



## CAPÍTULO IV

### ANÁLISIS ORIENTADO A OBJETOS

#### 4.1 ESPECIFICACIÓN DE CASOS DE USO

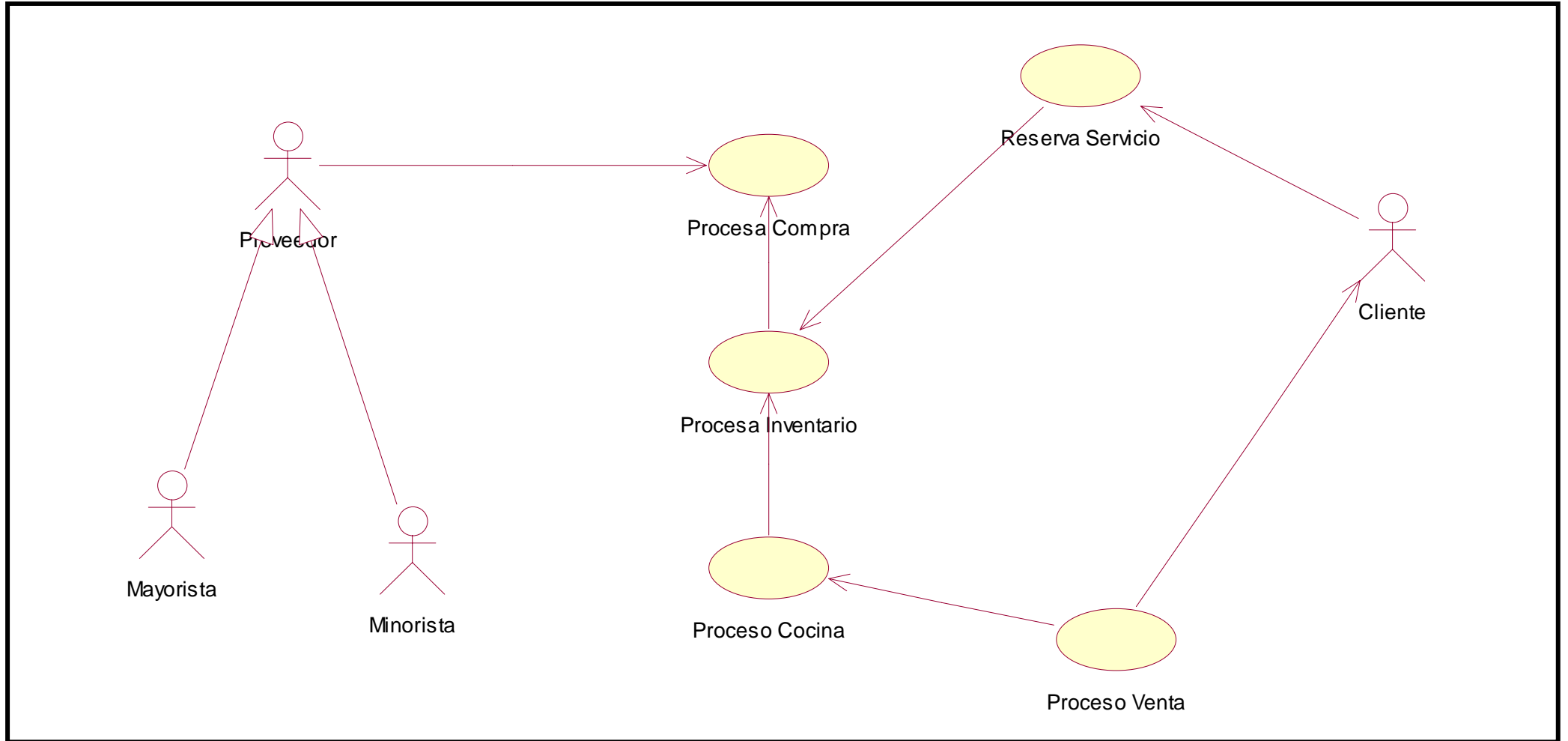
##### 4.1.1 DIAGRAMA GENERAL:

Tabla 4.1 (Caso de Uso Diagrama General)

<b>Nombre:</b>	Diagrama General
<b>Actores:</b>	Proveedor, Cliente
<b>Función:</b>	Diagrama General del Proyecto
<b>Descripción:</b>	<p>Aquí detallamos de manera macro los procesos que realiza el Hotel.</p> <p>El cliente reserva un servicio, el cual modifica el inventario; para esto se realiza una orden a cocina y se emite una factura que es entregada al cliente. Internamente los departamentos realizan pedidos a compras los cuales emite un proveedor que devuelve estas órdenes a compras con una factura.</p>



Figura No.4 Caso de Uso Diagrama General





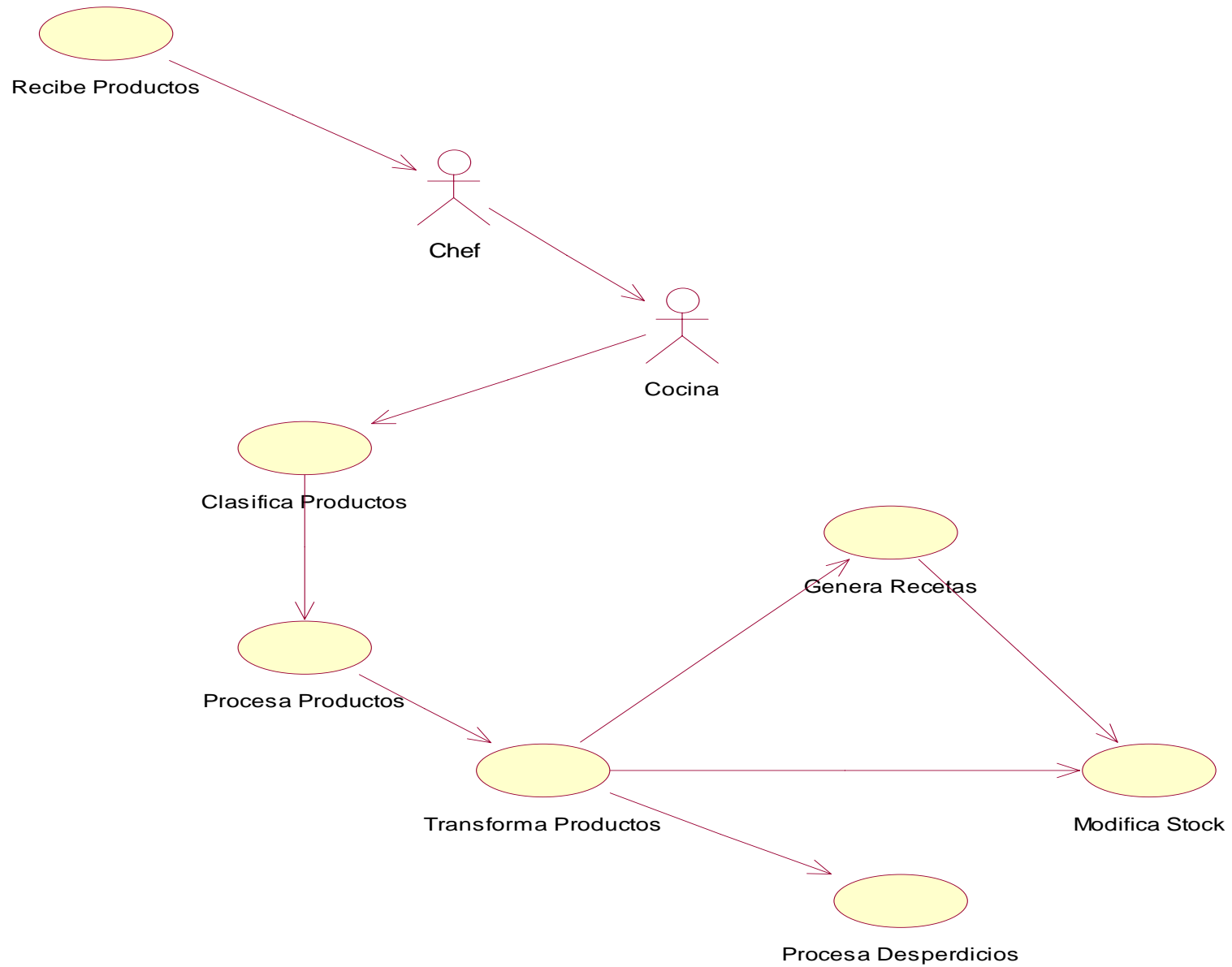
#### 4.1.2 DIAGRAMA COCINA:

Tabla 4.2 (Caso de Uso Diagrama Cocina)

<b>Nombre:</b>	Diagrama Cocina
<b>Actores:</b>	Chef, cocina
<b>Función:</b>	Control de Productos que Llegan a Cocina  Dentro de este diagrama describimos como los productos que recibe son procesados.  Inicia este proceso con la entrega de productos al Chef el cual comprueba que estén todos los productos pedidos y los envía a cocina, aquí cada producto es chequeado y verificado su estado para llegar a su clasificación.  Esta clasificación no es más que separar los vegetales, carnes rojas, carnes blancas, mariscos, etc. para llegar a la fase de procesar los productos.  Con estos productos clasificados y procesados transformamos a los mismos en subrecetas (condimentos, mayonesas, etc.) que serán enviados a generar recetas y, al mismo tiempo, procesamos los desperdicios que los mismos productos han dejado.  Se culmina este proceso modificando el stock lo cual aumenta y disminuye en inventario la cantidad de productos que se tiene.
<b>Descripción:</b>	



Figura No.5 Caso de Uso Diagrama Cocina





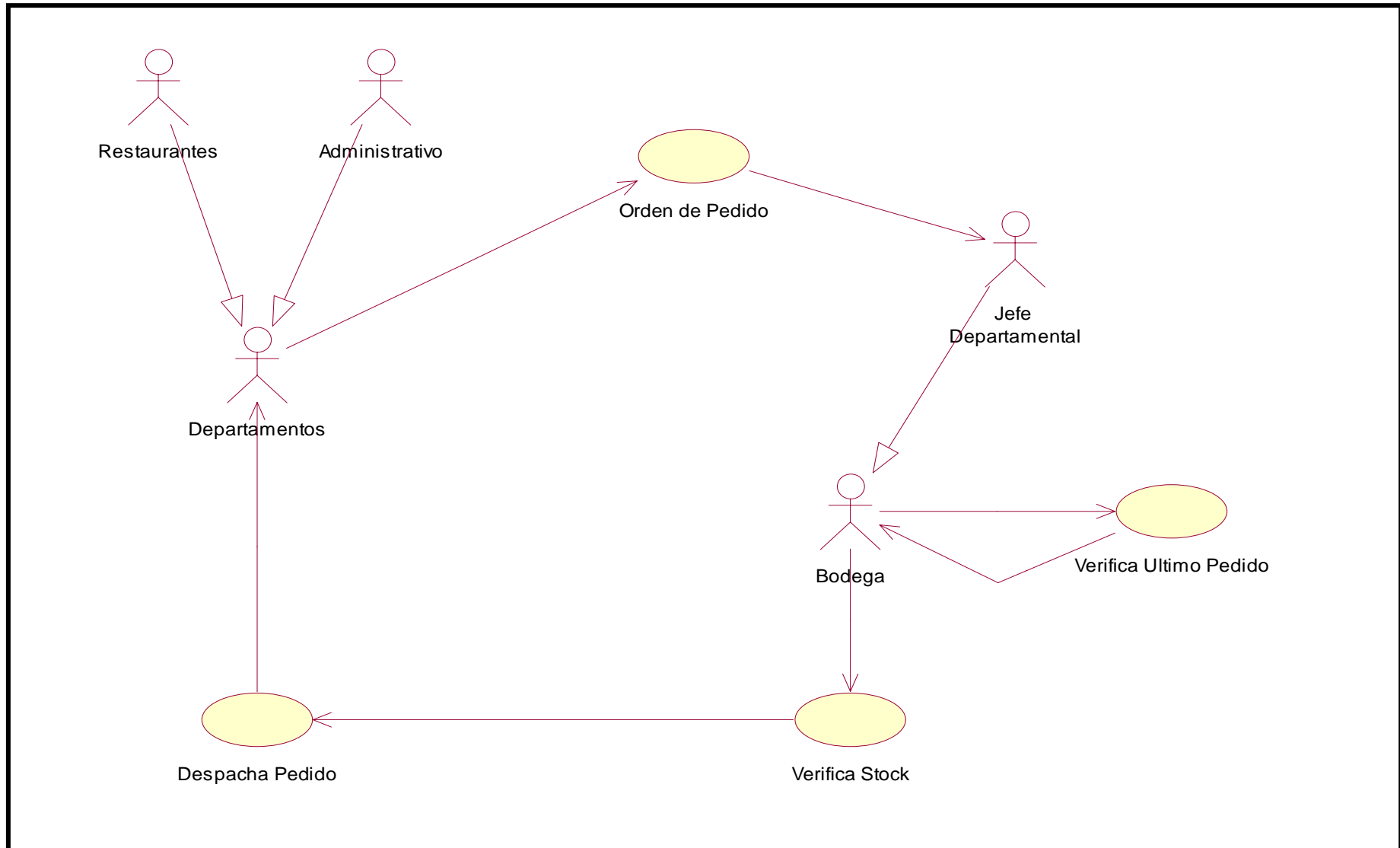
### 4.1.3 DIAGRAMA INVENTARIO

Tabla 4.3 (Caso de Uso Diagrama Inventario)

<b>Nombre:</b>	Diagrama Inventario
<b>Actores:</b>	Restaurantes, Administrativo, Departamentos, Jefe departamental, Bodega.
<b>Función:</b>	Control de Inventarios y pedidos
<b>Descripción:</b>	<p>Los diversos departamentos que posee el hotel como por ejemplo restaurante, costos, ama de llaves, etc., realizan una orden de pedido en el cual informan las necesidades de cada uno, esta orden es entregada al jefe departamental para ser aprobada, la bodega despacha todos los pedidos que fueron aprobados; esto se realiza mediante un control de pedidos que el personal de bodega posee.</p> <p>Las órdenes no aprobadas regresan al departamento hasta que sea aceptada.</p>



Figura No.6 Caso de Uso Diagrama Inventario





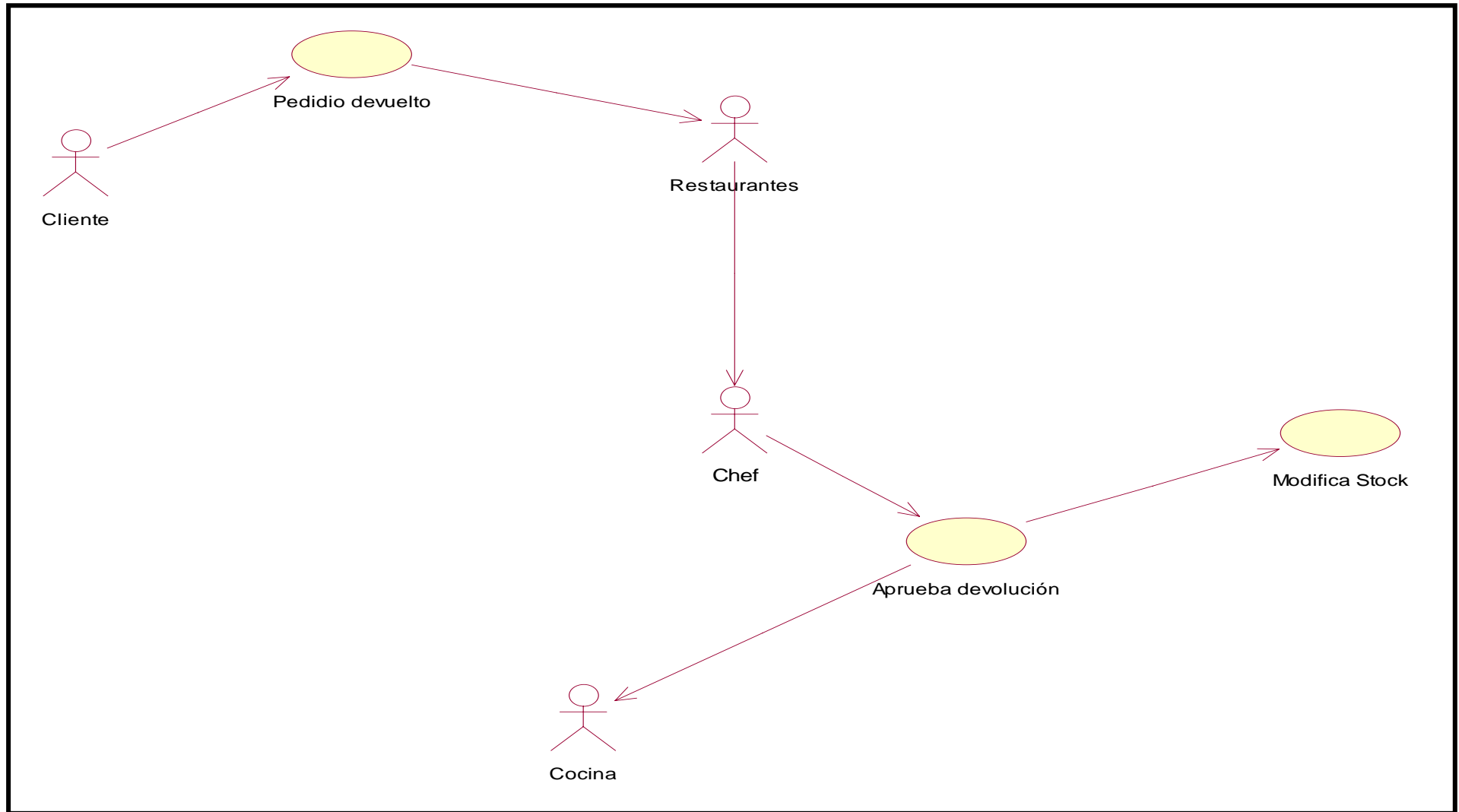
#### 4.1.4 DIAGRAMA DEVOLUCION:

Tabla 4.4 (Caso de Uso Diagrama Devolución)

<b>Nombre:</b>	Diagrama Devolución
<b>Actores:</b>	Cliente, Restaurantes, Chef y Cocina
<b>Función:</b>	Control de devolución de pedidos de los restaurantes
<b>Descripción:</b>	Cuando un cliente realiza un pedido, en el caso de que ese sea en uno de los restaurantes y el cliente no se sienta conforme con el plato que recibió, este puede realizar la devolución de dicho pedido el cual es recibido en el restaurante; éste envía el pedido al chef el cual verifica el por que fue devuelto para llegar a la aprobación del pedido y enviar este plato a cocina y modificar el stock.



