</b>	
<b>Estereotipo:</b>	Control
<b>Propiedades:</b>	Abstracta
<b>Superclases:</b>	controlMostrarMensaje
<b>Subclases:</b>	controlAñadirEspacioPlanta, controlAñadirPersonal, controlCreacionContrato, controlCreacionProceso, controlIngresoMateriaPrima, controlCreacionPedido, controlEntregaMaterial, controlCreacionCotizacion
<b>Atributos:</b>	
<b>8. Crear</b>	
Crear() devuelve void	Datos (8)
<b>Tarjetas CRC</b>	
<b>Objeto:</b>	controladorPrincipal
<b>Descripción:</b>	Controlador encargado de solicitar el despliegue de la mayoría de interfaces y realizar la comunicación entre controladores especializados en una actividad
<b>Módulo:</b>	
<b>Estereotipo:</b>	Control
<b>Propiedades:</b>	Concreto
<b>Superclases:</b>	controlDespliegue
<b>Subclases:</b>	
<b>Atributos:</b>	
<b>1. Desplegar pantalla</b>	
solicitaDesplegar(VinterfaceActualizacionPedido) devuelve void	VinterfaceActualizaciónPedido
solicitaDesplegar(interfacelInicio) devuelve void	interfacelInicio
solicitaDesplegar(VPantallaValidacionCodigoPedido) devuelve void	VPantallaValidacionCodigoPedido
solicitaDesplegar(VinterfacelIngresoMatPrima ) devuelve void	VinterfacelIngresoMatPrima
solicitaDesplegar(VPedido) devuelve void	VPedido
solicitaDesplegar(VInterfaceAdministracion)	VInterfaceAdministracion

on) devuelve void	
solicitaDesplegar(VinterfaceVerificacionRecursosDisponibles) devuelve void	VinterfaceVerificacionRecursosDisponibles
solicitaDesplegar(InterfaceSuper) devuelve void	InterfaceSuper
solicitaDesplegar(InterfaceGerente) devuelve void	InterfaceGerente
solicitaDesplegar(InterfaceAlmacenista) devuelve void	InterfaceAlmacenista
solicitaDesplegar(VInterfaceEliminacion) devuelve void	VInterfaceEliminacion
solicitaDesplegar(VInterfaceCreacion) devuelve void	VInterfaceCreacion
solicitaDesplegar(VInterfaceModificacion) devuelve void	VInterfaceModificacion
solicitaDesplegar(VPantallaMostrarContrato) devuelve void	VPantallaMostrarContrato
solicitaDesplegar(VPantallaMostrarCotizacion) devuelve void	VPantallaMostrarCotizacion
solicitaDesplegar(VPantallaMostrarPedido) devuelve void	VPantallaMostrarPedido
solicitaDesplegar(VinterfaceCreacionProceso) devuelve void	VinterfaceCreacionProceso
solicitaDesplegar(VinterfaceCreacionEspacioPlanta) devuelve void	VinterfaceCreaciónEspacioPlanta
solicitaDesplegar(VinterfaceCrecionPersonal) devuelve void	VinterfaceCreaciónPersonal
solicitaDesplegar(VinterfaceConsultaContrato) devuelve void	VinterfaceConsultaContrato
solicitaDesplegar(VinterfaceConsultaCotizacion) devuelve void	VinterfaceConsultaCotizacion
solicitaDesplegar(VinterfaceConsultaPedido) devuelve void	VinterfaceConsultaPedido
solicitaDesplegar(interfaceConsultaActualizacionPedido) devuelve void	interfaceConsultaActualizacionPedido
solicitaDesplegar(interfaceEntregaMateriaPrima) devuelve void	interfaceEntregaMateriaPrima
solicitaDesplegar(interfaceEntregaMateriaPrima) devuelve void	
solicitaDesplegar(interfaceCreacionCotizacion) devuelve void	interfaceCreacionCotizacion
solicitaDesplegar(VVentanaValidacionCotizacion) devuelve void	VVentanaValidacionCotizacion

solicitaDesplegar(VVentanaValidacionContrato) devuelve void	VVentanaValidaciónContrato
crearNumeroCotizacion() devuelve int	
getFechaActual() devuelve String	
getIbjContPrinc() devuelve ControladorPrincipal	
<b>Tarjetas CRC</b>	
<b>Objeto:</b>	controlConsultaCotizacion
<b>Descripción:</b>	Controlador encargado de realizar validaciones de códigos de cotizaciones, así como de consultar y mostrar sobre interfaces información relacionada con esta.
<b>Módulo:</b>	
<b>Estereotipo:</b>	Control
<b>Propiedades:</b>	Concreta
<b>Superclases:</b>	controlValidacion
<b>Subclases:</b>	
<b>Atributos:</b>	objContConsCot (ControlConsultaCotizacion) objCotizacion (cotizacion) objCreCont (controlCreacionContrato)
<b>7. Consultar</b>	
Consultar(int codigo) devuelve ArrayList	cotización
<b>5. Validar código</b>	
Validar(int codigo) devuelve void	cotizacion
getObjConstConsCot() devuelve controlConsultaCotizacion	
<b>Responsabilidades Privadas</b>	
exportarInformacion (ArrayList datos)	
<b>Tarjetas CRC</b>	
<b>Objeto:</b>	controlCreacionContrato
<b>Descripción:</b>	Objeto encargado de controlar la creación de contratos en el sistema
<b>Módulo:</b>	
<b>Estereotipo:</b>	Control
<b>Propiedades:</b>	Concreta
<b>Superclases:</b>	controlCreacion, controlMostrar
<b>Subclases:</b>	
<b>Atributos:</b>	contCreaCont (ControlCreacionContrato) datosControl (ArrayList) objCont (Contrato)
<b>8. Crear</b>	
Crear() devuelve void	Contrato