Figura No 7 Caso de Uso Diagrama Devolución







#### 4.1.5 DIAGRAMA VENTAS:

Tabla 4.5 (Caso de Uso Diagrama Ventas)

**Nombre:** Diagrama Ventas

**Actores:** Restaurantes, Cliente, Habitaciones, chef, Cocina

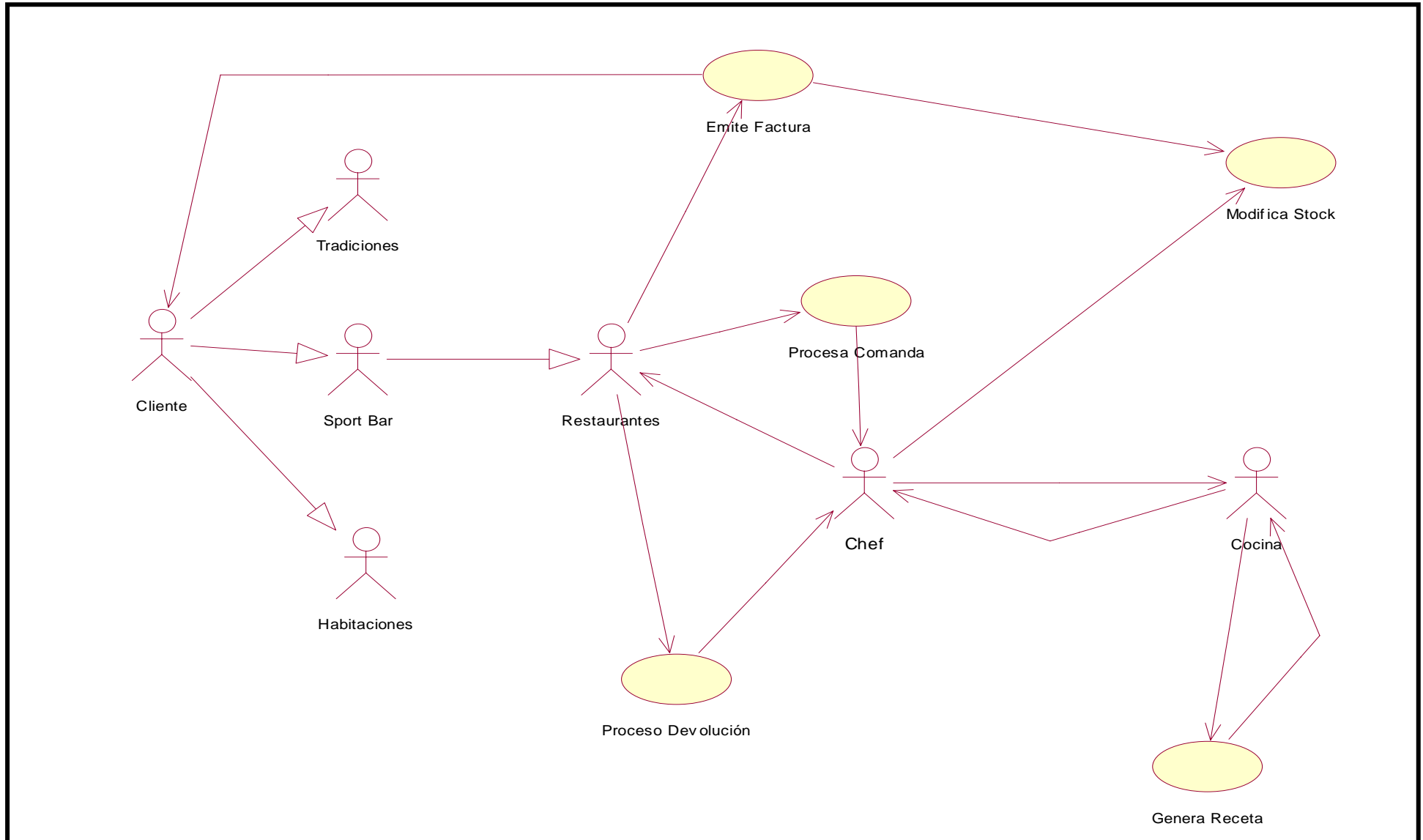
**Función:** Control de Ventas

**Descripción:** El cliente puede realizar una compra o pedido en cualquiera de sus restaurantes o desde su habitación, este pedido es recibido por nuestros restaurantes, los cuales emiten una comanda que no es más que la especificación del plato que el cliente desea consumir. Esta comanda pasa por el chef de turno el cual envía a cocina para su elaboración, éste genera la receta y devuelve al chef para su aprobación, el chef aprueba el plato y envía al cliente; una vez terminado el servicio se emite la factura y modificamos stock.

Si el cliente no se sintió satisfecho con el plato se realiza el proceso de devolución que es controlado por el chef; aprobado este proceso se emite factura para poder modificar stock ya que los ingredientes del plato ya se utilizaron y con esto se descargan en inventario.



Figura No 8 Caso de Uso Diagrama Ventas





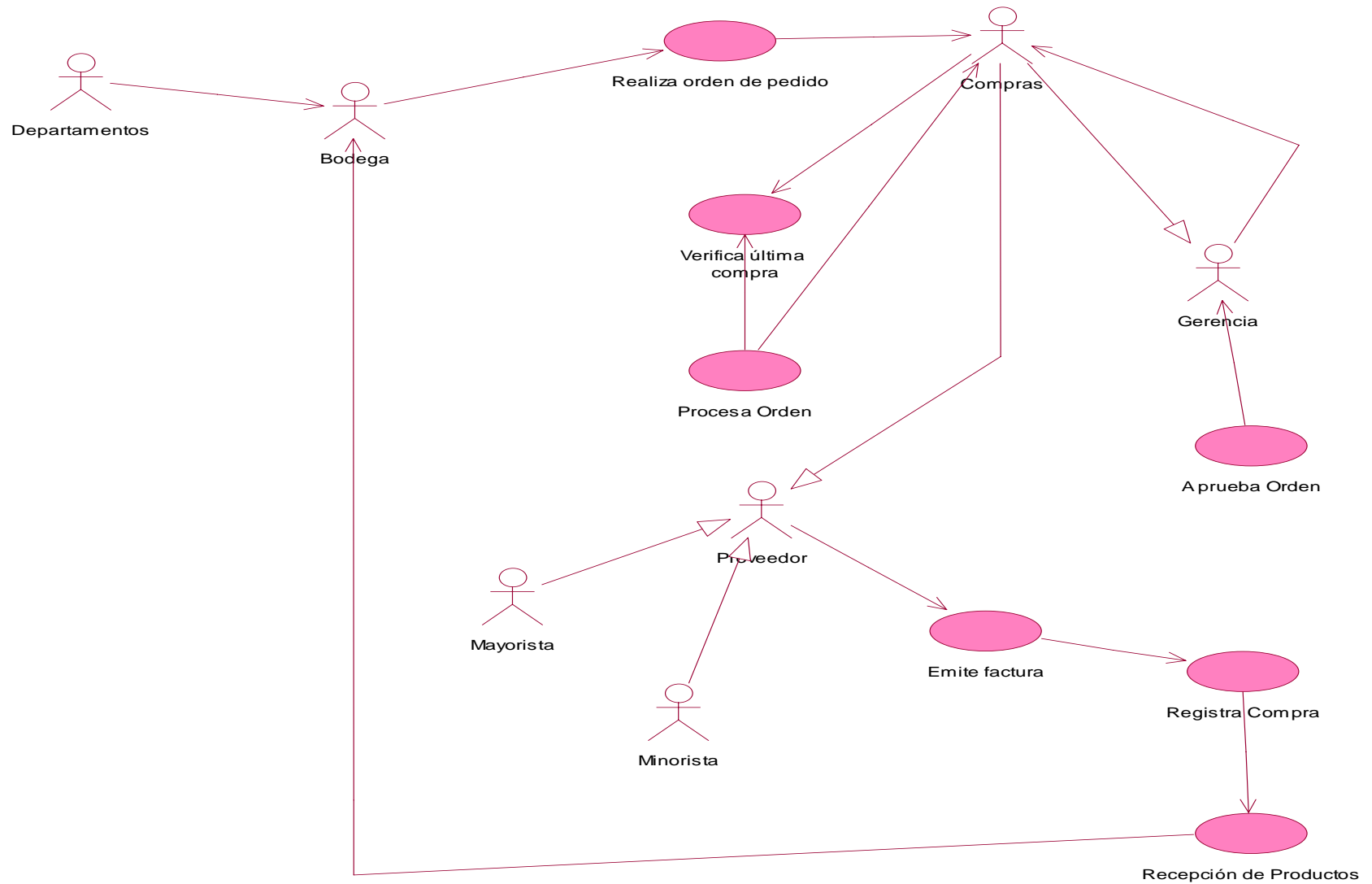
#### 4.1.6 DIAGRAMA COMPRAS:

Tabla 4.6 (Caso de Uso Diagrama Compras)

<b>Nombre:</b>	Diagrama Compras
<b>Actores:</b>	Departamentos, Bodega, Compras, Gerencia, Proveedor
<b>Función:</b>	Control de Compras
<b>Descripción:</b>	Los diversos departamentos realizan un pedido que es recibido por bodega; esta orden de compra es enviada a compras para que pueda verificar cual fue su última compra; esto se procesa y se envía todas las ordenes de compras a la gerencia para que de su aprobación. Al ser aprobadas, compras envía las órdenes a los proveedores, los cuales emiten la factura, inmediatamente se registra la compra, se reciben los productos y se entregan las ordenes correspondientes a cada departamento.



Figura No 9 Caso de Uso Diagrama Compras





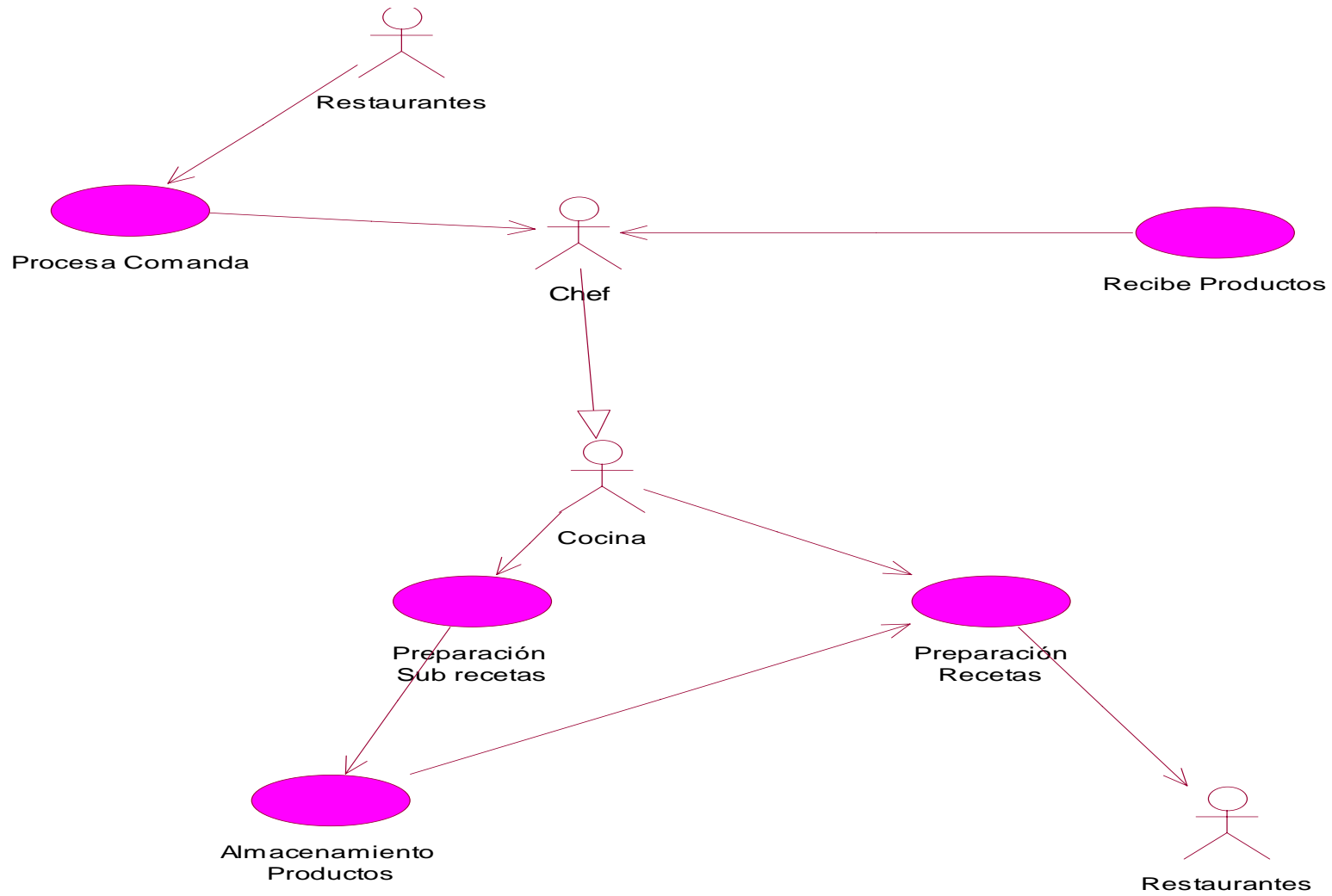
#### 4.1.7 DIAGRAMA GENERA RECETA:

Tabla 4.7 (Caso de Uso Diagrama Genera Receta)

<b>Nombre:</b>	Diagrama Genera Receta
<b>Actores:</b>	Restaurantes, Chef, Cocina
<b>Función:</b>	Control de la Generación de recetas
<b>Descripción:</b>	<p>El restaurante emite una comanda la cual es la especificación del plato; ésta es enviada al chef, el cual envía a cocina para la elaboración la receta (comanda), aquí involucra subrecetas si fuese el caso (ceviches, salsas, etc.) Estas subrecetas se almacenan y serán utilizadas en la preparación de posteriores recetas.</p> <p>Una vez preparada la receta se envía al restaurante.</p>



Figura No 10 Caso de Uso Genera Recetas





#### 4.1.8 DIAGRAMA RESERVA SALONES:

Tabla 4.8 (Caso de Uso Diagrama Reserva Salones)

**Nombre:** Diagrama Reserva Salones

**Actores:** Cliente, Coordinadora, Chef, Cocina, Steward

**Función:** Control y Reserva de salones

**Descripción:** El cliente puede reservar su salón vía Web o en las instalaciones del hotel, inicia revisando cada salón que posee el hotel.

Realiza el sub total de productos y montajes para la reservación del salón vía online; esto sirve para la actualización del cronograma que la coordinadora posee.

Después que se confirma la reserva, se envía el tipo de montaje a Steward, que recibe y devuelve el montaje a la coordinadora para la realización del proceso venta.

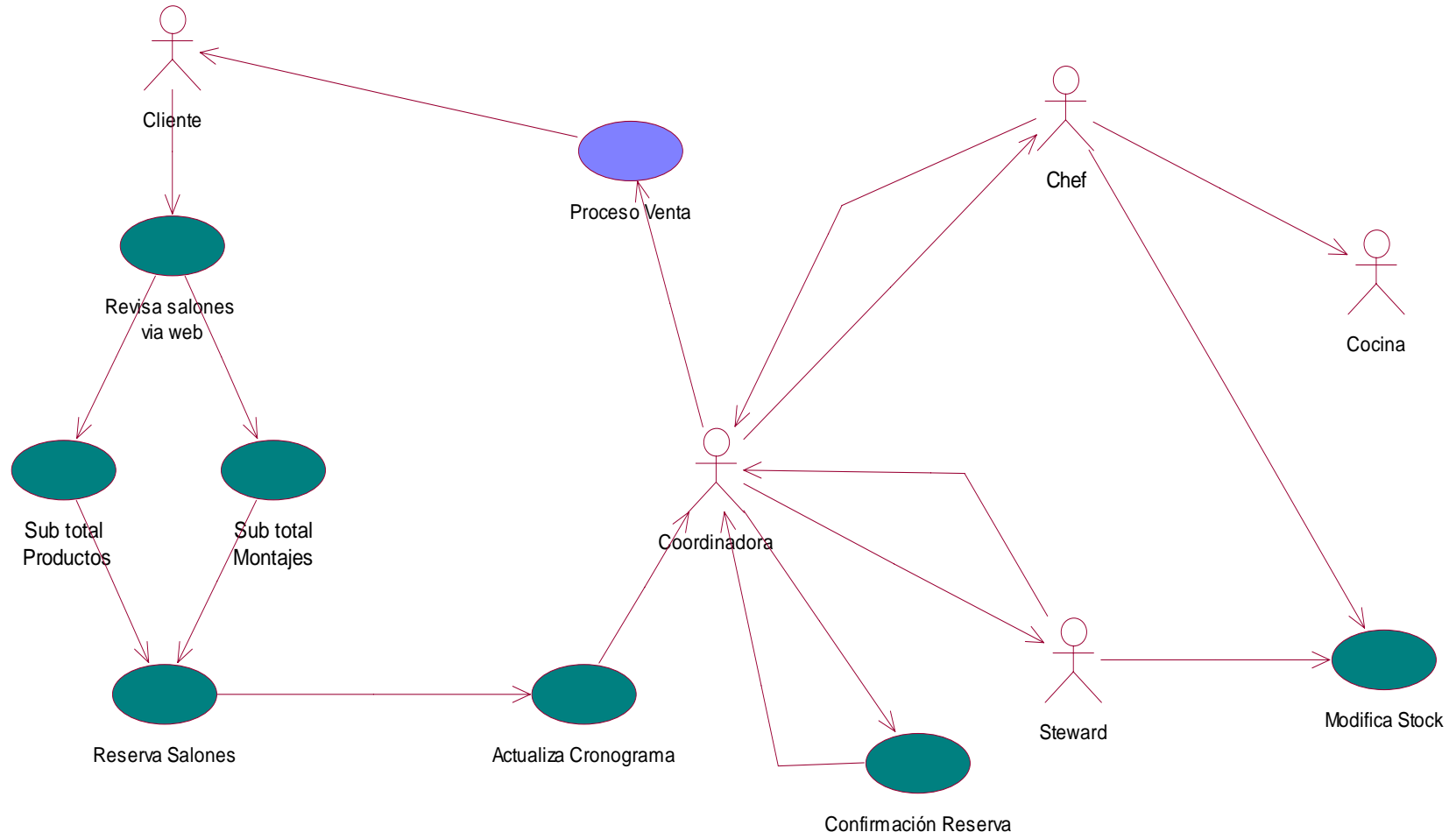
El tipo de menú se envía al chef, éste envía a cocina para su elaboración y envían el menú a la coordinadora para poder realizar la venta.

Tanto Steward como el chef modifican inventario.

El cliente recibe la factura por el servicio que le brindó el Hotel.



Figura No 11 Caso de Uso Diagrama Reserva Salones







## **4.2 MODELAMIENTO DE OBJETOS**

### **4.2.1 IDENTIFICACIÓN DE CLASES DE OBJETOS**



## 4.2.2 DICCIONARIO DE DATOS

```
/*=====*/
/* Database name: PHYSICALDATAMODEL_ */
/* DBMS name: Microsoft SQL Server 2000 */
/* Created on: 08/05/2006 16:21:04 */
/*=====*/
```

```
/*=====*/
/* Table: SYS_AREAS */
/*=====*/
```

```
create table dbo.SYS_AREAS (
    codigo          nvarchar(4)    not null,
    descripcion     nvarchar(30)   null,
    Responsable     nvarchar(30)   null,
    id              nvarchar(15)   null,
    password        nvarchar(15)   null,
    constraint PK_SYS_AREAS primary key clustered (codigo)
)
```

```
/*=====*/
/* Table: SYS_BODEGAS */
/*=====*/
```

```
create table dbo.SYS_BODEGAS (
    codigo          nvarchar(4)    not null,
    nombre          nvarchar(4)    not null,
    responsable     nvarchar(40)   null,
    descripcion     nvarchar(150)  null,
    constraint PK_SYS_BODEGAS primary key clustered (nombre)
)
```

```
/*=====*/
/* Table: SYS_BOTON */
/*=====*/
```

```
create table dbo.SYS_BOTON (
    codigo          nvarchar(5)    not null,
    nombre          nvarchar(15)   not null,
    constraint PK_SYS_BOTON primary key clustered (nombre)
)
```

```
/*=====*/
/* Table: SYS_CAB_CC */
/*=====*/
```

```
create table dbo.SYS_CAB_CC (
    numero          nvarchar(15)   not null,
    cod_cliente     nvarchar(13)   null,
    bodega          nvarchar(15)   null,
    despachado      nvarchar(30)   null,
    fecha_cc        datetime       null,
    pagado          char(1)        null,
    fecha_pago      datetime       null,
    entregado       char(1)        null,
    fecha_entrega   datetime       null,
    entregado_a     nvarchar(30)   null,
    codigo_real     int             not null,
    total           nvarchar(15)   null,
    constraint PK_SYS_CAB_CC primary key clustered (codigo_real)
)
```



```
/*=====*/
/* Table: SYS_CAB_COM */
/*=====*/

create table dbo.SYS_CAB_COM (
    numero          nvarchar(15)    not null,
    cod_depart      nvarchar(4)     null,
    pedido_por      nvarchar(30)    null,
    fecha_pedido    datetime        null,
    aprobado        char(1)         null,
    fecha_aprobacion datetime       null,
    entregado       char(1)         null,
    fecha_entrega   datetime        null,
    entregado_a     nvarchar(30)    null,
    codigo_real     int              not null,
    total           nvarchar(15)    null,
    cod_proveedor  nvarchar(10)    null,
    detalle        nvarchar(500)   null,
    constraint PK_SYS_CAB_COM primary key clustered (codigo_real)
)

/*=====*/
/* Table: SYS_CAB_PED */
/*=====*/

create table dbo.SYS_CAB_PED (
    numero          nvarchar(15)    not null,
    cod_depart      nvarchar(4)     null,
    pedido_por      nvarchar(30)    null,
    fecha_pedido    datetime        null,
    aprobado        char(1)         null,
    fecha_aprobacion datetime       null,
    entregado       char(1)         null,
    fecha_entrega   datetime        null,
    entregado_a     nvarchar(30)    null,
    codigo_real     int              not null,
    constraint PK_SYS_CAB_PED primary key clustered (codigo_real)
)

/*=====*/
/* Table: SYS_CLASES */
/*=====*/

create table dbo.SYS_CLASES (
    codigo_clase    nvarchar(3)     not null,
    descripcion     nvarchar(30)    null,
    constraint PK_SYS_CLASES primary key clustered (codigo_clase)
)

/*=====*/
/* Table: SYS_CONTEO */
/*=====*/

create table dbo.SYS_CONTEO (
    codigo_cont     char(4)         not null,
    Descripcion     nvarchar(20)    null,
    intervalo       char(5)         null,
    fecha_inventario datetime      null,
    numero          int              null,
    constraint PK_SYS_CONTEO primary key clustered (codigo_cont)
)
```



```
/*=====*/
/* Table: SYS_CLIENTES */
/*=====*/

create table dbo.SYS_CLIENTES (
    Ruc                nvarchar(13)    not null,
    Nombre             nvarchar(50)    null,
    direccion1         nvarchar(50)    null,
    direccion2         nvarchar(50)    null,
    telefono1          nvarchar(10)    null,
    telefono2          nvarchar(10)    null,
    telefono3          nvarchar(10)    null,
    contacto           nvarchar(30)    null,
    dias_pago          nvarchar(3)     null,
    email              nvarchar(50)    null,
    ciudad             nvarchar(15)    null,
    observacion        nvarchar(100)   null,
    saldo              float(15)        null,
    constraint PK_SYS_CLIENTES primary key clustered (Ruc)
)

/*=====*/
/* Table: SYS_DET_CC */
/*=====*/

create table dbo.SYS_DET_CC (
    ruc                nvarchar(13)    not null,
    nombre             nvarchar(50)    null,
    chk                int             not null,
    total              nvarchar(15)    null,
    fecha_compra       datetime        null,
    num_cc             int             not null,
    fecha_cc           datetime        null,
    constraint PK_SYS_DET_CC primary key clustered (ruc, chk, num_cc)
)

/*=====*/
/* Table: SYS_DET_COM */
/*=====*/

create table dbo.SYS_DET_COM (
    codigo             nvarchar(8)     not null,
    numero_cab         int             not null,
    codigo_item        nvarchar(8)     not null,
    cantidad_pedida    float(15)       null,
    cantidad_entregada float(15)       null,
    saldo_stock        float(15)       null,
    nombre_articulo    nvarchar(30)    null,
    precio             nvarchar(50)    null,
    constraint PK_SYS_DET_COM primary key clustered (codigo, numero_cab, codigo_item)
)

/*=====*/
/* Table: SYS_DET_PED */
/*=====*/

create table dbo.SYS_DET_PED (
    codigo             nvarchar(8)     not null,
    numero_cab         int             not null,
    codigo_item        nvarchar(8)     not null,
    cantidad_pedida    float(15)       null,
    cantidad_entregada float(15)       null,
    saldo_stock        float(15)       null,
    nombre_articulo    nvarchar(30)    null,
    constraint PK_SYS_DET_PED primary key clustered (codigo, numero_cab, codigo_item)
)
```



```
/*=====*/
/* Table: SYS_EVENTOS_WEB */
/*=====*/
```

```
create table dbo.SYS_EVENTOS_WEB (
    codigo          nvarchar(15)    not null,
    nombre          nvarchar(50)    null,
    apellidos      nvarchar(50)    null,
    ciudad         nvarchar(15)    null,
    pais           nvarchar(15)    null,
    ci              nvarchar(14)    null,
    telefono       nvarchar(20)    null,
    email          nvarchar(50)    null,
    huesped       nvarchar(50)    null,
    horario        nvarchar(50)    null,
    fecha_inicio  datetime         null,
    fecha_final   datetime         null,
    tipo_evento   int              null,
    tipo_menu     int              null,
    adicionales  nvarchar(1500)   null,
    pax          int              null,
    conversado   nvarchar(2)      null,
    desc_conver  nvarchar(300)     null,
    reservado    nvarchar(1)      null,
    terminado    nvarchar(50)     null,
    desc_termi   nvarchar(300)     null,
    division     nvarchar(2)      null,
    codigo_real  nvarchar(15)     null,
    constraint PK_SYS_EVENTOS_WEB primary key clustered (codigo)
)
```

```
/*=====*/
/* Table: SYS_FISICO */
/*=====*/
```

```
create table dbo.SYS_FISICO (
    codigo_prod    nvarchar(8)    not null,
    descripcion    nvarchar(50)   not null,
    tipo           char(4)        not null,
    unidad         nvarchar(50)   not null,
    stock_inicial float(15)       null,
    Stock         float(15)       not null,
    Costo_inicial money           null,
    Costo_promedio money         null,
    costo_compra  money           null,
    costo         money           null,
    codigo_clase  nvarchar(3)     not null,
    unidad_compra char(3)        null,
    tipo_impuesto char(3)        null,
    CODIGO_PROVEEDOR nvarchar(10) null,
    maximos      int             null,
    minimos      int             null,
    codigo_bodega nvarchar(4)    null,
    constraint PK_SYS_FISICO primary key clustered (codigo_prod)
)
```

```
/*=====*/
/* Table: SYS_FIS_STOCK_DIA_1 */
/*=====*/
```

```
create table dbo.SYS_FIS_STOCK_DIA_1 (
    codigo_prod    nvarchar(8)    not null,
    descripcion    nvarchar(50)   not null,
    tipo           char(4)        not null,
    unidad         nvarchar(50)   not null,
    stock_inicial  int             null,
```



```
        Stock          nvarchar(8)      null,
        codigo_clase   nvarchar(3)     not null,
        CODIGO_PROVEEDOR nvarchar(10)    null,
        constraint PK_SYS_FIS_STOCK_DIA_1 primary key clustered (codigo_prod)
    )

/*=====*/
/* Table: SYS_FIS_STOCK_MEN_1 */
/*=====*/

create table dbo.SYS_FIS_STOCK_MEN_1 (
        codigo_prod     nvarchar(8)      not null,
        descripcion     nvarchar(50)     not null,
        tipo            char(4)         not null,
        unidad          nvarchar(50)     not null,
        stock_inicial   int              null,
        Stock           nvarchar(8)      null,
        codigo_clase    nvarchar(3)     not null,
        CODIGO_PROVEEDOR nvarchar(10)    null,
        constraint PK_SYS_FIS_STOCK_MEN_1 primary key clustered (codigo_prod)
    )

/*=====*/
/* Table: SYS_FIS_STOCK QUI_1 */
/*=====*/

create table dbo.SYS_FIS_STOCK QUI_1 (
        codigo_prod     nvarchar(8)      not null,
        descripcion     nvarchar(50)     not null,
        tipo            char(4)         not null,
        unidad          nvarchar(50)     not null,
        stock_inicial   int              null,
        Stock           nvarchar(8)      null,
        codigo_clase    nvarchar(3)     not null,
        CODIGO_PROVEEDOR nvarchar(10)    null,
        constraint PK_SYS_FIS_STOCK QUI_1 primary key clustered (codigo_prod)
    )

/*=====*/
/* Table: SYS_FIS_STOCK_SEM_1 */
/*=====*/

create table dbo.SYS_FIS_STOCK_SEM_1 (
        codigo_prod     nvarchar(8)      not null,
        descripcion     nvarchar(50)     not null,
        tipo            char(4)         not null,
        unidad          nvarchar(50)     not null,
        stock_inicial   int              null,
        Stock           nvarchar(8)      null,
        codigo_clase    nvarchar(3)     not null,
        CODIGO_PROVEEDOR nvarchar(10)    null,
        constraint PK_SYS_FIS_STOCK_SEM_1 primary key clustered (codigo_prod)
    )

/*=====*/
/* Table: SYS_PERFIL */
/*=====*/

create table dbo.SYS_PERFIL (
        codigo          nvarchar(5)      not null,
        descripcion     nvarchar(50)     null,
        accesos         nvarchar(100)    null,
        constraint PK_SYS_PERFIL primary key clustered (codigo)
    )
```



```
/*=====*/
/* Table: SYS_POS_CAB */
/*=====*/
create table dbo.SYS_POS_CAB (
    chk                int                not null,
    fecha              datetime           null,
    nombre             nvarchar(50)      null,
    cedula             nvarchar(10)      null,
    mesa              int                null,
    subtotal           nvarchar(15)      null,
    iva                nvarchar(15)      null,
    servicio           nvarchar(15)      null,
    total             nvarchar(15)      null,
    numero_productos  int                null,
    forma_pago        nvarchar(15)      null,
    vuelto            nvarchar(15)      null,
    pago_cliente      nvarchar(15)      null,
    nombre_tarjeta    nvarchar(50)      null,
    numero_tarjeta    nvarchar(14)      null,
    numero_aprobacion nvarchar(15)      null,
    stock             nvarchar(2)        null,
    fecha_stock       datetime           null,
    constraint PK_SYS_POS_CAB primary key clustered (chk)
)

/*=====*/
/* Table: SYS_POS_DET */
/*=====*/
create table dbo.SYS_POS_DET (
    COD                nvarchar(8)       not null,
    CHK                int                not null,
    NOMBRE             nvarchar(50)      not null,
    PRECIO             nvarchar(50)      null,
    CANTIDAD           int                null,
    constraint PK_SYS_POS_DET primary key clustered (COD, CHK, NOMBRE)
)

/*=====*/
/* Table: SYS_PROVEEDOR */
/*=====*/
create table dbo.SYS_PROVEEDOR (
    Codigo            nvarchar(10)       not null,
    Nombre            nvarchar(40)       not null,
    Direccion         nvarchar(50)       not null,
    Telefono          nvarchar(15)       null,
    Fax               nvarchar(15)       null,
    Celular           nvarchar(15)       null,
    Email             nvarchar(30)       null,
    Dias_pago         decimal(2)         null,
    Ciudad            nvarchar(15)       null,
    Pais              nvarchar(15)       null,
    Ruc               nvarchar(13)       null,
    Nu_ref            char(20)            null,
    Fpago             char(3)             null,
    Comment           nvarchar(50)       null,
    Contacto          nvarchar(30)       null,
    constraint PK_SYS_PROVEEDOR primary key clustered (Codigo)
)
```



```
/*=====*/
/* Table: SYS_RECETA */
/*=====*/

create table dbo.SYS_RECETA (
    codigo          nvarchar(8)      not null,
    nombre          nvarchar(50)     not null,
    descripcion     nvarchar(50)     null,
    producto_1     nvarchar(8)       null,
    nombre_1       nvarchar(50)      null,
    cantidad_1     nvarchar(53)      null,
    unidad_1       char(10)          null,
    costo_1        nvarchar(10)      null,
    producto_2     nvarchar(8)       null,
    nombre_2       nvarchar(50)      null,
    cantidad_2     nvarchar(53)      null,
    unidad_2       char(10)          null,
    costo_2        nvarchar(10)      null,
    producto_3     nvarchar(8)       null,
    nombre_3       nvarchar(50)      null,
    cantidad_3     nvarchar(53)      null,
    unidad_3       char(10)          null,
    costo_3        nvarchar(10)      null,
    producto_4     nvarchar(8)       null,
    nombre_4       nvarchar(50)      null,
    cantidad_4     nvarchar(53)      null,
    unidad_4       char(10)          null,
    costo_4        nvarchar(10)      null,
    producto_5     nvarchar(8)       null,
    nombre_5       nvarchar(50)      null,
    cantidad_5     nvarchar(53)      null,
    unidad_5       char(10)          null,
    costo_5        nvarchar(10)      null,
    ruta           nvarchar(50)      null,
    boton          nvarchar(15)      null,
    costo_sistema  nvarchar(15)      null,
    costo_usuario  nvarchar(15)      null,
    precio_final   money             null,
    constraint PK_SYS_RECETA primary key clustered (codigo)
)

/*=====*/
/* Table: SYS_SECUENCIAL */
/*=====*/

create table dbo.SYS_SECUENCIAL (
    tabla          nvarchar(15)      not null,
    numero         int               not null,
    descripcion    nvarchar(50)      null,
    constraint PK_SYS_SECUENCIAL primary key clustered (tabla)
)

/*=====*/
/* Table: SYS_STOCK_TEMP */
/*=====*/

create table dbo.SYS_STOCK_TEMP (

    codigo         nvarchar(10)     not null,
    codigo_item    nvarchar(10)     null,
    cantidad_vend  int               null,
    fecha         datetime           null,
    chk           int               not null,
    receta        nvarchar(50)      null,
    constraint PK_SYS_STOCK_TEMP primary key clustered (codigo, chk)
)
```





```
/*=====*/
/* Table: SYS_SUBRECETA */
/*=====*/
create table dbo.SYS_SUBRECETA (
    codigo                nvarchar(8)        not null,
    nombre                nvarchar(50)       null,
    descripcion           nvarchar(50)       null,
    unidad                nvarchar(3)        null,
    costo                 nvarchar(10)       null,
    producto_1            nvarchar(8)        null,
    nombre_1              nvarchar(50)       null,
    cantidad_1            nvarchar(53)       null,
    unidad_1              nvarchar(3)        null,
    costo_1               nvarchar(50)       null,
    producto_2            nvarchar(8)        null,
    nombre_2              nvarchar(50)       null,
    cantidad_2            nvarchar(53)       null,
    unidad_2              nvarchar(3)        null,
    costo_2               nvarchar(10)       null,
    producto_3            nvarchar(8)        null,
    nombre_3              nvarchar(50)       null,
    cantidad_3            nvarchar(53)       null,
    unidad_3              nvarchar(3)        null,
    costo_3               nvarchar(10)       null,
    producto_4            nvarchar(8)        null,
    nombre_4              nvarchar(50)       null,
    cantidad_4            nvarchar(53)       null,
    unidad_4              nvarchar(3)        null,
    costo_4               nvarchar(10)       null,
    producto_5            nvarchar(8)        null,
    nombre_5              nvarchar(50)       null,
    cantidad_5            nvarchar(53)       null,
    unidad_5              nvarchar(3)        null,
    costo_5               nvarchar(10)       null,
    producto_6            nvarchar(8)        null,
    nombre_6              nvarchar(50)       null,
    cantidad_6            nvarchar(53)       null,
    unidad_6              nvarchar(3)        null,
    costo_6               nvarchar(10)       null,
    producto_7            nvarchar(8)        null,
    nombre_7              nvarchar(50)       null,
    cantidad_7            nvarchar(53)       null,
    unidad_7              nvarchar(3)        null,
    costo_7               nvarchar(10)       null,
    producto_8            nvarchar(8)        null,
    nombre_8              nvarchar(50)       null,
    cantidad_8            nvarchar(53)       null,
    unidad_8              nvarchar(3)        null,
    costo_8               nvarchar(10)       null,
    producto_9            nvarchar(8)        null,
    nombre_9              nvarchar(50)       null,
    cantidad_9            nvarchar(53)       null,
    unidad_9              nvarchar(3)        null,
    costo_9               nvarchar(10)       null,
    producto_10           nvarchar(8)        null,
    nombre_10             nvarchar(50)       null,
    cantidad_10           nvarchar(53)       null,
    unidad_10             nvarchar(3)        null,
    costo_10              nvarchar(10)       null,
    constraint PK_SYS_SUBRECETA primary key clustered (codigo)
)
```



```
/*=====*/
/* Table: SYS_SYS */
/*=====*/
create table dbo.SYS_SYS (
    ID                nvarchar(8)        not null,
    Servidor          nvarchar(30)       null,
    Version           nvarchar(15)       null,
    Autor             nvarchar(30)       null,
    Empresa           nvarchar(30)       null,
    Ruc               nvarchar(13)       null,
    Direccion         nvarchar(50)       null,
    Telefono          nvarchar(15)       null,
    Celular           nvarchar(15)       null,
    Representante     nvarchar(50)       null,
    constraint PK_SYS_SYS primary key clustered (ID)
)

/*=====*/
/* Table: SYS_TABLAS_INVENTARIO */
/*=====*/
create table dbo.SYS_TABLAS_INVENTARIO (
    cod               nvarchar(4)        not null,
    numero            int                not null,
    nombre_tabla     nvarchar(30)       null,
    fecha_tabla      datetime           null,
    constraint PK_SYS_TABLAS_INVENTARIO primary key clustered (cod)
)

/*=====*/
/* Table: SYS_TIP_EVENTOS */
/*=====*/
create table dbo.SYS_TIP_EVENTOS (
    codigo            int                not null,
    tipo_evento      nvarchar(30)       null,
    descripcion      nvarchar(30)       null,
    ruta             nvarchar(150)      null,
    constraint PK_SYS_TIP_EVENTOS primary key clustered (codigo)
)

/*=====*/
/* Table: SYS_TIP_MENUS */
/*=====*/
create table dbo.SYS_TIP_MENUS (
    codigo            int                not null,
    nombre           nvarchar(15)       null,
    entrada          nvarchar(30)       null,
    fuerte           nvarchar(30)       null,
    postre           nvarchar(30)       null,
    costo            money              null,
    descripcion      nvarchar(50)       null,
    ruta             nvarchar(500)      null,
    constraint PK_SYS_TIP_MENUS primary key clustered (codigo)
)

/*=====*/
/* Table: SYS_USER */
/*=====*/
create table dbo.SYS_USER (
    id                nvarchar(15)      not null,
    password          nvarchar(15)      null,
    fecha_creacion    datetime          null,
    accesos           nvarchar(5)       null,
    constraint PK_SYS_USER primary key clustered (id)
)
```



```
/*=====*/
/* Table: TEMP_DET_COM */
/*=====*/
create table dbo.TEMP_DET_COM (
    codigo                int                not null,
    numero_cab            int                not null,
    codigo_item           nvarchar(8)       null,
    cantidad_pedida       float(15)         null,
    cantidad_entregada    float(15)         null,
    saldo_stock           float(15)         null,
    nombre_articulo       nvarchar(50)      null,
    precio                nvarchar(53)     null,
    constraint PK_TEMP_DET_COM primary key clustered (codigo, numero_cab)
)

/*=====*/
/* Table: TEMP_DET_PED */
/*=====*/
create table dbo.TEMP_DET_PED (
    codigo                nvarchar(8)       not null,
    numero_cab            int                not null,
    codigo_item           nvarchar(8)       null,
    cantidad_pedida       float(15)         null,
    cantidad_entregada    float(15)         null,
    saldo_stock           float(15)         null,
    nombre_articulo       nvarchar(50)      null,
    constraint PK_TEMP_DET_PED primary key clustered (codigo, numero_cab)
)

/*=====*/
/* PROCEDURES / EVENTOS */
/*=====*/

CREATE PROCEDURE sp_eventos @cat int
as
    select codigo, tipo_evento, descripcion,ruta from sys_tip_eventos
    where codigo = @cat

CREATE PROCEDURE sp_eventos_select
as
    select max(codigo) as codigo from sys_eventos_web

CREATE PROCEDURE sp_eventos_web1
(
    @cod varchar(15),
    @nombre varchar(50),
    @apellido varchar(50),
    @ciudad varchar(15),
    @pais varchar(15),
    @cedu varchar(14),
    @telefono varchar(20),
    @email varchar(20),
    @huesped varchar(50),
    @horario varchar(50),
    @adicional varchar(1500),
    @menu int,
    @evento int,
    @fecha_final datetime,
    @fecha_inicio datetime,
    @pax int,
    @conversado varchar(1),
    @reservado varchar(1),
    @terminado varchar(1)