<b>3. Solicitar Confirmación</b>		
solicitarConfirmacion(ArrayList datos) devuelve void		ventanaConfirmacion
<b>2. Mostrar Información</b>		
solicitaMostrar (ArrayList datos) devuelve void		VinterfaceCreacionContrato
getContCreaCont() devuelve ControlCreacionContrato		
<b>Tarjetas CRC</b>		
<b>Objeto:</b>	controlConsultaActualizacionPedido	
<b>Descripción:</b>	Objeto encargado de controlar todas las operaciones de consulta a actualizaciones	
<b>Módulo:</b>		
<b>Estereotipo:</b>	Control	
<b>Propiedades:</b>	Concreta	
<b>Superclases:</b>	controlConsulta	
<b>Subclases:</b>		
<b>Atributos:</b>	objCtrlConsulta (ControlConsultaActualizacionPedido) objPedido (Pedido)	
<b>7. Consultar</b>		
Consultar(int codigo) devuelve ArrayList		pedido
<b>1. Desplegar Pantalla</b>		
solicitaDesplegar(VpantallaResultadoBusqueda) devuelve void		VPantallaResultadoBusqueda
getObjCtrlConsulta() devuelve ControlConsultaActualizacion		
<b>Tarjetas CRC</b>		
<b>Objeto:</b>	controlConsultaAsignacion	
<b>Descripción:</b>	Objeto encargado de realizar todas las consultas a realizar a la asignación de recursos	
<b>Módulo:</b>		
<b>Estereotipo:</b>	Control	
<b>Propiedades:</b>	Concreta	
<b>Superclases:</b>	controlConsulta	
<b>Subclases:</b>		
<b>Atributos:</b>	objConsAsi (ControlConsultaAsignacion)	
<b>7. Consultar</b>		
Consultar(int codigo) devuelve ArrayList		espacioPlanta
Consultar(int codigo) devuelve ArrayList		personal
Consultar(int codigo) devuelve ArrayList		materiaPrima

getObjConsAsi() devuelve ControlConsultaAsignacion		
<b>Tarjetas CRC</b>		
<b>Objeto:</b>	controlIngresoMateriaPrima	
<b>Descripción:</b>	Objeto encargado de controlar el ingreso de materia prima en el sistema	
<b>Módulo:</b>		
<b>Estereotipo:</b>	Control	
<b>Propiedades:</b>	Concreta	
<b>Superclases:</b>	controlCreacion, controlDespliegue	
<b>Subclases:</b>		
<b>Atributos:</b>	datos (ArrayList) objContIngM (ControlIngresoMateriaPrima)	
<b>4. Cerrar Módulo</b>		
cerrarModulo() devuelve void		
<b>8. Crear</b>		
Crear() devuelve void		materiaPrima
<b>6. Mostrar Mensaje</b>		
solicitaMostrarMensaje() devuelve void		pantallaMensaje
<b>3. Solicitar Confirmación</b>		
solicitarConfirmacion(ArrayList datos) devuelve void (360)		ventanaConfirmacion
getObjContIngM() devuelve ControlIngresoMateriaPrima		
<b>Tarjetas CRC</b>		
<b>Objeto:</b>	controlAsignacion	
<b>Descripción:</b>	Objeto encargado de controlar el proceso de asignación de los diferentes recursos en el sistema	
<b>Módulo:</b>		
<b>Estereotipo:</b>	Control	
<b>Propiedades:</b>	Concreta	
<b>Superclases:</b>	controlDespliegue	
<b>Subclases:</b>		
<b>Atributos:</b>	objContAsig (ControlAsignacion) objContEspP (ControlAsignacionEspacioPlanta) objContMatP (ControlAsignacionMateriaPrima) objContPer (ControlAsignacionPersonal) objContVer (ControlVerificacion)	
<b>1. Desplegar pantalla</b>		
solicitaDesplegar(VInterfaceAsignacion		VInterfaceAsignacionEspacioPlant

EspacioPlanta) devuelve void	a
solicitaDesplegar(VInterfaceAsignacionMateriaPrima) devuelve void	VInterfaceAsignacionMateriaPrima
solicitaDesplegar(VInterfaceAsignacionPersonal) devuelve void	VInterfaceAsignacionPersonal
<b>2. Mostrar Información</b>	
solicitaMostrar(ArrayList datos) devuelve void	VInterfaceCreacionPedido
solicitaMostrar1 (ArrayList datos) devuelve void	VInterfaceCreacionPedido
solicitaMostrar2 (ArrayList datos) devuelve void	VInterfaceCreacionPedido
getObjContAsig () devuelve ControlAsignacion	
<b>Responsabilidades Privadas</b>	
consultarEspacioPlanta () devuelve void	espacioPlanta
consultarMateriaPrima () devuelve void	materiaPrima
consultarPersonal ()() devuelve void	personal
<b>Tarjetas CRC</b>	
<b>Objeto:</b>	controlVerificacion
<b>Descripción:</b>	Controlador encargado de verificar la disponibilidad de los diferentes recursos
<b>Módulo:</b>	
<b>Estereotipo:</b>	Control
<b>Propiedades:</b>	Concreta
<b>Superclases:</b>	controlConsulta
<b>Subclases:</b>	
<b>Atributos:</b>	codigo (int) datosEspacioPlanta (ArrayList) datosPersonal(ArrayList) datosMateriaPrima (ArrayList) objContVer (ControlVerificacion) objEspPla (espacioPlanta) objMatPr (materiaPrima) objPersona (personal)
<b>7. Consultar</b>	
Consultar(int codigo) devuelve ArrayList	espacioPlanta
Consultar(int codigo) devuelve ArrayList	personal
Consultar(int codigo) devuelve ArrayList	materiaPrima
<b>2. Mostrar Información</b>	
solicitaMostrar (ArrayList datos) devuelve void	VInterfaceVerificacionRecursosDisponibles