```



```

)
AS
insert into sys_eventos_web (codigo,nombre,apellidos,ciudad, pais,ci,telefono,
email,huesped, horario,adicionales,tipo_evento,tipo_menu,fecha_inicio, fecha_final,
pax,conversado,reservado,terminado)
values
(@cod,@nombre,@apellido,@ciudad,@pais,@cedu,@telefono,@email,@huesped,
@horario,@adicional,@evento,@menu,@fecha_inicio,@fecha_final,@pax,
@conversado,@reservado,@terminado)

```

```

CREATE PROCEDURE sp_menus @cat int
as
select codigo, nombre, entrada,fuerte,postre,costo,descripcion,ruta from sys_tip_menus
where codigo = @cat

```

### 4.2.3 IDENTIFICACIÓN DE ASOCIACIONES Y ATRIBUTOS

Entre las principales asociaciones tenemos:

#### 4.2.3.1 COMPRAS:

##### 4.2.3.1.1 compras\_cabecera\_proveedor

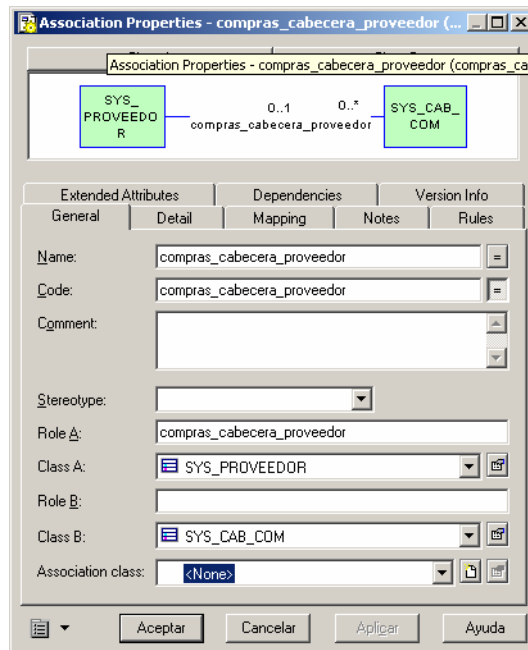


Figura No. 13 (Asociación de compras\_cabecera\_proveedor)



### 4.2.3.1.2 Compras\_cabecera\_detalle

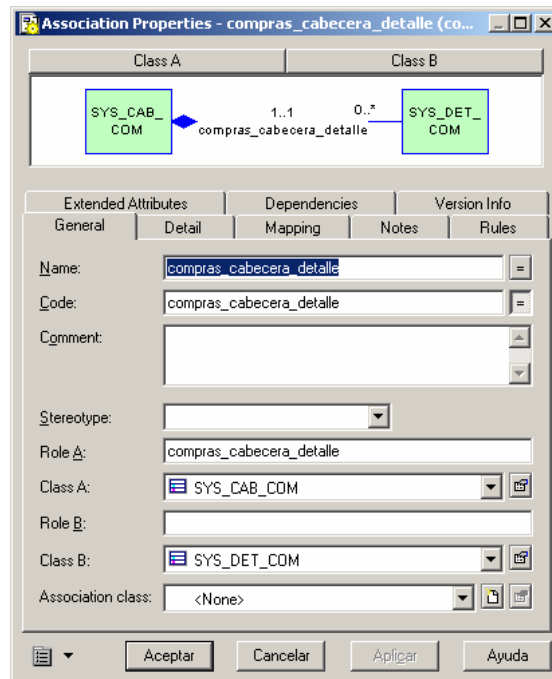


Figura No 14 (Asociación de compras\_cabecera\_detalle)

### 4.2.3.1.3 compras\_detalle\_producto

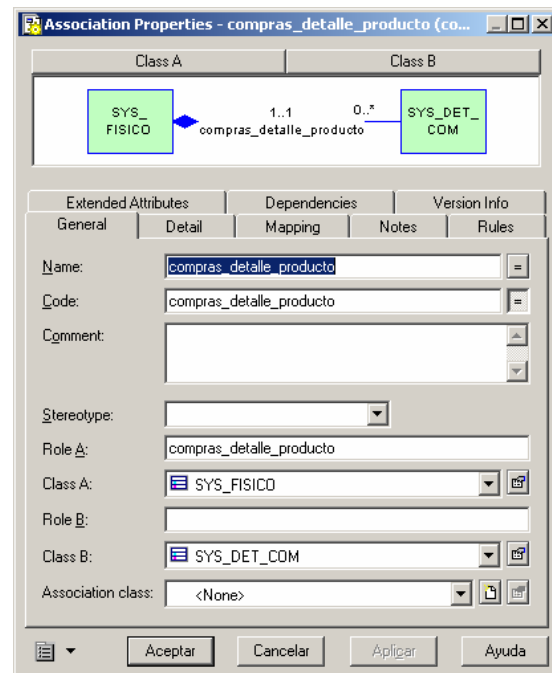


Figura No.15 (Asociación de compras\_detalle\_producto)



### 4.2.3.2 PEDIDOS:

#### 4.2.3.2.1 pedidos\_cabecera\_areas

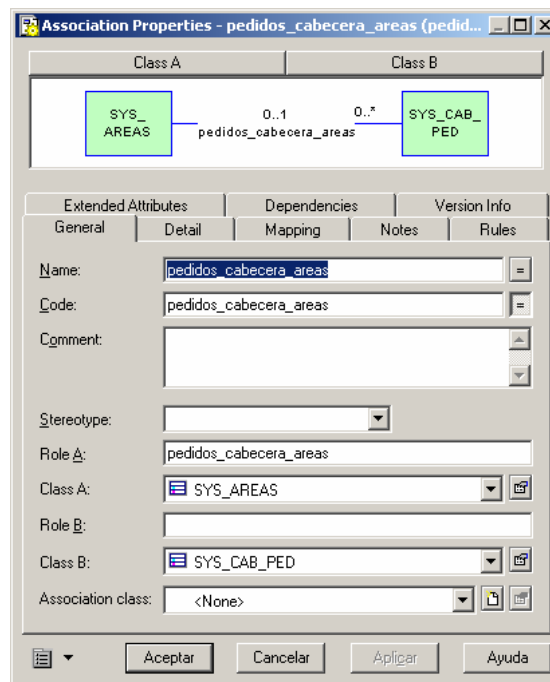


Figura No.16 (Asociación de pedidos\_cabecera\_areas)

#### 4.2.3.2.2 Pedidos\_cabecera\_detalle

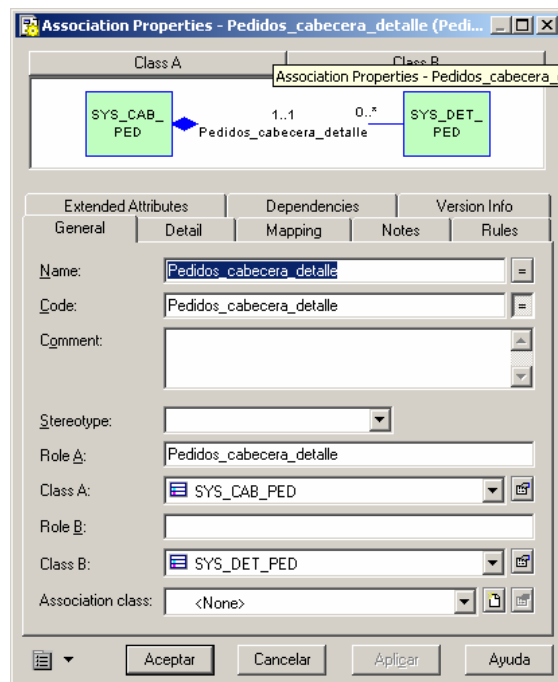


Figura No.17 (Asociación de pedidos\_cabecera\_detalle)



### 4.2.3.2 pedidos\_detalle\_producto

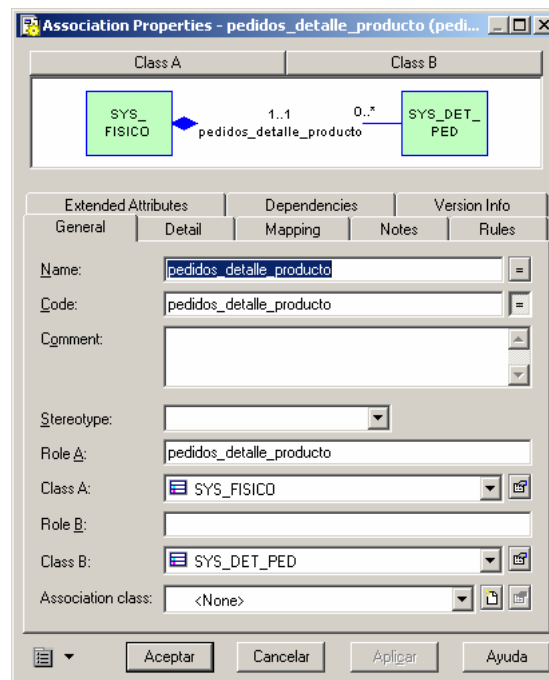


Figura No.18 (Asociación de pedidos\_detalle\_producto)

### 4.2.3.3 POS

#### 4.2.3.3.1 producto\_subreceta

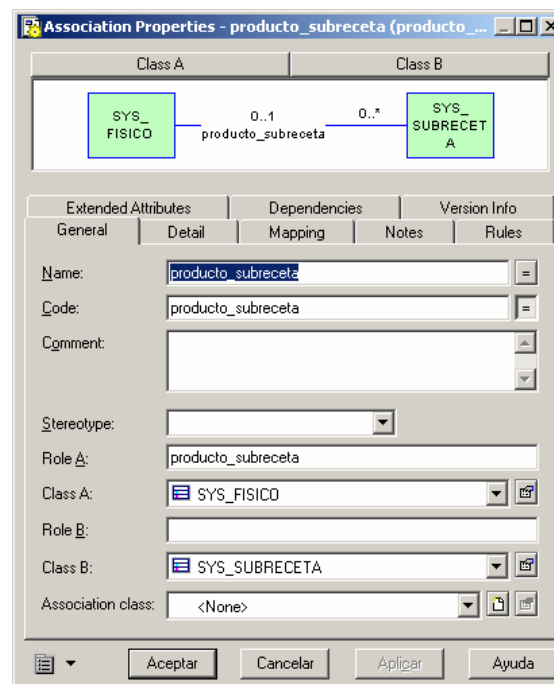


Figura No.19 (Asociación de producto\_subreceta)



### 4.2.3.3.2 producto\_receta

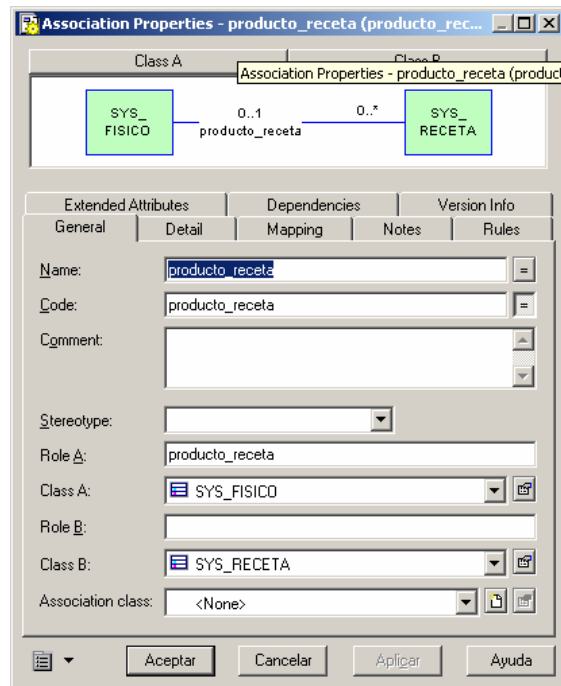


Figura No.20 (Asociación de producto\_receta)

### 4.2.3.3.2 receta\_subreceta

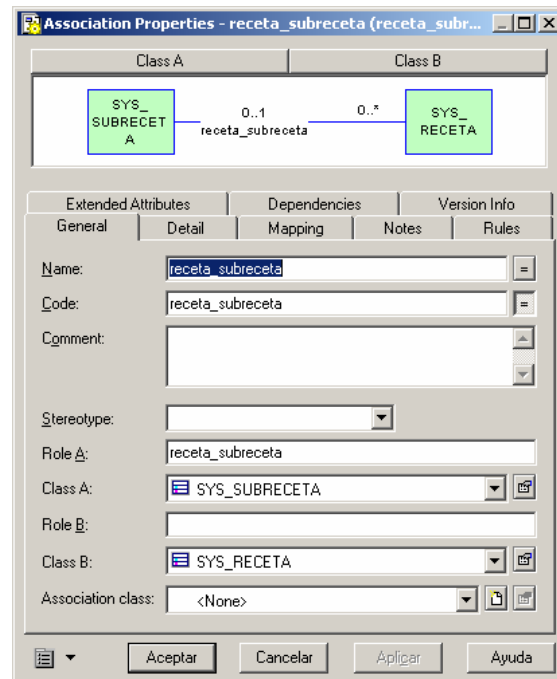


Figura No.21 (Asociación de receta\_subreceta)