getObjContVer() devuelve ControlVerificacion		
<b>Tarjetas CRC</b>		
<b>Objeto:</b>	controlAsignacionPersonal	
<b>Descripción:</b>	Objeto encargado de controlar y realizar la asignación de personal en el sistema	
<b>Módulo:</b>		
<b>Estereotipo:</b>	Control	
<b>Propiedades:</b>	Concreta	
<b>Superclases:</b>	controlConfirmacion, controlMostrar	
<b>Subclases:</b>		
<b>Atributos:</b>	objContAsPer (controlAsignacionPersonal) objIntAsPer (VInterfaceAsignacionPersonal) objPersonal (Personal)	
<b>3. Solicitar confirmación</b>		
solicitarConfirmacion(ArrayList datos) devuelve void	ventanaConfirmacion	
<b>2. Mostrar Información</b>		
solicitaMostrar (ArrayList datos) devuelve void	VInterfaceAsignacionPersonal	
getObjContAsPer() devuelve ControlAsignacionPersonal		
solicitarAsignacion (ArrayList datosPer) devuelve void	personal	
<b>Tarjetas CRC</b>		
<b>Objeto:</b>	controlAsignacionEspacioPLanta	
<b>Descripción:</b>	Objeto encargado de controlar y realizar la asignación de especio en planta en el sistema	
<b>Módulo:</b>		
<b>Estereotipo:</b>	Control	
<b>Propiedades:</b>	Concreta	
<b>Superclases:</b>	controlConfirmacion, controlMostrar	
<b>Subclases:</b>		
<b>Atributos:</b>	objConEspPla (ControlAsignacionEspacioPlanta) objEspPla (EspacioPlanta) objIntAsiEspP (VInterfaceAsignacionEspacioPlanta)	
<b>3. Solicitar Confirmación</b>		
solicitarConfirmacion(ArrayList datos) devuelve void	ventanaConfirmacion	
<b>2. Mostrar Información</b>		
solicitaMostrar (ArrayList datos)	VInterfaceAsignacionEspacioPlant	

devuelve void	a
getObjConEspPla() devuelve ControlAsignacionEspacioPlanta	
solicitarAsignacion (ArrayList datosEspPla) devuelve void	espacioPlanta
<b>Tarjetas CRC</b>	
<b>Objeto:</b>	controlAsignacionMateriaPrima
<b>Descripción:</b>	Objeto encargado de controlar y realizar la asignación de materia prima en el sistema
<b>Módulo:</b>	
<b>Estereotipo:</b>	Control
<b>Propiedades:</b>	Concreta
<b>Superclases:</b>	controlConfirmacion, controlMostrar
<b>Subclases:</b>	
<b>Atributos:</b>	objContMatPri (ControlAsignacionMateriaPrima) objIntMatP (VInterfaceAsignacionMateriaPrima) objMatPri (materiaPrima)
<b>3. Solicitar Confirmacion</b>	
solicitarConfirmacion(ArrayList datos) devuelve void	ventanaConfirmacion
<b>2. Mostrar Informacion</b>	
solicitaMostrar (ArrayList datos ) devuelve void	VInterfaceAsignacionMateriaPrima
getObjContMatPri() devuelve ControlAsignacionMateriaPrima	
solicitarAsignacion(ArrayList datosMat) devuelve void	materiaPrima
<b>Tarjetas CRC</b>	
<b>Objeto:</b>	controlConsultaContrato
<b>Descripción:</b>	Objeto encargado de las operaciones de consulta de contrato que se hagan en el sistema
<b>Módulo:</b>	
<b>Estereotipo:</b>	Control
<b>Propiedades:</b>	Concreta
<b>Superclases:</b>	controlValidacion
<b>Subclases:</b>	
<b>Atributos:</b>	objCont (Contrato) objContConsCot (ControlConsultaContrato)
<b>7. Consultar</b>	
Consultar(int codigo) devuelve ArrayList	contrato

<b>5. Validar código</b>		
Validar(int codigo) devuelve void		contrato
getObjContConsCont() devuelve ControlConsultaContrato		
<b>Responsabilidades privadas</b>		
exportarInformacion (ArrayList datos) devuelve void		controlCreacionPedido
<b>Tarjetas CRC</b>		
<b>Objeto:</b>	controlCreacionProceso	
<b>Descripción:</b>	Objeto encargado de la creación de un nuevo proceso en el sistema	
<b>Módulo:</b>		
<b>Estereotipo:</b>	Control	
<b>Propiedades:</b>	Concreta	
<b>Superclases:</b>	controlCreacion, controlDespliegue	
<b>Subclases:</b>		
<b>Atributos:</b>	Datos (ArrayList) objCtrlCreaProceso(ControlCreacionProceso)	
<b>4. Cerrar Módulo</b>		
cerrarModulo() devuelve void		VInterfaceCreacionProceso
<b>1. Desplegar Pantalla</b>		
solicitarDesplegar(VInterfaceCreacion) devuelve void		VInterfaceCreacion
<b>8. Crear</b>		
Crear() devuelve void		proceso
<b>6. Mostrar Mensaje</b>		
solicitaMostrarMensaje() devuelve void		ventanaMensaje
<b>3. Solicitar Confirmación</b>		
solicitarConfirmacion(ArrayList datos) devuelve void		ventanaConfirmacion
cargarTipoProducto() devuelve ArrayList()		proceso
getObjCtrlCreaProceso() devuelve ControlCreacionProceso		
<b>Tarjetas CRC</b>		
<b>Objeto:</b>	controlEliminacionProceso	
<b>Descripción:</b>	Objeto encargado de controlar el proceso de eliminación de un proceso en el sistema	
<b>Módulo:</b>		
<b>Estereotipo:</b>	Control	
<b>Propiedades:</b>	Concreta	

<b>Superclases:</b>	controlMostrarMensaje, controlDespliegue	
<b>Subclases:</b>		
<b>Atributos:</b>	datosE (ArrayList) objEliminaProceso (ControlEliminacionProceso) objProceso (Proceso)	
<b>4. Cerrar Módulo</b>		
cerrarModulo() devuelve void	VInterfaceEliminacionProceso	
<b>7. Consultar</b>		
Consultar() devuelve void	proceso	
<b>1. Desplegar Pantalla</b>		
solicitaDesplegar(interfaceEliminacionProceso) devuelve void	VinterfaceEliminacionProceso	
solicitaDesplegar(VinterfaceEliminacion) devuelve void	VinterfaceEliminacion	
<b>6. Mostrar Mensaje</b>		
solicitaMostrarMensaje() devuelve void	ventanaMensaje	
<b>3. Solicitar confirmación</b>		
solicitarConfirmacion(ArrayList datos) devuelve void	ventanaConfirmacion	
<b>2. Mostrar Información</b>		
solicitaMostrar (ArrayList datos) devuelve void	VinterfaceEliminacionProceso	
Eliminar() devuelve void	proceso	
getObjEliminaProceso() devuelve ControlEliminacionProceso		
<b>Tarjetas CRC</b>		
<b>Objeto:</b>	controlActualizacionPedido	
<b>Descripción:</b>	Objeto encargado de controlar el proceso de actualización de actividades en un pedido	
<b>Módulo:</b>		
<b>Estereotipo:</b>	Control	
<b>Propiedades:</b>	Concreta	
<b>Superclases:</b>	controlMostrarMensaje, controlDespliegue	
<b>Subclases:</b>		
<b>Atributos:</b>	datosA (ArrayList) objCtrlActualizaPedido (ControlActualizacionPedido) objCtrlConsultaPedido (ControlConsultaPedido) objPedido (Pedido)	
<b>4. Cerrar Módulo</b>		
cerrarModulo() devuelve void	VFormularioIngresoActualizacion	
getObjCtrlActualizaPedido() devuelve		