### 4.2.3.3.3 pos\_cabecera\_detalle

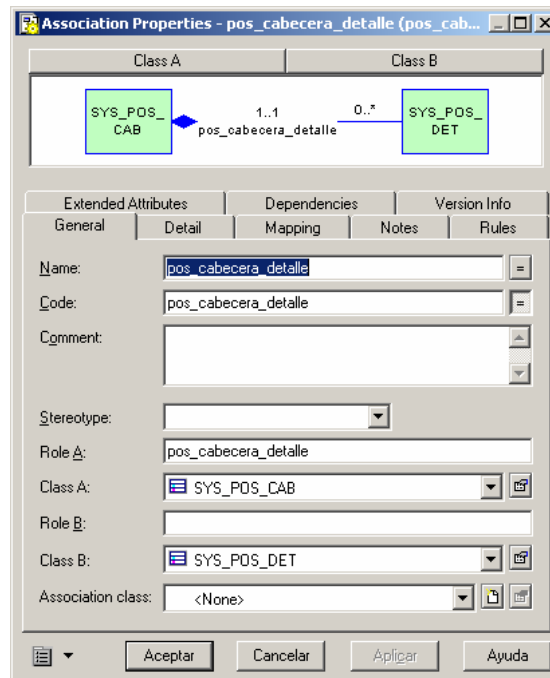


Figura No.22 (Asociación de pos\_cabecera\_detalle)

### 4.2.3.3.4 pos\_detalle\_receta

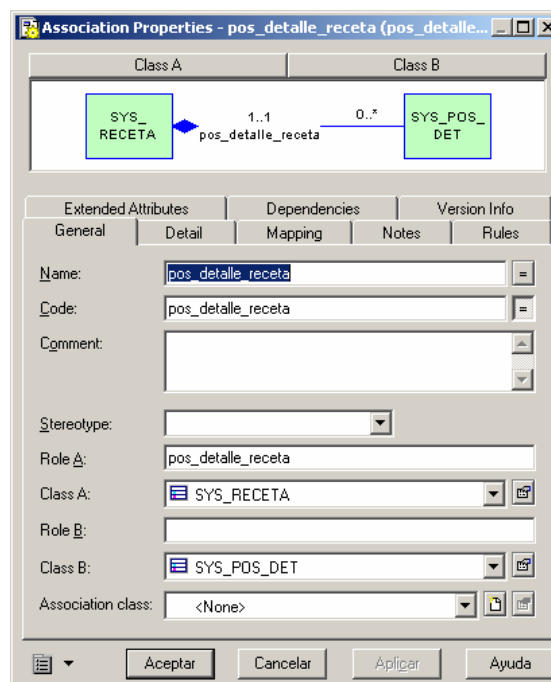


Figura No.23 (Asociación de pos\_detalle\_receta)



## 4.2.3.4 INVENTARIO

### 4.2.3.4.1 productos\_bodega

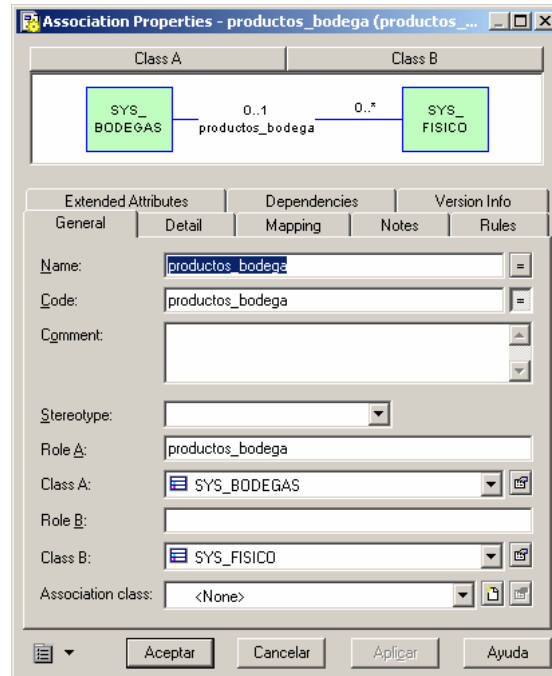


Figura No.24 (Asociación de productos\_bodega)

### 4.2.3.4.2 producto\_clase

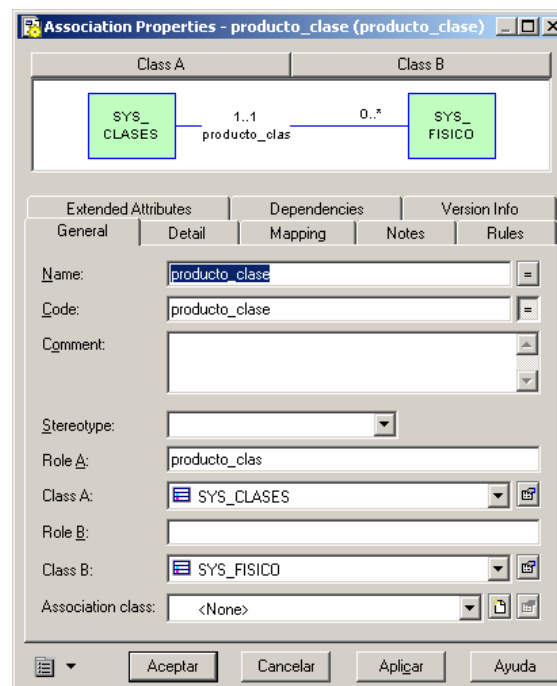


Figura No.25 (Asociación de producto\_clase)



### 4.2.3.4.3 producto\_conteo

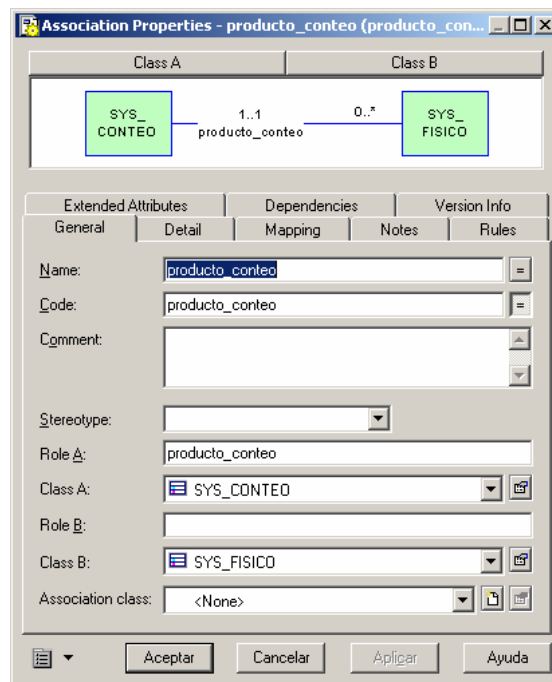


Figura No.26 (Asociación de producto\_conteo)

### 4.2.3.4.4 conteo\_fisico\_semanal

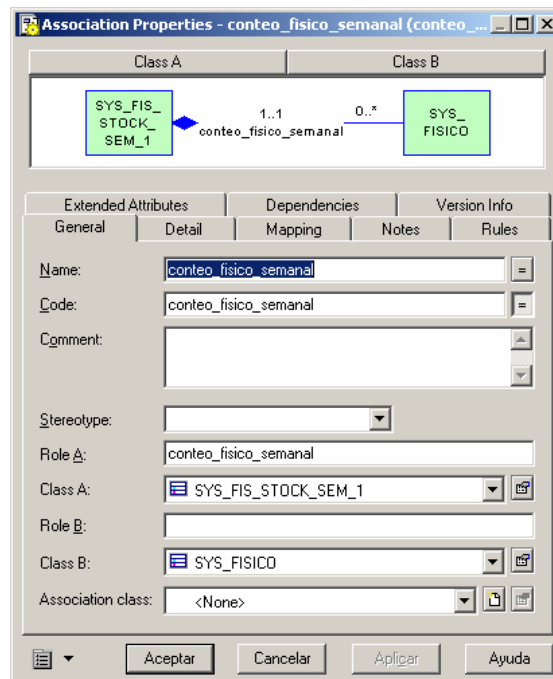


Figura No.27 (Asociación de conteo\_fisico\_semanal)



### 4.2.3.4.5 conteo\_fisico\_quincenal

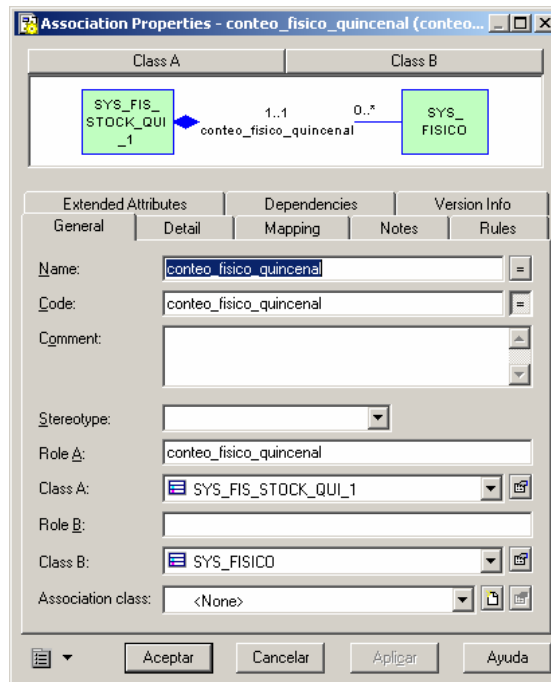


Figura No.28 (Asociación de conteo\_fisico\_quincenal)

### 4.2.4 DETERMINACIÓN DE HERENCIA

Aquí determinamos las tablas principales del sistema:

SYS_FISICO	
+ descripcion	: String
+ unidad	: String
+ stock_inicial	: Float
+ Stock	: Float
+ Costo_inicial	: Double
+ Costo_promedio	: Double
+ costo_compra	: Double
+ costo	: Double
+ unidad_compra	: String
+ tipo_impuesto	: String
+ CODIGO_PROVEEDOR	: String
+ maximos	: Integer
+ minimos	: Integer

De aquí se hereda las siguientes tablas:

SYS_FIS_STOCK_DIA_1	
+ codigo_prod	: String
+ descripcion	: String
+ tipo	: String
+ unidad	: String
+ stock_inicial	: Integer
+ Stock	: String
+ codigo_clase	: String
+ CODIGO_PROVEEDOR	: String

SYS_FIS_STOCK_MEN_1	
+ codigo_prod	: String
+ descripcion	: String
+ tipo	: String
+ unidad	: String
+ stock_inicial	: Integer
+ Stock	: String
+ codigo_clase	: String
+ CODIGO_PROVEEDOR	: String

SYS_FIS_STOCK_QUI_1	
+ codigo_prod	: String
+ descripcion	: String
+ tipo	: String
+ unidad	: String
+ stock_inicial	: Integer
+ Stock	: String
+ codigo_clase	: String
+ CODIGO_PROVEEDOR	: String

SYS_FIS_STOCK_SEM_1	
+ codigo_prod	: String
+ descripcion	: String
+ tipo	: String
+ unidad	: String
+ stock_inicial	: Integer
+ Stock	: String
+ codigo_clase	: String
+ CODIGO_PROVEEDOR	: String

Figura No.29 (Tablas Heredadas)



### **4.3 MODELAMIENTO DINÁMICO**

Dentro de este modelamiento se representan todos los aspectos relacionados con las secuencias posibles de eventos (vidas posibles) y la interacción entre objetos.

#### **4.3.1 ESCENARIO DE INTERACCIÓN:**

Para tener este escenario clasificamos en diagramas de Secuencia y Colaboración.

##### **4.3.1.1 DIAGRAMA DE SECUENCIA:**

Este muestra la interacción del conjunto de objetos que interactúan a través del tiempo.



### 4.3.1.1.1 DIAGRAMA GENERAL

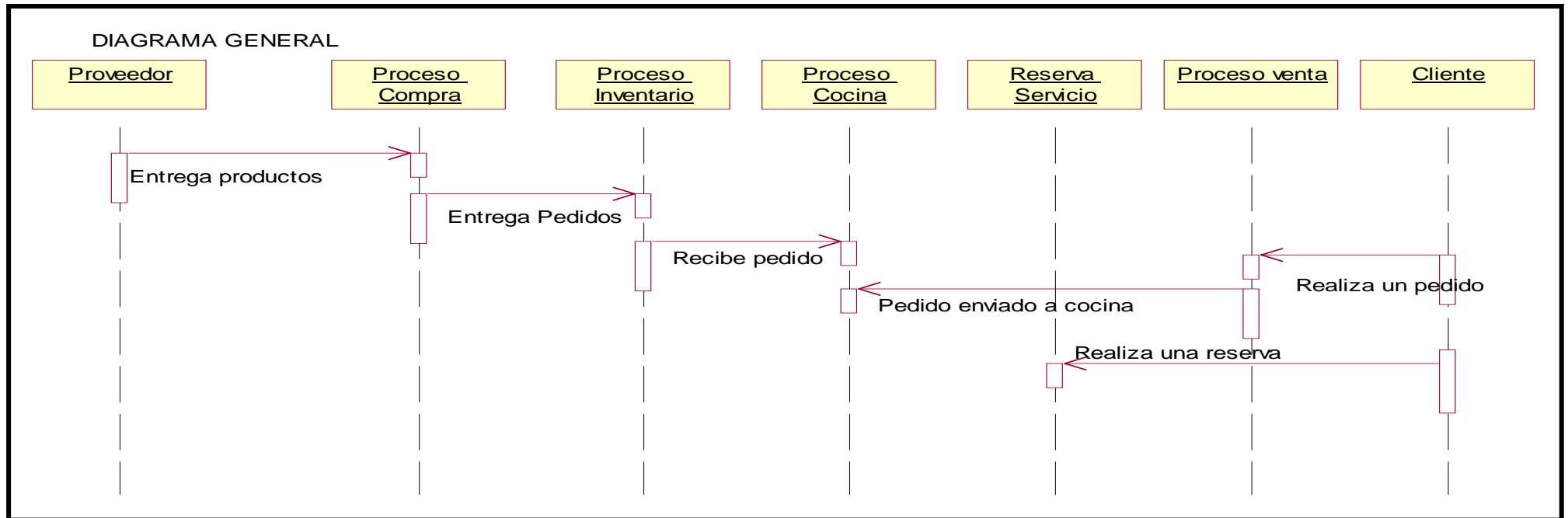


Figura No.30 (Secuencia de Diagrama General)

El diagrama general especifica de manera global los procesos que abarca nuestro sistema. Se especifica paso a paso, cómo el proveedor entrega los productos; esto es enviado al proceso de compra; este proceso se debe registrar en el proceso de inventarios para ir actualizando lo que existe en el hotel; de aquí se entrega todos los productos a cocina para registrarlos y trabajarlos.

En la parte de reservaciones tenemos que tomar en cuenta al cliente, el cual no solo realiza una reservación, también hace un pedido en el restaurante del hotel, por eso se crea el proceso de venta, el cual verifica si es una reservación o es un pedido e inmediatamente envía al proceso requerido. Este diagrama representa de forma global como se ha estructurado al sistema y los diversos procesos contiene



### 4.3.1.1.2 DIAGRAMA INVENTARIO

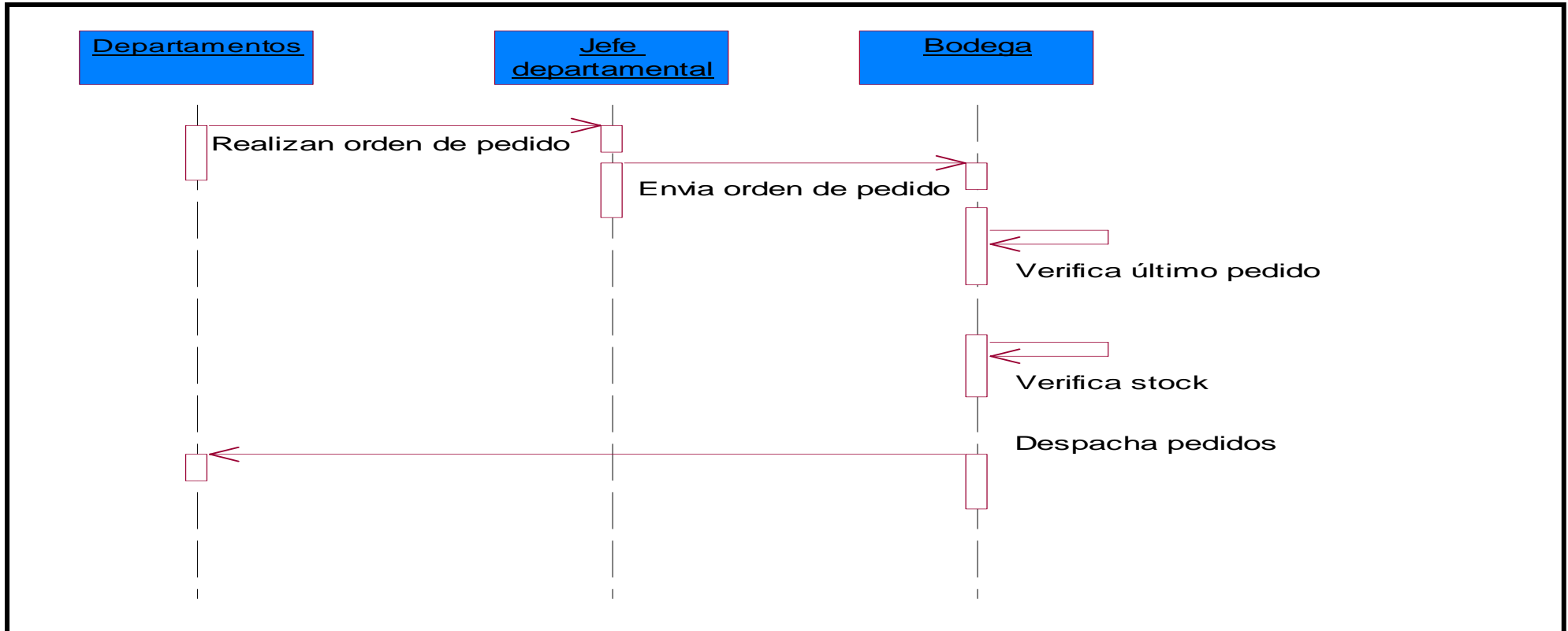


Figura No.31 (Secuencia de Diagrama Inventario)

El diagrama de Inventarios representa los pedidos que realiza cada departamento; éste realiza una orden de pedido, la cual es enviada al jefe del departamento; éste puede rechazar o aceptar la orden. Al aceptarla, ésta es enviada a la bodega, en la cual se verifican los pedidos que ha tenido el departamento, verifica el stock y despacha el pedido al departamento.