controlActualizacionPedido		
<b>6. Mostrar Mensaje</b>		
solicitaMostrarMensaje() devuelve void		ventanaMensaje
<b>3. Solicitar confirmación</b>		
solicitarConfirmacion(ArrayList datos) devuelve void		ventanaConfirmación
actualizarPedido(ArrayList datosA)		pedido
<b>Tarjetas CRC</b>		
<b>Objeto:</b>	controlCreacionCotizacion	
<b>Descripción:</b>	Objeto encargado de controlar la creación de una nueva cotización en el sistema	
<b>Módulo:</b>		
<b>Estereotipo:</b>	Control	
<b>Propiedades:</b>	Concreta	
<b>Superclases:</b>	controlValidacion	
<b>Subclases:</b>		
<b>Atributos:</b>	contCreaCoti (controlCreacionCotizacion) datosCoti(ArrayList) objCli(Cliente) objCotizaicon(Cotizacion)	
<b>8. Crear</b>		
Crear() devuelve void		cotización
Crear() devuelve void		cliente
<b>6. Mostrar Mensaje</b>		
solicitaMostrarMensaje () devuelve void		ventanaMensaje
<b>3. Solicitar Confirmación</b>		
solicitarConfirmación (ArrayList datos)		ventanaConfirmación
generarCodigoCotización()devuelve void		cotizacion
getContCreaCoti () devuelve ControlCreacionCotizacion		
<b>Tarjetas CRC</b>		
<b>Objeto:</b>	controlCreacionPedido	
<b>Descripción:</b>	Objeto encargado de controlar la creación de nuevos pedidos en el sistema	
<b>Módulo:</b>		
<b>Estereotipo:</b>	Control	
<b>Propiedades:</b>	Concreta	
<b>Superclases:</b>	controlCreacion, controlMostrar	
<b>Subclases:</b>		

<b>Atributos:</b>	datosControl (ArrayList) objContCrePe (ControlCreacionPedido) objPed (Pedido)	
<b>8. Crear</b>		
crear() devuelve void		pedido
<b>6. Mostrar Mensaje</b>		
solicitaMostrarMensaje() devuelve void		ventanaMensaje
<b>2. Mostrar Información</b>		
solicitarMostrar(java.util.ArrayList datos) devuelve void		VInterfaceCreacionPedido
getObjContCrePe() devuelve ControlCreacionPedido		
<b>Tarjetas CRC</b>		
<b>Objeto:</b>	controlAñadirPersonal	
<b>Descripción:</b>	Objeto encargado de controlar el proceso de añadir un nuevo empleado al sistema	
<b>Módulo:</b>		
<b>Estereotipo:</b>	Control	
<b>Propiedades:</b>	Concreta	
<b>Superclases:</b>	controlCreacion, controlDespliegue	
<b>Subclases:</b>		
<b>Atributos:</b>	Datos (Arraylist) objCtrlCrearPersonal (ControlAñadirPersonal)	
<b>4. Cerrar módulo</b>		
cerrarModulo() devuelve void		VInterfaceCreacionPersonal
<b>8. Crear</b>		
Crear()		personal
<b>1. Desplegar Pantalla</b>		
Solicita desplegar(VInterfaceCreacion) devuelve void		VInterfaceCreacion
<b>6. Mostrar Mensaje</b>		
solicitaMostrarMensaje() devuelve void		pantallaResultado
<b>3. Solicitar Confirmación</b>		
solicitarConfirmacion(Arraylist datos) devuelve void (147)		ventanaConfirmación
getObjCtrlCrearPersonal() devuelve ControlAñadirPersonal		
<b>Tarjetas CRC</b>		
<b>Objeto:</b>	controlAñadirEspacioPlanta	
<b>Descripción:</b>	Objeto encargado de controlar el proceso de añadir una	

	nueva zona de fabricación en el sistema
<b>Módulo:</b>	
<b>Estereotipo:</b>	Control
<b>Propiedades:</b>	Concreta
<b>Superclases:</b>	controlCreacion, controlDespliegue
<b>Subclases:</b>	
<b>Atributos:</b>	Datos (ArrayList) objCtrlAñadirEspacio (ControlAñadirEspacioPlanta)
<b>4. Cerrar Módulo</b>	
cerrarModulo() devuelve void	VInterfaceCreacionEspacioPlanta
<b>8. Crear</b>	
Crear() devuelve void	EspacioPlanta
<b>1. Desplegar Pantalla</b>	
Solicitudesplegar(VInterfaceCreacion) devuelve void	VInterfaceCreacion
<b>6. Mostrar Mensaje</b>	
solicitaMostrarMensaje() devuelve void	ventanaMensaje
<b>3. Solicitar Confirmación</b>	
solicitarConfirmacion(ArrayList datos) devuelve void	ventanaConfirmación
getObjCtrlAñadirEspacio() devuelve ControlAñadirEspacioPlanta	
<b>Tarjetas CRC</b>	
<b>Objeto:</b>	controlModificacionPersonal
<b>Descripción:</b>	Objeto encargado de realizar la modificación de los datos de un empleado en el sistema
<b>Módulo:</b>	
<b>Estereotipo:</b>	Control
<b>Propiedades:</b>	Concreta
<b>Superclases:</b>	controlMostrarMensaje, controlConsulta
<b>Subclases:</b>	
<b>Atributos:</b>	datosP (ArrayList) datosPersona (ArrayList) objCtrlModificaPersonal (ControlModifiacionPersonal) objIntModificaPersona(VInterfaceModificacionPersonal) objPersona (Personal)
<b>4. Cerrar Módulo</b>	
cerrarModulo() devuelve void	VInterfaceModificacionPersonal()
<b>7. Consultar</b>	
Consultar(int codigo) devuelve ArrayList	personal
<b>1. Desplegar pantalla</b>	