### 4.3.1.1.3 DIAGRAMA VENTAS

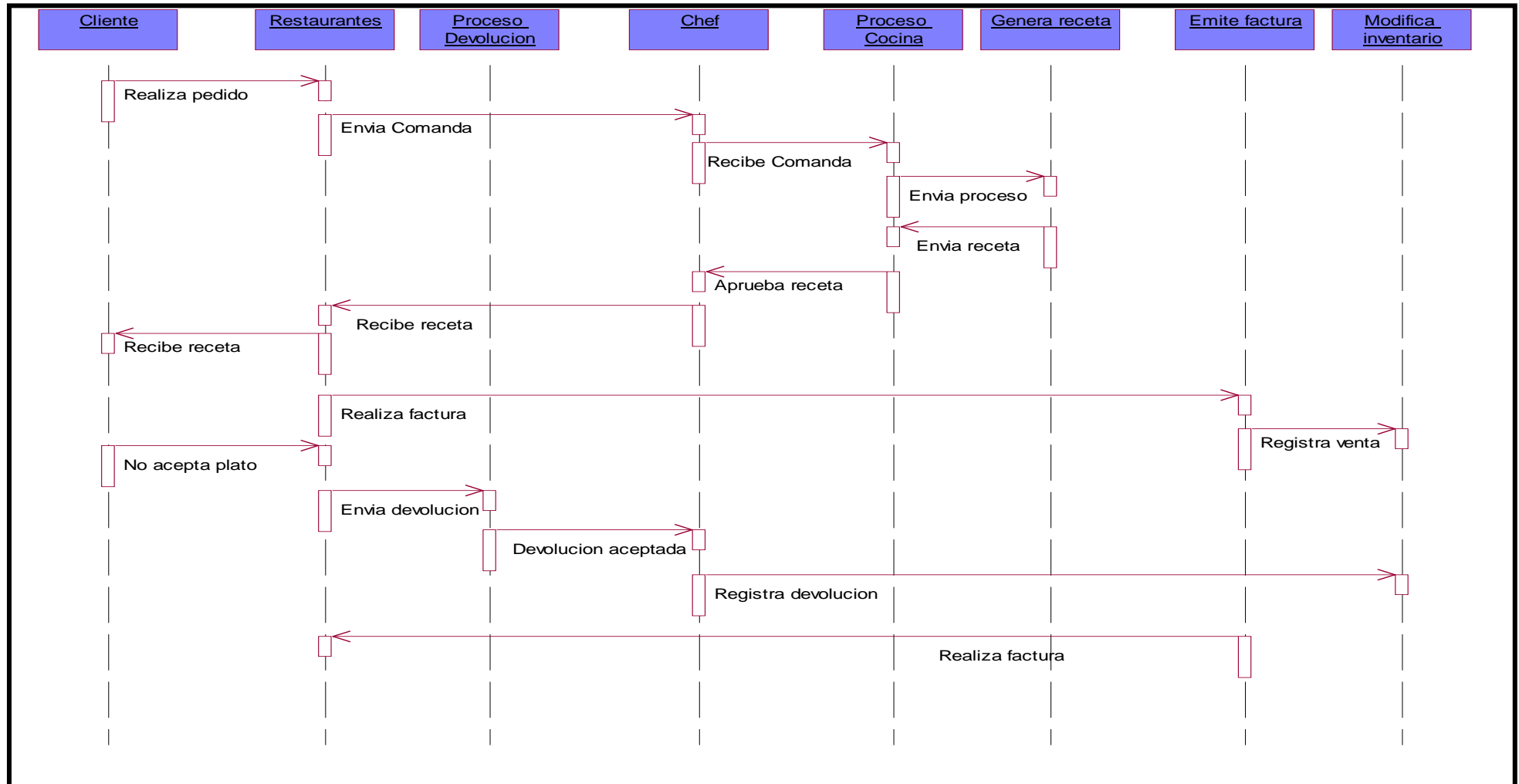


Figura No.32 (Secuencia de Diagrama Ventas)





El diagrama de ventas es la interacción del cliente con el hotel para la realización de pedidos y reservaciones. Se puede ver paso a paso la comunicación del Hotel al realizar una venta.

El cliente realiza un pedido al restaurante, el restaurante envía una comanda al chef (ésta no es más que la especificación del pedido del cliente). El chef revisa la comanda y envía al proceso de cocina; éste genera la receta y es enviada al chef para su aprobación. Después de aprobarla éste entrega al restaurante para que el cliente deguste su orden. Después de terminado esto, el restaurante emite la factura y registra la venta para poder modificar el inventario.

En caso de que el cliente no acepte el producto, éste envía el plato al restaurante; se realiza un proceso de devolución, el cual es finalizado con la aceptación del chef y se registra la devolución en el inventario. Si el cliente canceló con anterioridad, se le emite la factura de devolución y queda finalizado el sistema de ventas.



#### 4.3.1.1.4 DIAGRAMA DEVOLUCION

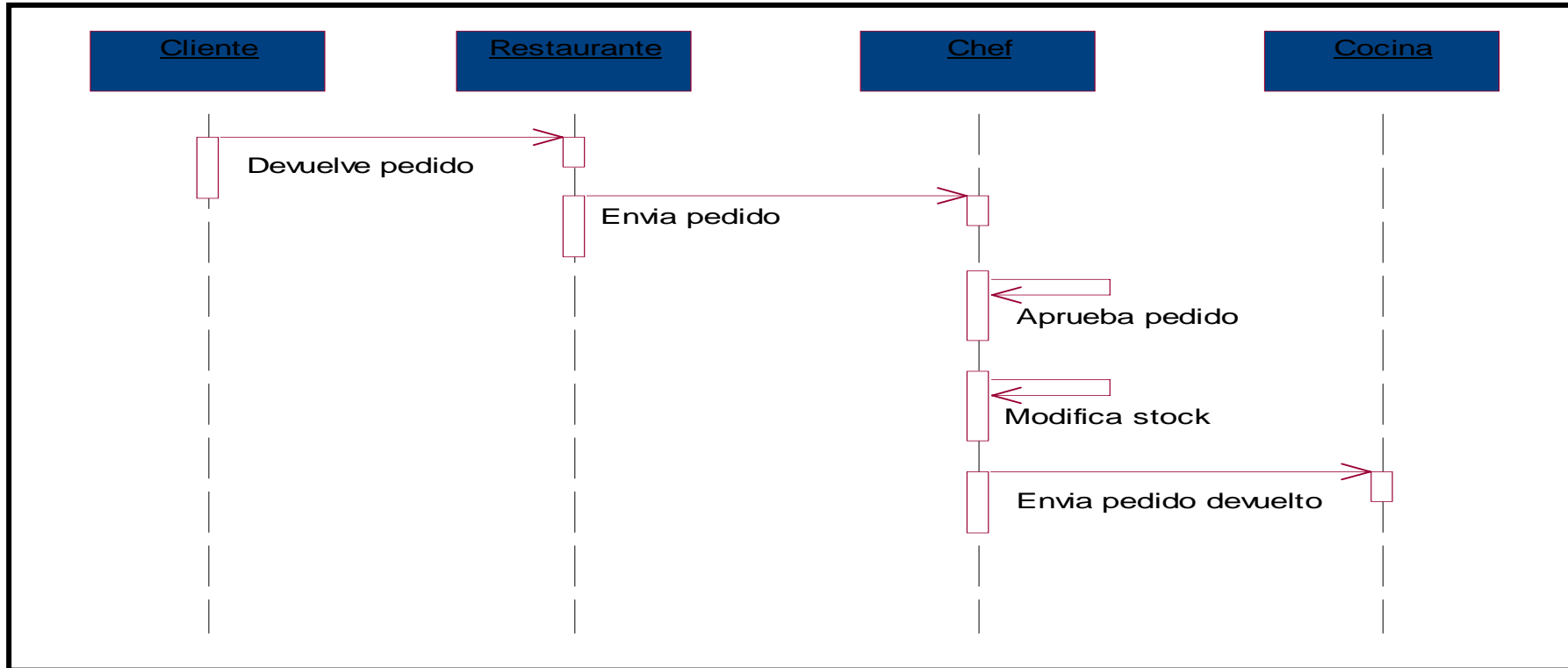


Figura No.33 (Secuencia de Diagrama Devolución)

Se inicia este proceso con la devolución del cliente al restaurante; éste regresa al chef, el cual aprueba la devolución del pedido, modifica el stock en el inventario y finaliza enviando el pedido a cocina.



### 4.3.1.1.5 DIAGRAMA GENERA RECETA

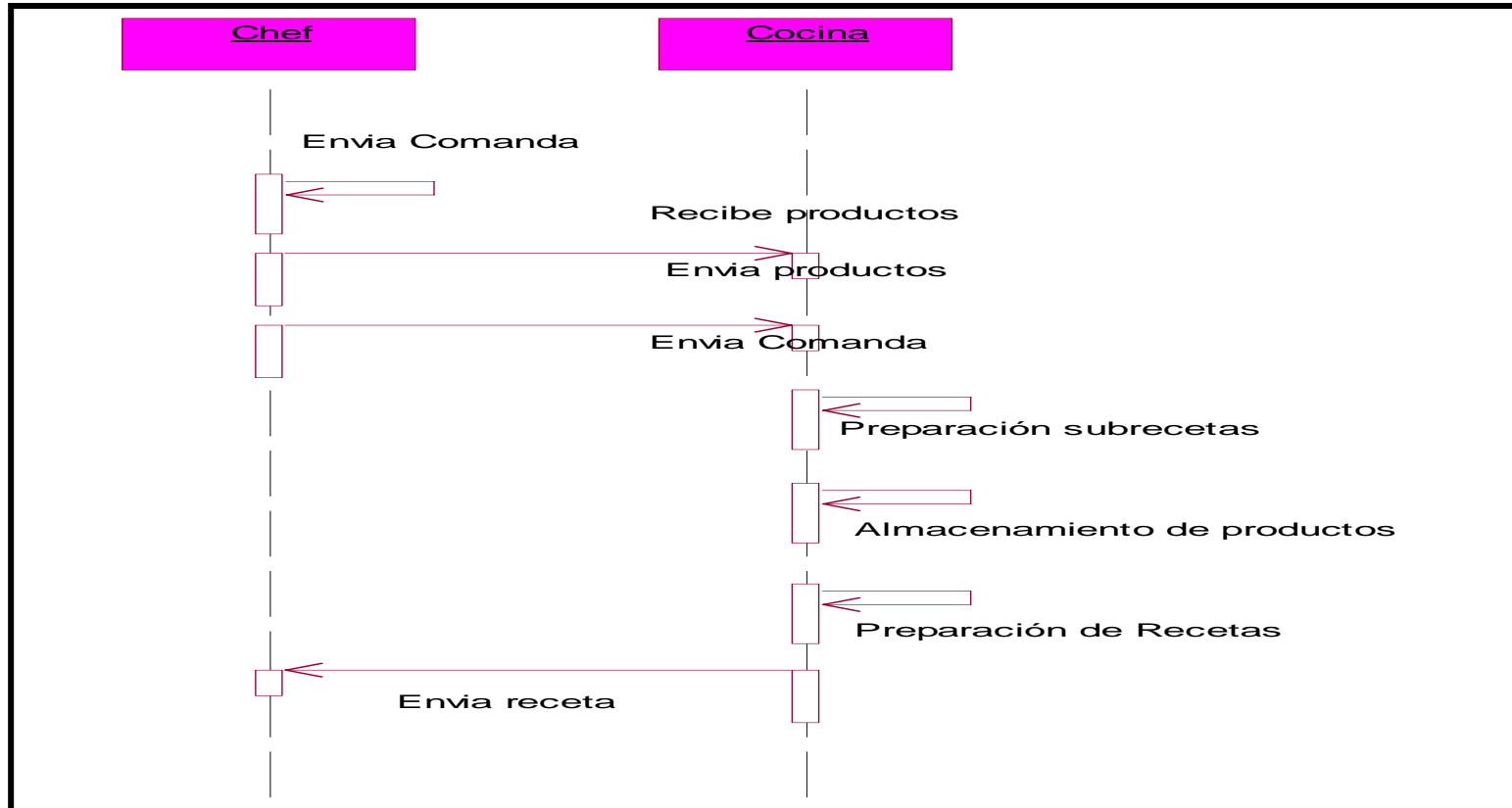


Figura No.34 (Secuencia de Diagrama Genera Receta)

Para generar una receta el restaurante envía la comanda al chef, el cual, a parte de recibir el pedido, también se hace cargo de los productos, los cuales son enviados al proceso de cocina. Este devuelve la generación de subrecetas, recetas, almacenamiento de producto y finalmente envía la receta para ser controlada y pueda llegar a manos del usuario.



### 4.3.1.1.6 DIAGRAMA COCINA

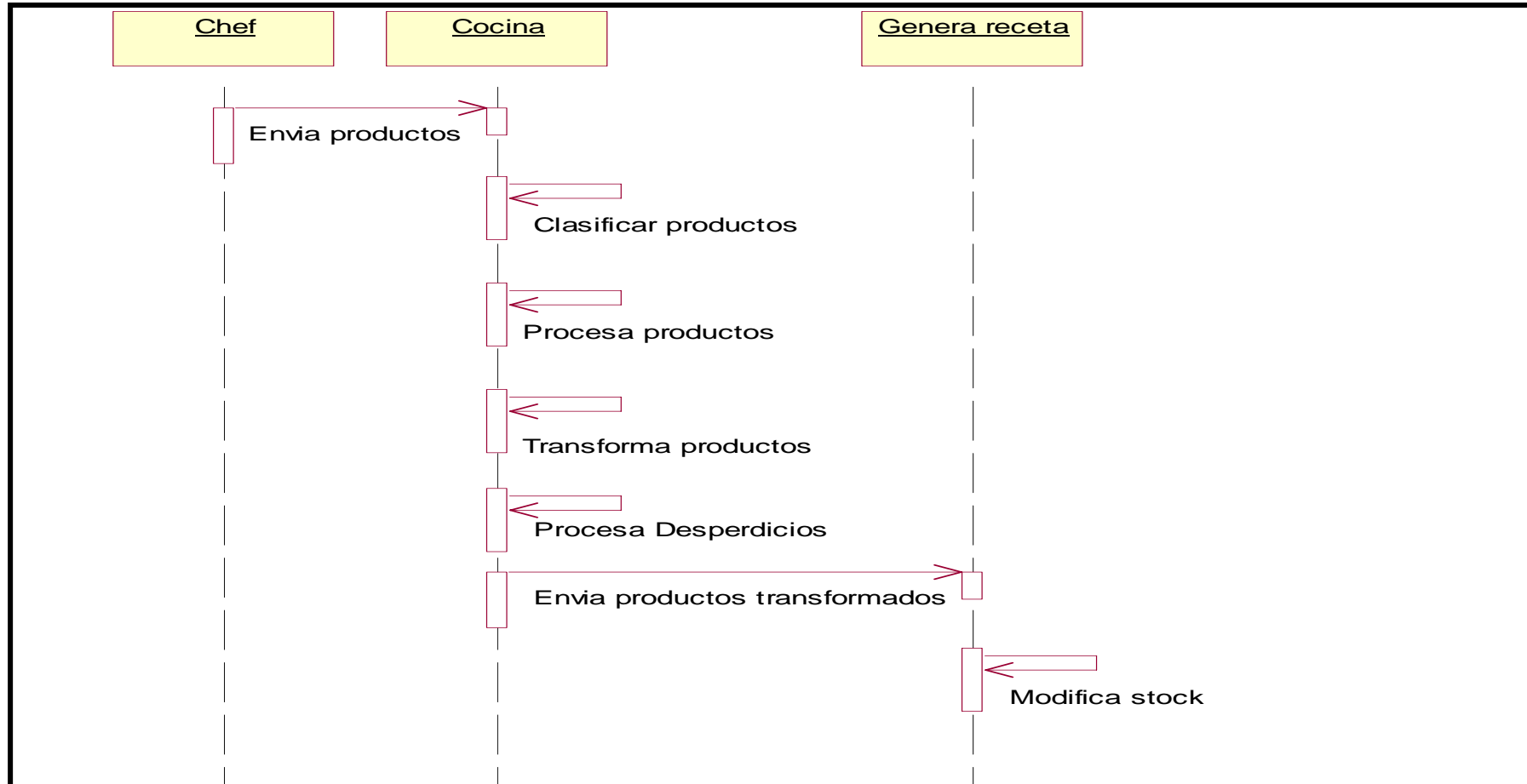


Figura No.35 (Secuencia de Diagrama Cocina)

Cocina recibe los productos del chef; aquí se encarga de clasificar productos, procesar productos y transformar los mismos para poder procesar los desperdicios. Todos los productos transformados son llevados al proceso de generar receta y, dentro de este, se modifica el stock para el inventario



### 4.3.1.1.7 DIAGRAMA RESERVA SALONES

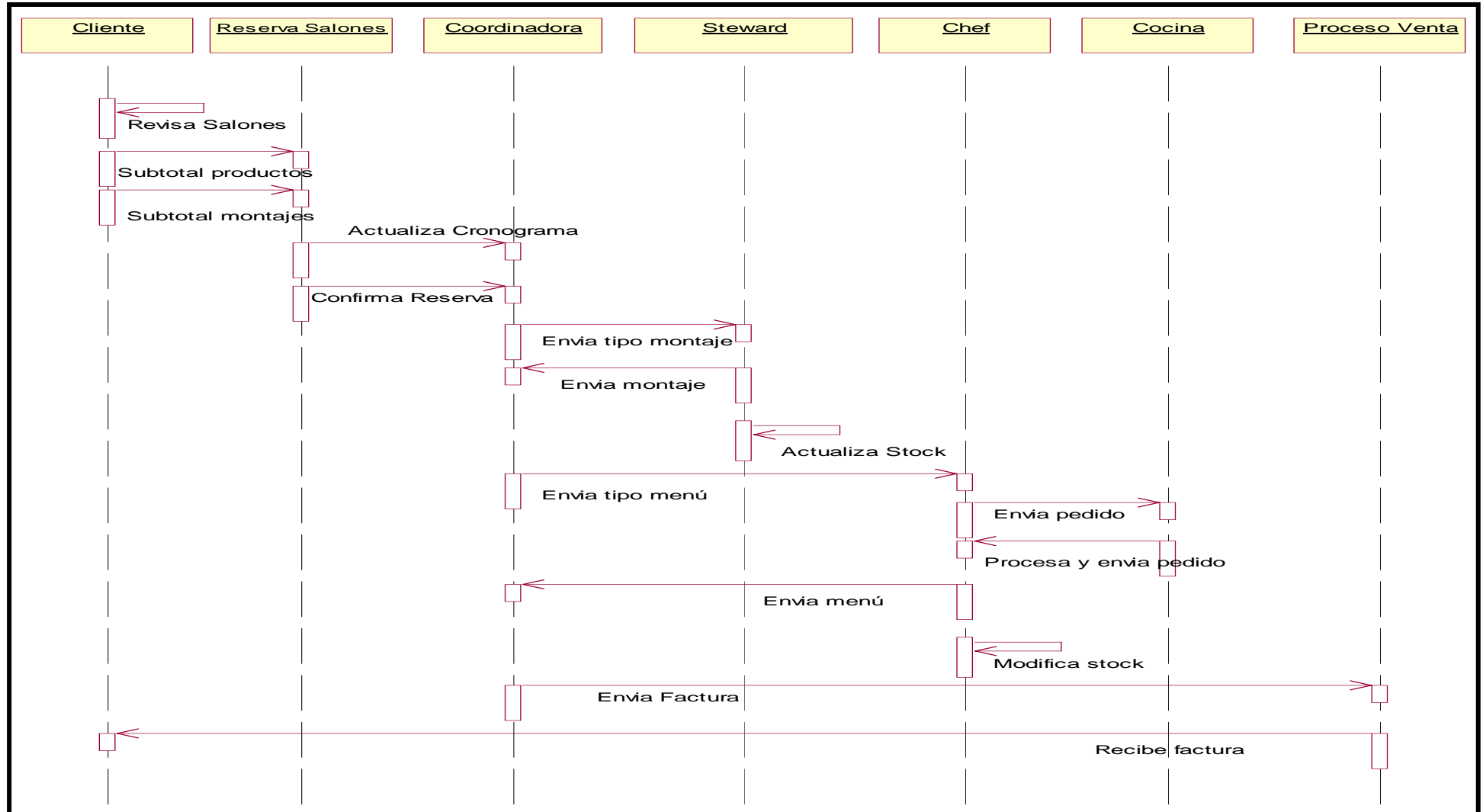


Figura No.36 (Secuencia de Diagrama Reserva Salones)



El cliente revisa los salones; con esto se envía un subtotal de productos y un subtotal de montajes para poder actualizar la reserva; dentro de ésta se actualiza un cronograma de eventos y se confirma la reserva; se envía a la coordinadora el tipo de montaje, éste confirma la reservación y costos.

Una vez ya confirmado se procede a la reserva del salón escogido el cual es entregado al departamento de Steward para la reservación de dicho montaje; este actualiza stock y acepta el montaje, al mismo tiempo, la coordinadora envía al chef el menú para el evento; éste pide a cocina que lo procesen, modifiquen stock y se regresa el menú listo a la coordinadora, la cual se encarga de la facturación y entrega de factura al cliente.



### 4.3.1.1.8 DIAGRAMA COMPRA

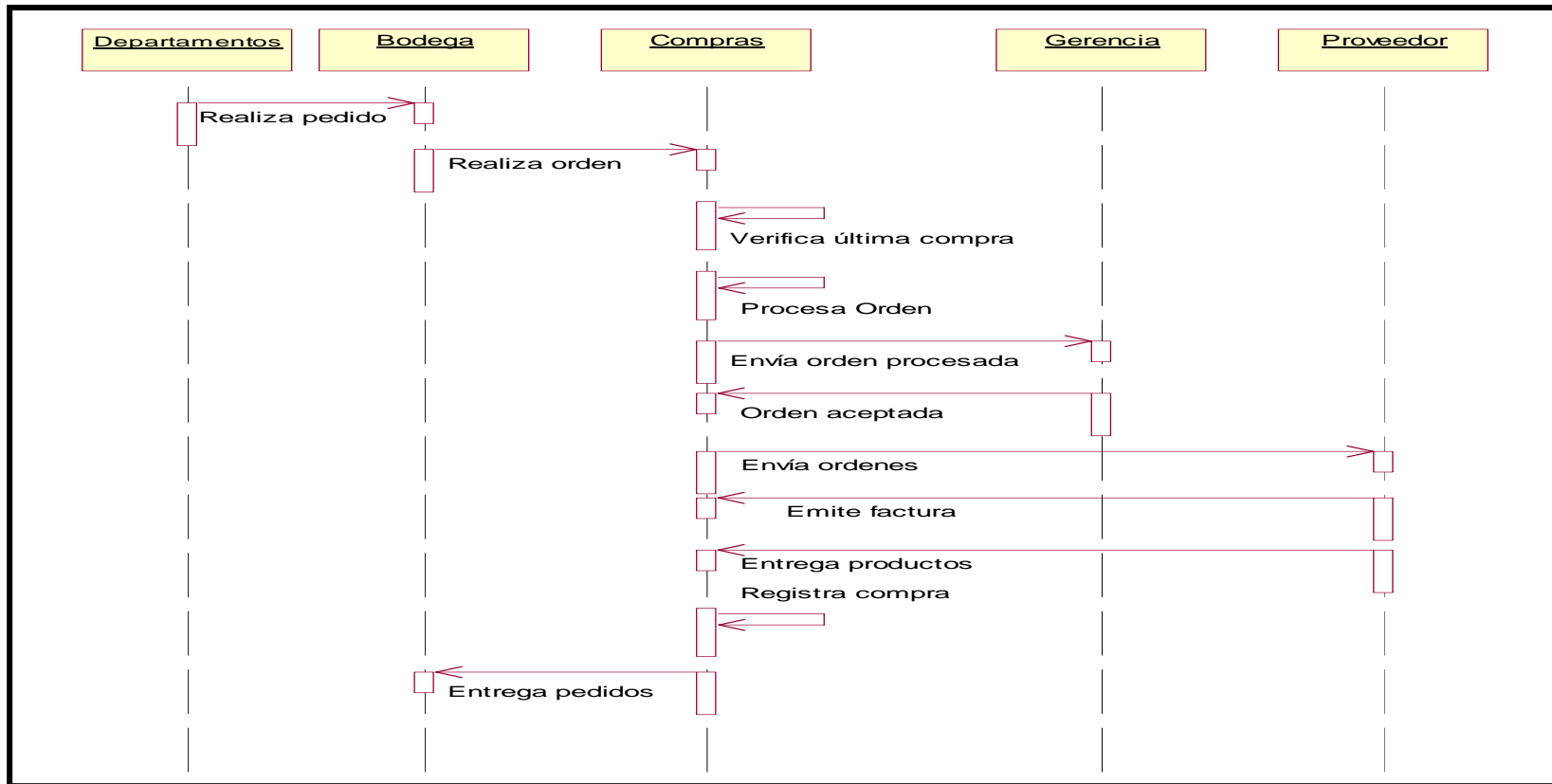


Figura No.37 (Secuencia de Diagrama Compra)

Cada departamento realiza un pedido a bodega; ésta es encargada de realizar la orden a compras; compras se encarga de verificar la última compra, procesa la orden, la cual es enviada a la gerencia. De aquí se acepta o se rechaza; si la orden es aceptada se envía una orden general al proveedor, el cual emite su factura, entrega los productos y el Hotel registra la compra y entrega los pedidos a cada departamento



#### **4.3.1.2 DIAGRAMAS DE COLABORACIÓN**

Se da inicio a la fase exploratoria para identificar objetos. La distribución de los objetos en el diagrama permite observar adecuadamente la interacción de un objeto con respecto de los demás.





### 4.3.1.2.1 DIAGRAMA GENERAL

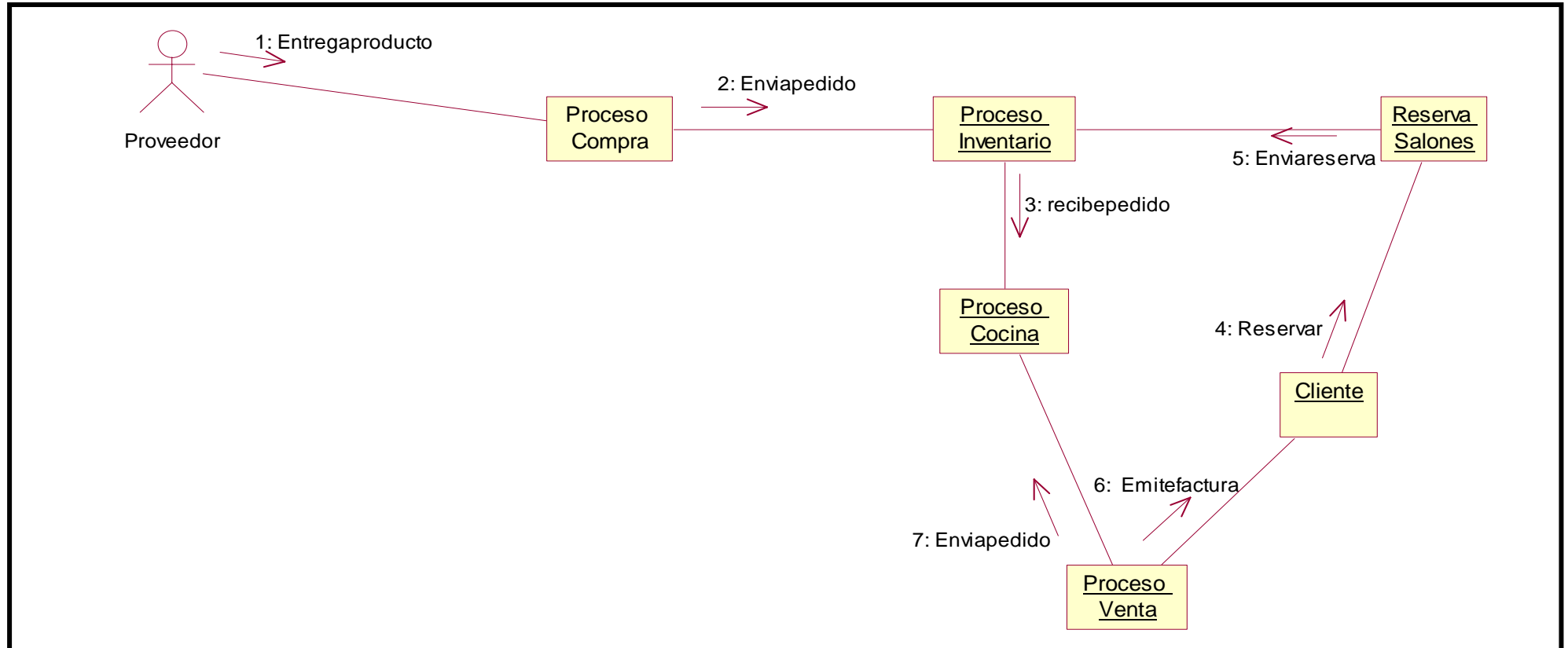


Figura No.38 (Colaboración de Diagrama General)

Se puede ver cómo los procesos Principales interactúan para la lógica de realización del proceso. Desde este diagrama podemos observar los diversas partes que tendrán el sistema y cómo se unirán entre sí.



### 4.3.1.2.2 DIAGRAMA INVENTARIO

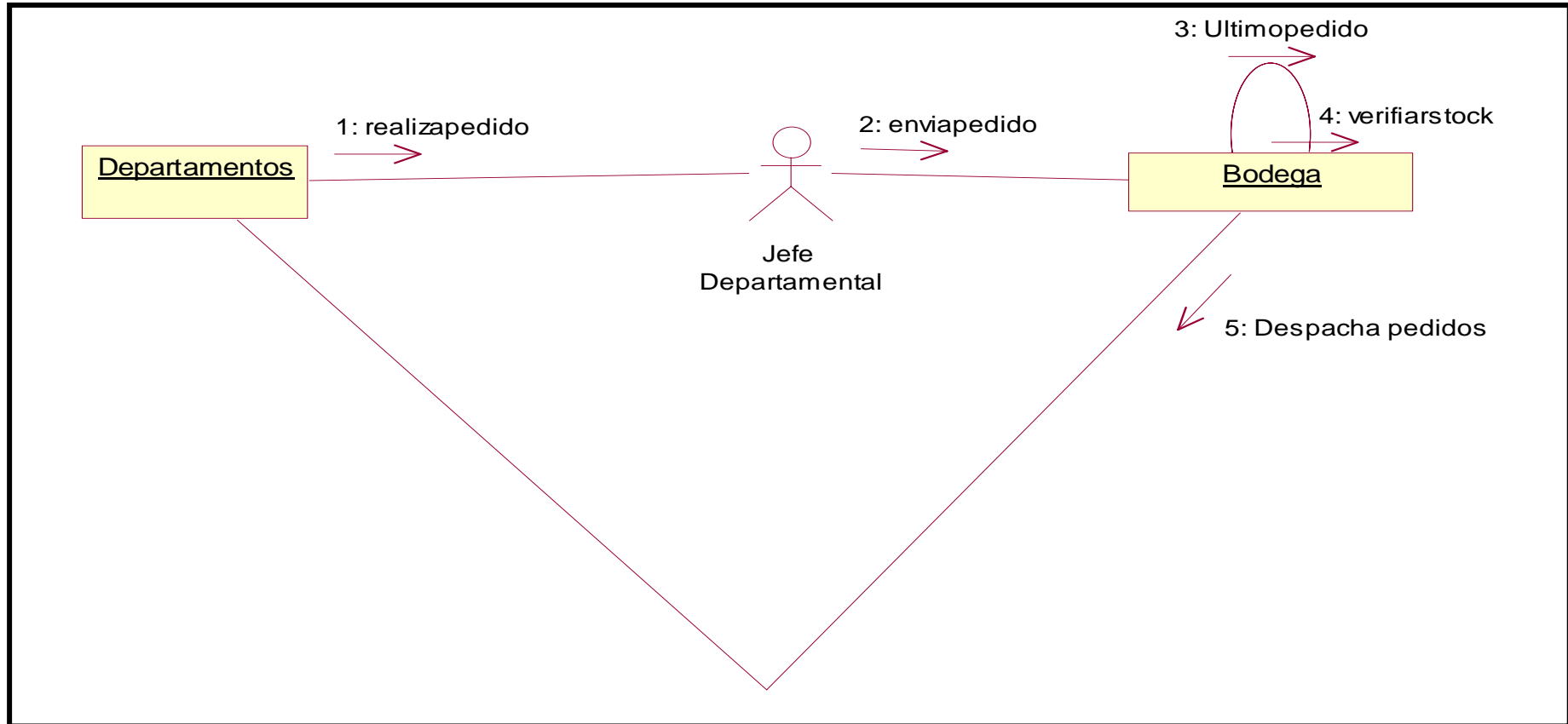


Figura No.39 (Colaboración de Diagrama Inventario)

El diagrama inventario tendrá como procesos: pedidos, verificación de stock, aprobación y despacho de pedidos. De esta manera se realiza la secuencia, inicia desde cuando cada departamento realiza un pedido, dicho pedido es receptado por el jefe departamental; este acepta y envía a revisión en bodega; con esto se despacha el pedido



### 4.3.1.2.3 DIAGRAMA VENTAS

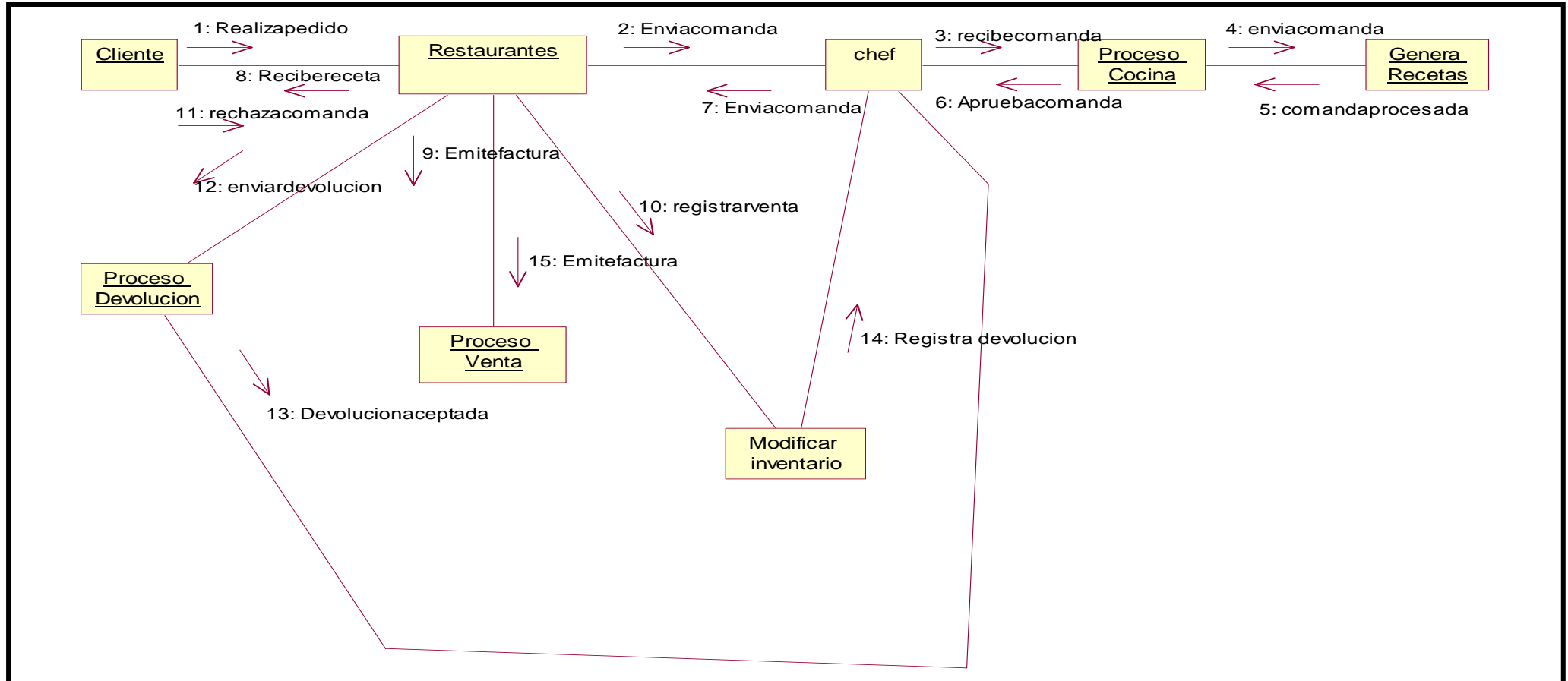


Figura No.40 (Colaboración de Diagrama Ventas)