solicitaDesplegar(VInterfaceModificacion) devuelve void	VInterfaceModificacion
solicitaDesplegar(formularioPersonal) devuelve void (185*)(195*)	VInterfaceModificacionPersonal
<b>6. Mostrar Mensaje</b>	
solicitaMostrarMensaje() devuelve void	ventanaMensaje
<b>3. Solicitar confirmación</b>	
solicitarConfirmacion(ArrayList datos) (189)	ventanaConfirmación
<b>2. Mostrar Información</b>	
solicitaMostrar (ArrayList datos) devuelve void (186*)	VInterfaceModificacionPersonal
getObjCtrlModificarPersonal() devuelve ControlModificacionPersonal	
Modificar(ArrayList datos) devuelve void	personal
<b>Tarjetas CRC</b>	
<b>Objeto:</b>	controlModificacionEspacioPlanta
<b>Descripción:</b>	Objeto encargado de controlar la modificación de los datos de una zona en el sistema
<b>Módulo:</b>	
<b>Estereotipo:</b>	Control
<b>Propiedades:</b>	Concreta
<b>Superclases:</b>	controlMostrarMensaje, controlValidacion
<b>Subclases:</b>	
<b>Atributos:</b>	datosEspacio (Arraylist) datos (ArrayList) objCtrlModificaEspacio (ControlModificacionEspacioPlanta) objEspacio (EspacioPlanta) objIntModificaEspacio (VInterfaceModificacionEspacioPlanta)
<b>4. cerrar Módulo</b>	
cerrarModulo() devuelve void	VInterfaceModificacionEspacioPlanta
<b>7. Consultar</b>	
Consultar(int codigo) devuelve ArrayList	EspacioPlanta
<b>1. Desplegar Pantalla</b>	
solicitaDesplegar(VInterfaceModificacion) devuelve void	VInterfaceModificacion
solicitaDesplegar(VInterfaceModificacionEspacioPlanta) devuelve void	VInterfaceModificacionEspacioPlanta
<b>6. Mostrar Mensaje</b>	

solicitaMostrarMensaje() devuelve void	ventanaMensaje
<b>3. Solicitar Confirmación</b>	
solicitarConfirmacion(ArrayList datos) devuelve void	ventanaConfirmación
<b>2. Mostrar Información</b>	
solicitaMostrar (ArrayList datos) devuelve void	VInterfaceModificacionEspacioPlan ta
getObjCtrlModificaEspacio() devuelve ControlModifiacionEspacioPlanta	
Modificar(ArrayList datos)	espacioPlanta
<b>Tarjetas CRC</b>	
<b>Objeto:</b>	controlModificacionCliente
<b>Descripción:</b>	Objeto encargado de controlar la modificación de los datos de un cliente en el sistema
<b>Módulo:</b>	
<b>Estereotipo:</b>	Control
<b>Propiedades:</b>	Concreta
<b>Superclases:</b>	controlMostrarMensaje, controlValidacion
<b>Subclases:</b>	
<b>Atributos:</b>	datosC (ArrayList) datosCliente (ArrayList) objCliente (cliente) objCtrlModificaCliente (ControlModificacionCliente) objIntModificaCliente (VInterfaceModificacionCliente)
<b>4. Cerrar Módulo</b>	
cerrarModulo() devuelve void	VInterfaceModificacion
<b>7. Consultar</b>	
Consultar(int codigo ) devuelve ArrayList	Cliente
<b>1. Desplegar Pantalla</b>	
solicitaDesplegar(VInterfaceModificaci on) devuelve void	VInterfaceModificacion
solicitaDesplegar(VInterfaceModificaci onEspacioPlanta) devuelve void	VInterfaceModificacionCliente
<b>6. Mostrar Mensaje</b>	
solicitaMostrarMensaje() devuelve void	ventanaMensaje
<b>3. Solicitar Confirmación</b>	
solicitarConfirmacion(ArrayList datos) devuelve void	ventanaConfirmación
<b>2. Mostrar Información</b>	
solicitaMostrar (ArrayList datos)	VInterfaceModificacionCliente

devuelve void	
getObjCtrlModificaCliente() return ControlModificacionCliente	
modificar(ArrayList datos) devuelve void	Cliente
<b>Tarjetas CRC</b>	
<b>Objeto:</b>	controlEliminarPersonal
<b>Descripción:</b>	Objeto encargado de controlar la eliminación de un empleado en el sistema
<b>Módulo:</b>	
<b>Estereotipo:</b>	Control
<b>Propiedades:</b>	Concreta
<b>Superclases:</b>	ControlConsulta, controlMostrarMensaje
<b>Subclases:</b>	
<b>Atributos:</b>	datosE (ArrayList) datosPersona (ArrayList) objCtrlEliminaPersona (ControlEliminacionPersonal) objIntEliminaPersona (VInterfaceEliminarPersonal) objPersona (Personal)
<b>4. Cerrar Módulo</b>	
cerrarModulo() devuelve void	VInterfaceEliminacionPersonal
<b>7. Consultar</b>	
Consultar(int codigo) devuelve ArrayList	personal
<b>1. Desplegar Pantalla</b>	
solicitaDesplegar(VInterfaceEliminacionPersonal) devuelve void	VInterfaceEliminacionPersonal
solicitaDesplegar(VInterfaceEliminacionPersonal) devuelve void (267*)	VInterfaceEliminacion
<b>6. Mostrar Mensaje</b>	
solicitaMostrarMensaje() devuelve void	ventanaMensaje
<b>3. Solicitar confirmación</b>	
solicitarConfirmacion(ArrayList datos) (261)	ventanaConfirmación
<b>2. Mostrar Información</b>	
solicitaMostrar (ArrayList datos) devuelve void (258)	VInterfaceEliminacionPersonal
eliminar() devuelve void	personal
getObjCtrlEliminaPersona() devuelve ControlEliminacionPersonal	
<b>Tarjetas CRC</b>	

<b>Objeto:</b>	controlEliminarEspacioPlanta
<b>Descripción:</b>	Objeto encargado de controlar la eliminación de una zona en el sistema
<b>Módulo:</b>	
<b>Estereotipo:</b>	Control
<b>Propiedades:</b>	Concreta
<b>Superclases:</b>	controlConfirmacion, controlConsulta
<b>Subclases:</b>	
<b>Atributos:</b>	datosE (ArrayList) datosEspacio (ArrayList) objCtrlEliminaEspacio (ControlEliminacionEspacioPlanta) objEspacio (EspacioPlanta) objIntEliminaEspacio (VInterfaceEliminarEspacioPlanta)
<b>4. Cerrar Módulo</b>	
cerrarModulo() devuelve void	VInterfaceEliminarEspacioPlanta
<b>7. Consultar</b>	
consultar(int codigo) devuelve ArrayList	EspacioPlanta
<b>1. Desplegar Pantalla</b>	
solicitaDesplegar(VInterfaceEliminacionEspacioPlanta) devuelve void	VInterfaceEliminacionEspacioPlanta
solicitaDesplegar(VInterfaceEliminacion) devuelve void (267*)	VInterfaceEliminacion
<b>6. Mostrar Mensaje</b>	
solicitaMostrarMensaje() devuelve void	ventanaMensaje
<b>3. Solicitar Confirmación</b>	
solicitarConfirmacion(ArrayList datos) devuelve void	ventanaConfirmación
<b>2. Mostrar Información</b>	
solicitarMostrar(ArrayList datos) devuelve void	VInterfaceEliminacionEspacioPlanta
eliminar() devuelve void	EspacioPlanta
getObjCtrlEliminaEspacio() devuelve ControlEliminacionEspacioPlanta	
<b>Tarjetas CRC</b>	
<b>Objeto:</b>	controlConsultaPedido
<b>Descripción:</b>	Objeto encargado de controlar todas las consultas hechas a un pedido, así como de la validación de códigos de este
<b>Módulo:</b>	
<b>Estereotipo:</b>	Control

<b>Propiedades:</b>	Concreta
<b>Superclases:</b>	controlValidacion
<b>Subclases:</b>	
<b>Atributos:</b>	Opcion (int) contConsPed (ControlConsultaPedido) objIntActualizaPedido (VInterfaceActualizacionPedido) objPed (Pedido) objProceso (Proceso)
<b>7. Consultar</b>	
consultar(int codigo) devuelve ArrayList	pedido
consultar(int codigo) devuelve ArrayList	pedido
<b>1. Desplegar Pantalla</b>	
solicitaDesplegar(VInterfaceConsultaActualizacionPedido) devuelve void	VInterfaceConsultaActualizacionPedido
solicitaDesplegar(VFormularioIngresoActualizacion) devuelve void	VFormularioIngresoActualizacion
<b>2. Mostrar Información</b>	
solicitarMostrar(ArrayList datos) devuelve void	VInterfaceConsultaActualizacionPedido
solicitarMostrar(ArrayList datos) devuelve void	VFormularioIngresoActualizacion
solicitarMostrar(ArrayList datos) devuelve void	VInterfaceConsultaPedido
<b>5. validar Código</b>	
validar(int codigo) devuelve void	pedido
getContConsPed() devuelve ControlConsultaPedido	
paginarArray(ArrayList datos) devuelve void	
<b>Tarjetas CRC</b>	
<b>Objeto:</b>	controlEntregaMaterial
<b>Descripción:</b>	Objeto encargado de controlar el proceso de entrega de material
<b>Módulo:</b>	
<b>Estereotipo:</b>	Control
<b>Propiedades:</b>	Concreta
<b>Superclases:</b>	controlCreacion, controlDespliegue
<b>Subclases:</b>	
<b>Atributos:</b>	datosEntrega (ArrayList) objContEntMat (ControlEntregaMaterial)
<b>4. Cerrar Módulo</b>	

cerrarModulo() devuelve void	VInterfaceEntregaMateriaPrima
<b>7. Consultar</b>	
consultar(int codigo) devuelve ArrayList	Personal
consultar(int codigo) devuelve ArrayList	materiaPrima
<b>6. Mostrar Mensaje</b>	
solicitaMostrarMensaje() devuelve void	ventanaMensaje
<b>3. Solicitar Confirmación</b>	
solicitarConfirmacion(ArrayList datos) devuelve void	ventanaConfirmacion
<b>8. Crear</b>	
crear() devuelve void	materiaPrima
<b>2. Mostrar Información</b>	
solicitarMostrar(ArrayList datos) devuelve void	VinterfaceEntregaMatPrima
getObjContEntMat() devuelve ControlEntregaMaterial	
<b>Tarjetas CRC</b>	
<b>Objeto:</b>	controlModificacionProceso
<b>Descripción:</b>	Objeto encargado de realizar la modificación de los datos de un proceso en el sistema
<b>Módulo:</b>	
<b>Estereotipo:</b>	Control
<b>Propiedades:</b>	Concreta
<b>Superclases:</b>	controlMostrarMensaje, controlConsulta
<b>Subclases:</b>	
<b>Atributos:</b>	
<b>4. Cerrar Módulo</b>	
cerrarModulo() devuelve void	VInterfaceModificacionProceso()
<b>7. Consultar</b>	
Consultar(int codigo) devuelve ArrayList	proceso
<b>1. Desplegar pantalla</b>	
solicitaDesplegar(VInterfaceModificacio n) devuelve void	VInterfaceModificacion
solicitaDesplegar(formularioPersonal) devuelve void (185*)(195*)	VInterfaceModificacionProceso
<b>6. Mostrar Mensaje</b>	
solicitaMostrarMensaje() devuelve void	ventanaMensaje
<b>3. Solicitar confirmación</b>	
solicitarConfirmacion(ArrayList datos) (189)	ventanaConfirmación
<b>2. Mostrar Información</b>	

solicitaMostrar (ArrayList datos) devuelve void (186*)	VInterfaceModificacionProceso
getObjCtrlModificarProceso() devuelve ControlModificacionPersonal	
Modificar(ArrayList datos) devuelve void	proceso
<b>Tarjetas CRC</b>	
<b>Objeto:</b>	controlModificacionProducto
<b>Descripción:</b>	Objeto encargado de realizar la modificación de los datos de un producto en el sistema
<b>Módulo:</b>	
<b>Estereotipo:</b>	Control
<b>Propiedades:</b>	Concreta
<b>Superclases:</b>	controlMostrarMensaje, controlConsulta
<b>Subclases:</b>	
<b>Atributos:</b>	
<b>4. Cerrar Módulo</b>	
cerrarModulo() devuelve void	VInterfaceModificacionProducto()
<b>7. Consultar</b>	
Consultar(int codigo) devuelve ArrayList	producto
<b>1. Desplegar pantalla</b>	
solicitaDesplegar(VInterfaceModificacion) devuelve void	VInterfaceModificacion
solicitaDesplegar(VInterfaceModificacionProducto) devuelve void (185*)(195*)	VInterfaceModificacionProducto
<b>6. Mostrar Mensaje</b>	
solicitaMostrarMensaje() devuelve void	ventanaMensaje
<b>3. Solicitar confirmación</b>	
solicitarConfirmacion(ArrayList datos) (189)	ventanaConfirmación
<b>2. Mostrar Información</b>	
solicitaMostrar (ArrayList datos) devuelve void (186*)	VInterfaceModificacionProducto
getObjCtrlModificarProducto() devuelve ControlModificacionPersonal	
Modificar(ArrayList datos) devuelve void	producto
<b>Tarjetas CRC</b>	
<b>Objeto:</b>	controlEliminacionProducto
<b>Descripción:</b>	Objeto encargado de controlar el proceso de eliminación de un producto en el sistema
<b>Módulo:</b>	

<b>Estereotipo:</b>	Control
<b>Propiedades:</b>	Concreta
<b>Superclases:</b>	controlMostrarMensaje, controlDespliegue
<b>Subclases:</b>	
<b>Atributos:</b>	
<b>4. Cerrar Módulo</b>	
cerrarModulo() devuelve void	VinterfaceEliminacionProducto
<b>7. Consultar</b>	
Consultar() devuelve void	Producto
<b>1. Desplegar Pantalla</b>	
solicitaDesplegar(interfaceEliminacionProducto) devuelve void	VinterfaceEliminacionProducto
solicitaDesplegar(VinterfaceEliminacion) devuelve void	VinterfaceEliminacion
<b>6. Mostrar Mensaje</b>	
solicitaMostrarMensaje() devuelve void	ventanaMensaje
<b>3. Solicitar confirmación</b>	
solicitarConfirmacion(ArrayList datos) devuelve void	ventanaConfirmacion
<b>2. Mostrar Información</b>	
solicitaMostrar (ArrayList datos) devuelve void	VinterfaceEliminacionProducto
Eliminar() devuelve void	Producto
getObjEliminaProceso() devuelve ControlEliminacionProducto	

## Entidad.

Tarjetas CRC	
<b>Objeto:</b>	Datos
<b>Descripción:</b>	Objeto heredado por todos los objetos entidad quienes son los encargados de suministrar la información a los objetos control
<b>Módulo:</b>	
<b>Estereotipo:</b>	Entidad
<b>Propiedades:</b>	Abstracta
<b>Superclases:</b>	
<b>Subclases:</b>	Cotización, contrato, pedido, actualización, materiaPrima, personal, espacioPlanta, asignación, proceso, recursos, entidad, cliente
<b>Atributos:</b>	

<b>5. Validar código</b>		
retornarValidacion(int codigo) devuelve boolean		controlValidacion
<b>7. Consultar</b>		
retornarConsulta(int codigo) devuelve ArrayList		controlConsulta
<b>8. Crear</b>		
retornarCreacion(ArrayList datos) devuelve boolean		controlCreacion
<b>Tarjetas CRC</b>		
<b>Objeto:</b>	cotizacion	
<b>Descripción:</b>	Contiene toda la información pertinente a las cotizaciones creadas en el sistema	
<b>Módulo:</b>		
<b>Estereotipo:</b>	Entidad	
<b>Propiedades:</b>	Concreta	
<b>Superclases:</b>	Datos	
<b>Subclases:</b>		
<b>Atributos:</b>	cod (int) con (conectar) cotizacion (Cotizacion) datos (ArrayList) reg (Connection) sql (String)	
<b>5. Validar Código</b>		
retornarValidacion(int codigo) devuelve boolean		controlConsultaCotizacion
<b>7. Consultar</b>		
ArrayList retornarConsulta(int codigo)		controlConsultaCotizacion
<b>8. Crear</b>		
retornarCreacion(ArrayList datos) return boolean		controlCreacionCotizacion
consultarCodigoCotizacion() devuelve int		
getCotizacion() devuelve Cotizacion		
<b>Tarjetas CRC</b>		
<b>Objeto:</b>	Contrato	
<b>Descripción:</b>	Contiene toda la información pertinente a los contratos creados en el sistema	
<b>Módulo:</b>		

<b>Estereotipo:</b>	Entidad
<b>Propiedades:</b>	Concreta
<b>Superclases:</b>	Datos
<b>Subclases:</b>	
<b>Atributos:</b>	Con (Conectar) Reg (Connection) Sql (String) Datos (ArrayList) Contrato (Contrato)
<b>7. Consultar</b>	
retornarConsulta(int codigo) devuelve ArrayList	controlConsultaContrato
<b>8. Crear</b>	
retornarCreacion(ArrayList datos) devuelve boolean	controlCreacionContrato
<b>5. Validar Código</b>	
retornarValidacion(int codigo) devuelve boolean	controlConsultaContrato
getContrato() devuelve Contrato	
getFechas(int codigo) devuelve ArrayList	
<b>Tarjetas CRC</b>	
<b>Objeto:</b>	Pedido
<b>Descripción:</b>	Contiene toda la información pertinente a los pedidos creados en el sistema
<b>Módulo:</b>	
<b>Estereotipo:</b>	Entidad
<b>Propiedades:</b>	Concreta
<b>Superclases:</b>	Datos
<b>Subclases:</b>	
<b>Atributos:</b>	codigoPedido (int) con (Conectar) datos (ArrayList) pedido (Pedido) reg (Connection) sql (String)
<b>7. Consultar</b>	
retornarConsulta(int codigo) devuelve ArrayList	controlConsultaPedido
<b>8. Crear</b>	
retornarCreacion(ArrayList datos)	controlCreaciónPedido

devuelve boolean	
<b>5. Validar Código</b>	
retornarValidacion(int codigo) devuelve boolean	controlConsultaPedido
actualizarPedido(ArrayList datos) devuelve boolean	
consultarActualizacion(int codigo) devuelve ArrayList	
consultarTipoProducto(int codigo) devuelve int	
generarActualizacion(ArrayList datos) devuelve void	
getCodigoPedido() devuelve int	
getPedido() devuelve Pedido	
validacionActualizacion(int codigo) devuelve boolean	
<b>Tarjetas CRC</b>	
<b>Objeto:</b>	materiaPrima
<b>Descripción:</b>	Contiene toda la información pertinente a la materia prima almacenada en el sistema
<b>Módulo:</b>	
<b>Estereotipo:</b>	Entidad
<b>Propiedades:</b>	Concreta
<b>Superclases:</b>	Datos
<b>Subclases:</b>	
<b>Atributos:</b>	Con (Conectar) materiaPrima (MateriaPrima) reg (Connection) sql (String)
<b>7. Consultar</b>	
retornarConsulta(int codigo) devuelve ArrayList	controlVerificacion
<b>8. Crear</b>	
retornarCreacion(ArrayList datos) devuelve boolean	controlIngresoMateriaPrima
asignar(ArrayList matPrima) devuelve boolean	
entregar(ArrayList datosEntrega) devuelve void	
getMateriaPrima() devuelve	

MateriaPrima	
ingresar(ArrayList datos) devuelve void	
retornarConsultaNomCant(int codigo) devuelve ArrayList	
<b>Tarjetas CRC</b>	
<b>Objeto:</b>	Personal
<b>Descripción:</b>	Contiene toda la información pertinente al personal almacenado en el sistema
<b>Módulo:</b>	
<b>Estereotipo:</b>	Entidad
<b>Propiedades:</b>	Concreta
<b>Superclases:</b>	Datos
<b>Subclases:</b>	
<b>Atributos:</b>	objPersona (Personal)
<b>7. Consultar</b>	
retornarConsulta(int codigo) devuelve ArrayList	controlVerificacion
retornarConsulta(int codigo) devuelve ArrayList	controlEliminacionPersonal
retornarConsulta(int codigo) devuelve ArrayList	controlModificaciónPersonal
<b>8. Crear</b>	
boolean retornarCreacion(ArrayList datos)	controlAñadirPersonal
asignar(ArrayList personal) devuelve boolean	
eliminar(ArrayList datos) boolean	
getObjPersona() devuelve Personal	
modificacion(ArrayList datosP) devuelve boolean	
retornarConsultaNombre(int codigo) devuelve ArrayList	
<b>Tarjetas CRC</b>	
<b>Objeto:</b>	espacioPlanta
<b>Descripción:</b>	Contiene toda la información pertinente al espacio en planta almacenado en el sistema
<b>Módulo:</b>	
<b>Estereotipo:</b>	Entidad
<b>Propiedades:</b>	Concreta
<b>Superclases:</b>	Datos
<b>Subclases:</b>	espacioPlanta (EspacioPlanta)

<b>Atributos:</b>	
<b>7. Consultar</b>	
retornarConsulta(int codigo) devuelve ArrayList	controlVerificacion
retornarConsulta(int codigo) devuelve ArrayList	controlModificaciónEspacioPlanta
retornarConsulta(int codigo) devuelve ArrayList	controlEliminarEspacioPlanta
<b>8. Crear</b>	
retornarCreacion(ArrayList datos) devuelve boolean	controlañadirEspacioPlanta
asignar(ArrayList zona) devuelve boolean	
eliminar(ArrayList datosE) devuelve boolean	
getEspacioPlanta() devuelve EspacioPlanta	
modificacion (ArrayList datosM) devuelve boolean	
retornarConsultaZona(int codigo) devuelve ArrayList	
<b>Tarjetas CRC</b>	
<b>Objeto:</b>	Proceso
<b>Descripción:</b>	Contiene toda la información pertinente a los procesos almacenados en el sistema
<b>Módulo:</b>	
<b>Estereotipo:</b>	Entidad
<b>Propiedades:</b>	Concreta
<b>Superclases:</b>	Datos
<b>Subclases:</b>	
<b>Atributos:</b>	objProceso (Proceso)
consultarProcesos(int tipoProd) devuelve ArrayList	
eliminar(ArrayList datos) devuelve boolean	
getObjProceso() devuelve Proceso	
retornarTipoProducto() devuelve ArrayList	
<b>8. Crear</b>	

retornarCreacion(ArrayList datos) devuelve boolean	controlCreacionProceso
<b>7. Consultar</b>	
retornarConsulta(int codigo) devuelve ArrayList	controlEliminacionProceso
retornarConsulta(int codigo) devuelve ArrayList	controlConsultaPedido
<b>Tarjetas CRC</b>	
<b>Objeto:</b>	Cliente
<b>Descripción:</b>	Contiene la información pertinente a los clientes almacenados en el sistema
<b>Módulo:</b>	
<b>Estereotipo:</b>	Entidad
<b>Propiedades:</b>	Concreta
<b>Superclases:</b>	Datos
<b>Subclases:</b>	
<b>Atributos:</b>	Cliente (Cliente) Con (Conectar) Datos (ArrayList) Nombre (String) Reg (Connection) Sql (String)
<b>7. Consultar</b>	
retornarConsulta(int codigo) devuelve ArrayList	controlModificaciónCliente
<b>8. Crear</b>	
retornarCreacion(ArrayList datos) devuelve boolean	controlCreacionCotizacion
<b>5. Validar Código</b>	
retornarValidacion(int codigo) devuelve boolean	Cliente
getCliente() devuelve Cliente	
modificacion(ArrayList datosC) boolean	
retornarNombre(int codigo) devuelve String	
<b>Tarjetas CRC</b>	
<b>Objeto:</b>	Producto
<b>Descripción:</b>	Contiene toda la información pertinente a los procesos almacenados en el sistema
<b>Módulo:</b>	

<b>Estereotipo:</b>	Entidad
<b>Propiedades:</b>	Concreta
<b>Superclases:</b>	Datos
<b>Subclases:</b>	
<b>Atributos:</b>	objProceso (Proceso)
<b>8. Crear</b>	
retornarCreacion(ArrayList datos) devuelve boolean	controlCreacionProducto
<b>7. Consultar</b>	
retornarConsulta(int codigo) devuelve ArrayList	controlEliminacionProducto
retornarConsulta(int codigo) devuelve ArrayList	controlConsultaProducto