Este diagrama posee los siguientes procesos: pedido cliente, envío pedido, genera receta, aprobación pedido, emitir factura, registra venta, modifica stock, devoluciones. Y representa la secuencia al obtener una venta. Primero se realiza el pedido, se envía pedido hasta llegar a generar receta, al terminar la receta es devuelto al cliente para su aprobación; al termino se le emite la factura. Si el cliente no está conforme con la receta, se realiza el proceso de devolución y se debe registrar esa baja de stock ya que los ingredientes fueron utilizados



#### 4.3.1.2.4 DIAGRAMA DEVOLUCION

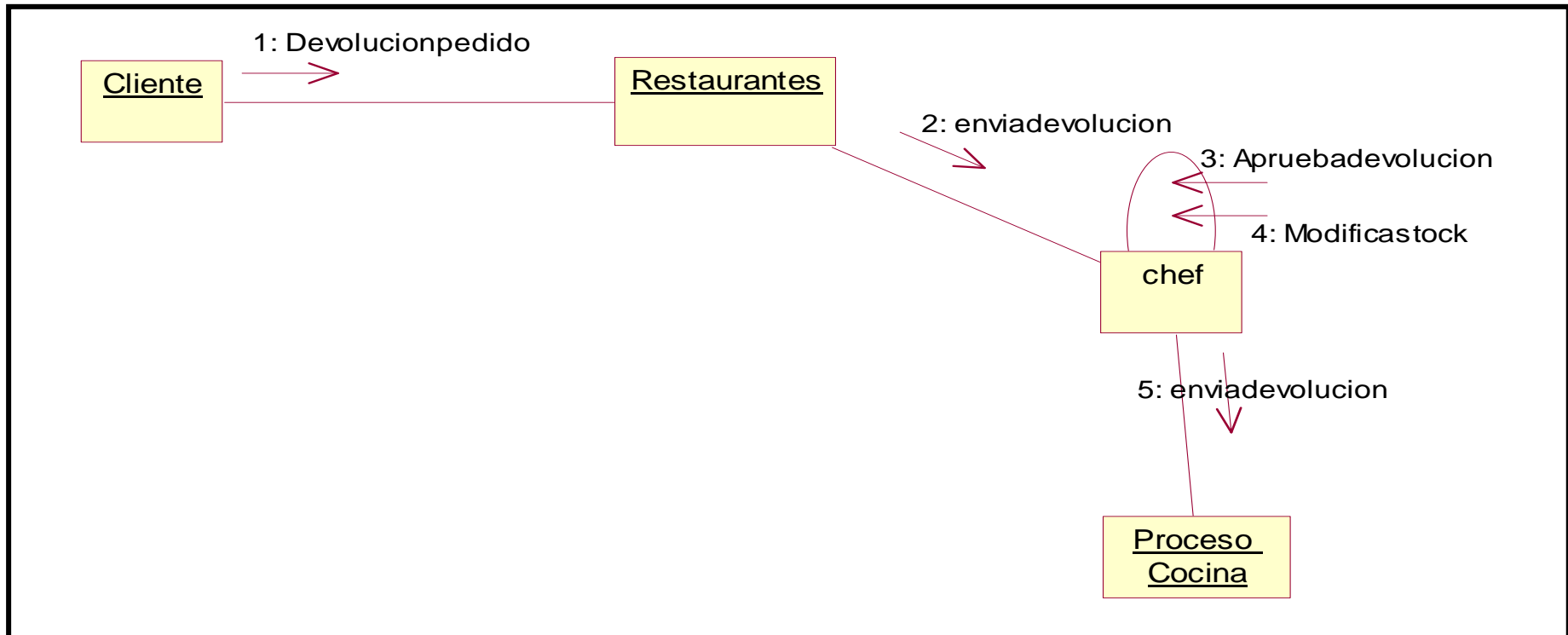


Figura No. 41 (Colaboración de Diagrama Devolución)

Este diagrama utiliza los procesos de: devolución pedido, aprueba devolución, modifica stock, envía proceso cocina. La secuencia que se debe seguir es, primero tener el pedido de devolución, se envía la devolución al chef éste aprueba e inmediatamente modifica stock y envía a cocina para su baja



#### 4.3.1.2.5 DIAGRAMA GENERA RECETA

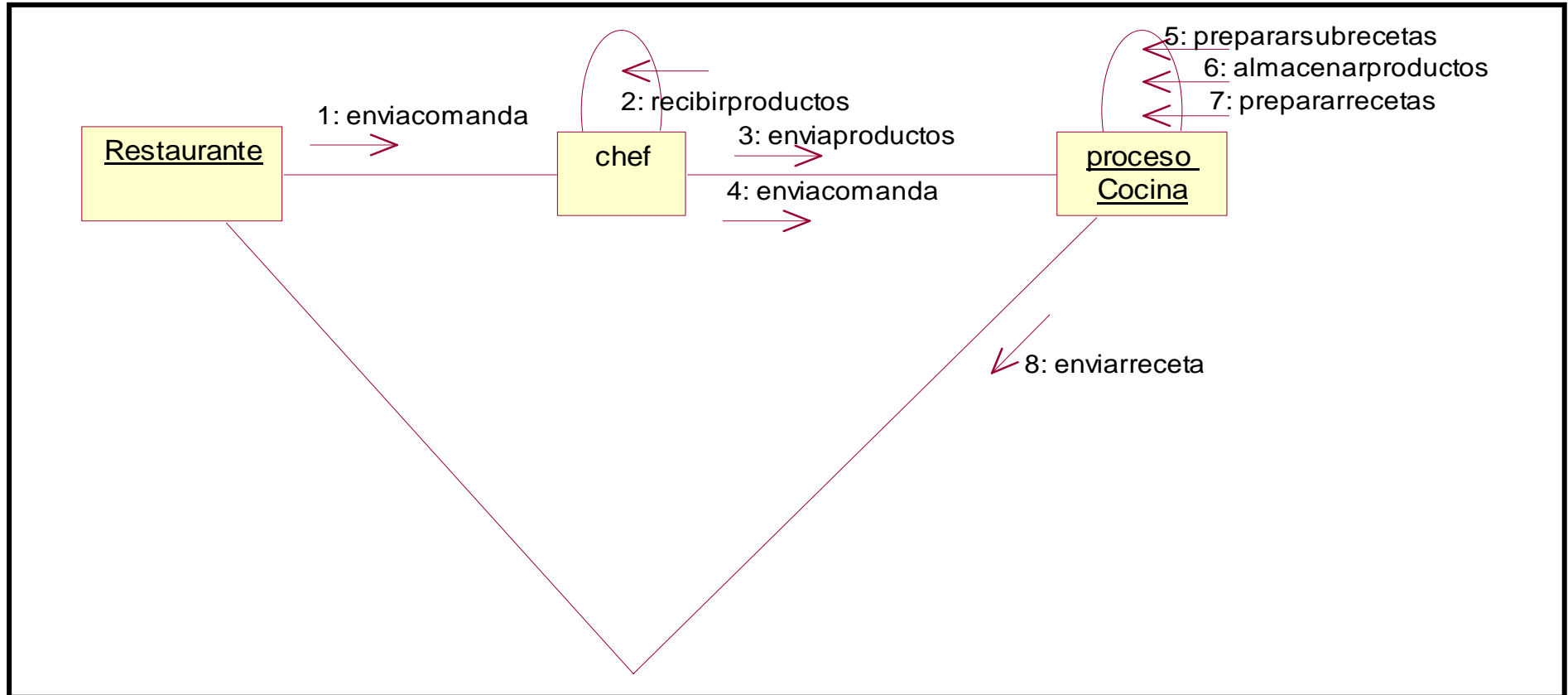


Figura No. 42 (Colaboración de Diagrama Genera Receta)

En este diagrama tenemos los procesos: envía comanda, recepción de productos, envío de productos a cocina, prepara subrecetas, enviar receta. Los pasos que realiza cocina para poder generar sus productos: se envía comanda al chef, éste recibe productos y envía a cocina; aquí elaboran subrecetas, recetas, almacenamiento de productos y envían la receta preparada.



### 4.3.1.2.6 DIAGRAMA COCINA

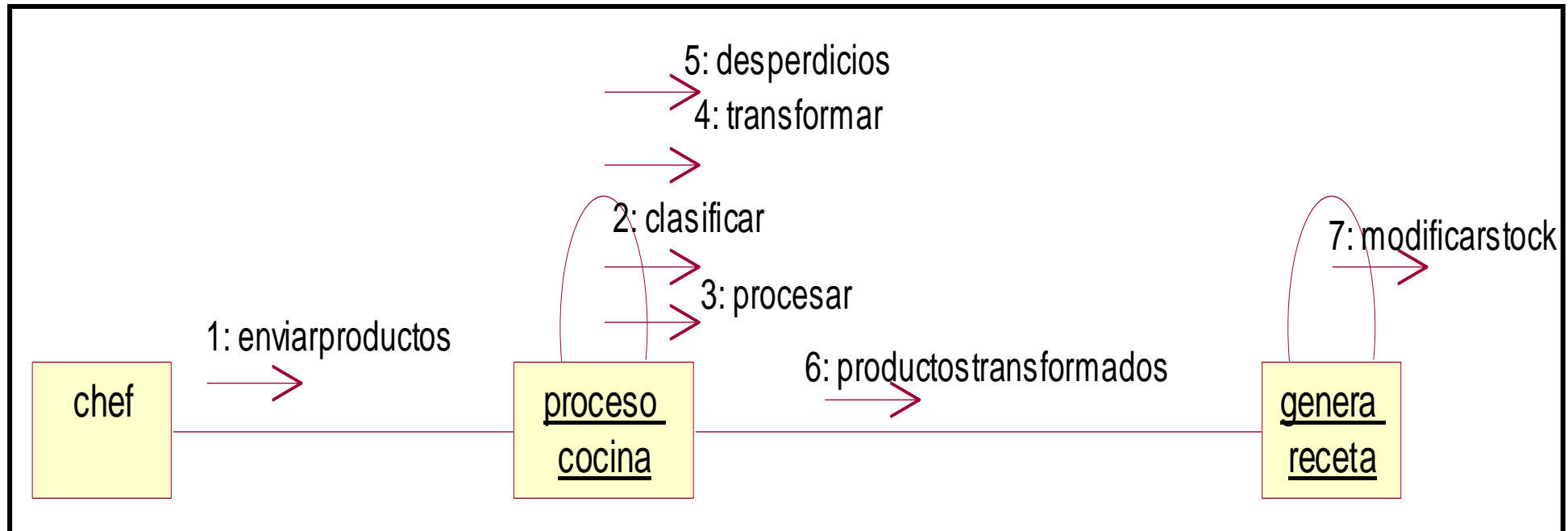


Figura No. 43 (Colaboración de Diagrama Cocina)

Los procesos que participan en este diagrama son: recepción de productos, ingreso de productos transformados, modificación stock, generar receta. La secuencia con que interactúan es: El chef envía los productos a cocina, ellos transforman los productos y clasifican los que son desperdicios.



### 4.3.1.2.7 DIAGRAMA COMPRAS

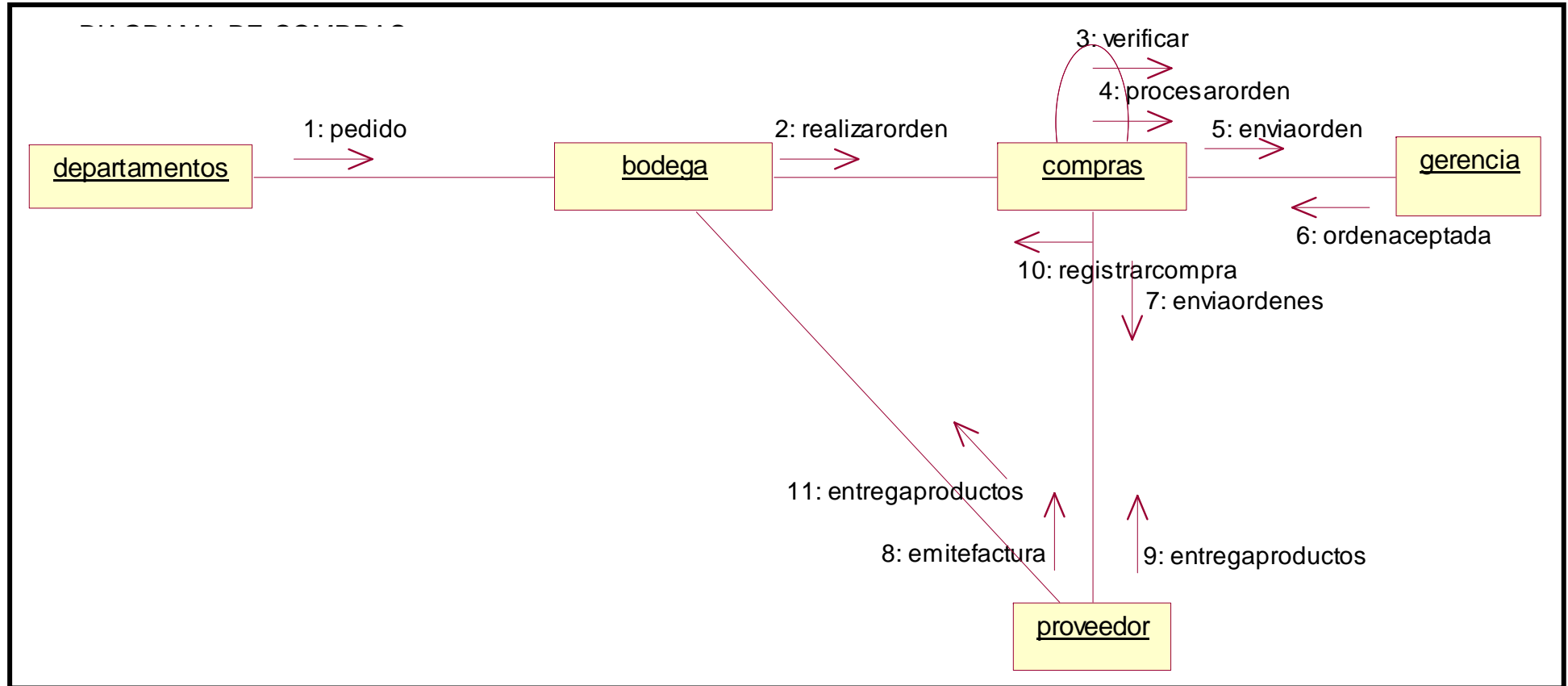


Figura No. 44 (Colaboración de Diagrama Compras)

Los procesos que interactúan en este diagrama son: pedido departamento, verificar pedidos, procesar órdenes, aceptar orden, adquisición de órdenes, registro de compra, entrega pedidos. La secuencia es la siguiente: cada departamento realiza su pedido, éste se envía a bodega para verificar últimos pedidos, procesan la orden y la envían para que sea aprobada. Una vez aprobada la orden, se encargan de la compra de productos, se realiza la recepción de los productos y la entrega a cada departamento



### 4.3.1.2.8 DIAGRAMA RESERVA SALONES

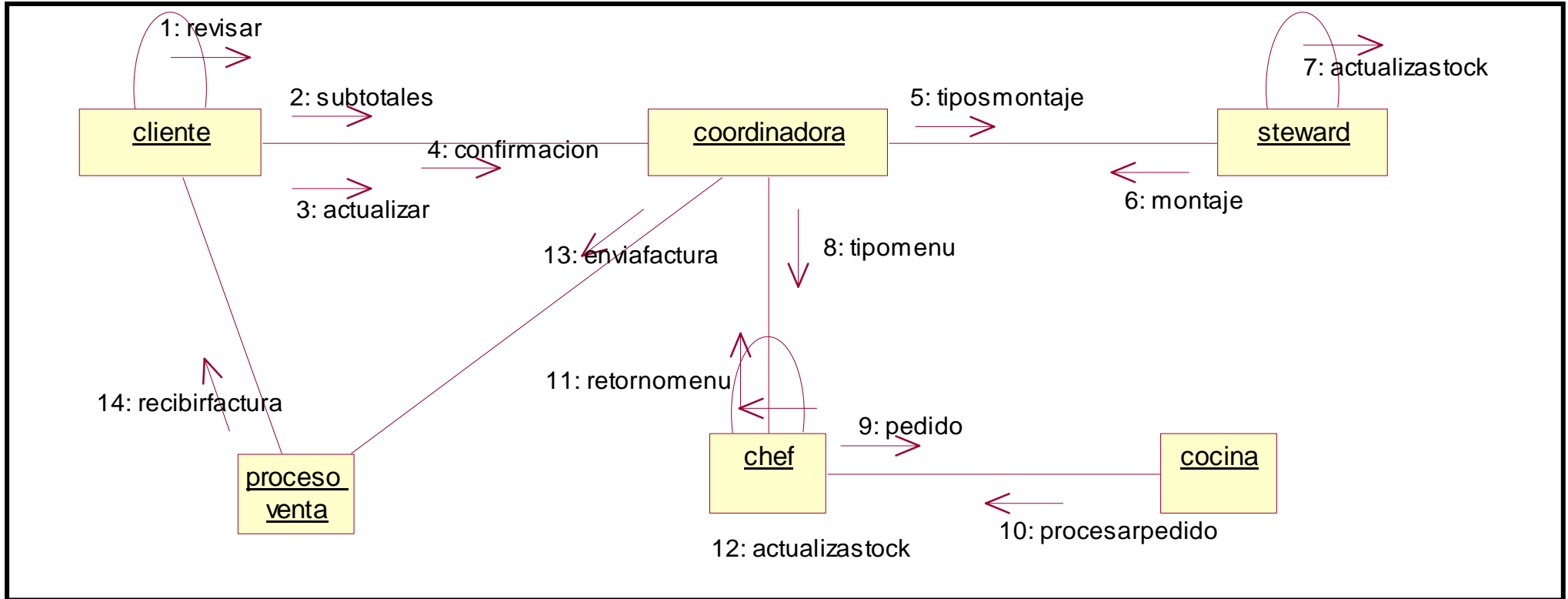


Figura No. 45 (Colaboración de Diagrama Reserva Salones)

Los procesos con que se maneja este diagrama son: revisar salones vía Web, subtotales, confirmación reserva, confirmación menú, actualización stock, emisión factura, procesa venta. La secuencia que sigue el diagrama es: revisar salones, realizar subtotal; se debe confirmar la reservación para que la coordinadora envíe el tipo de menú y montaje. Con esto se actualiza el stock, y se envía al proceso de venta.





### **4.3.2 ESCENARIOS DE ESTADOS**

El diagrama de estados representa las salidas que el sistema tendrá en el transcurso de sus reglas de negocio.



### 4.3.2.1 DIAGRAMA GENERAL

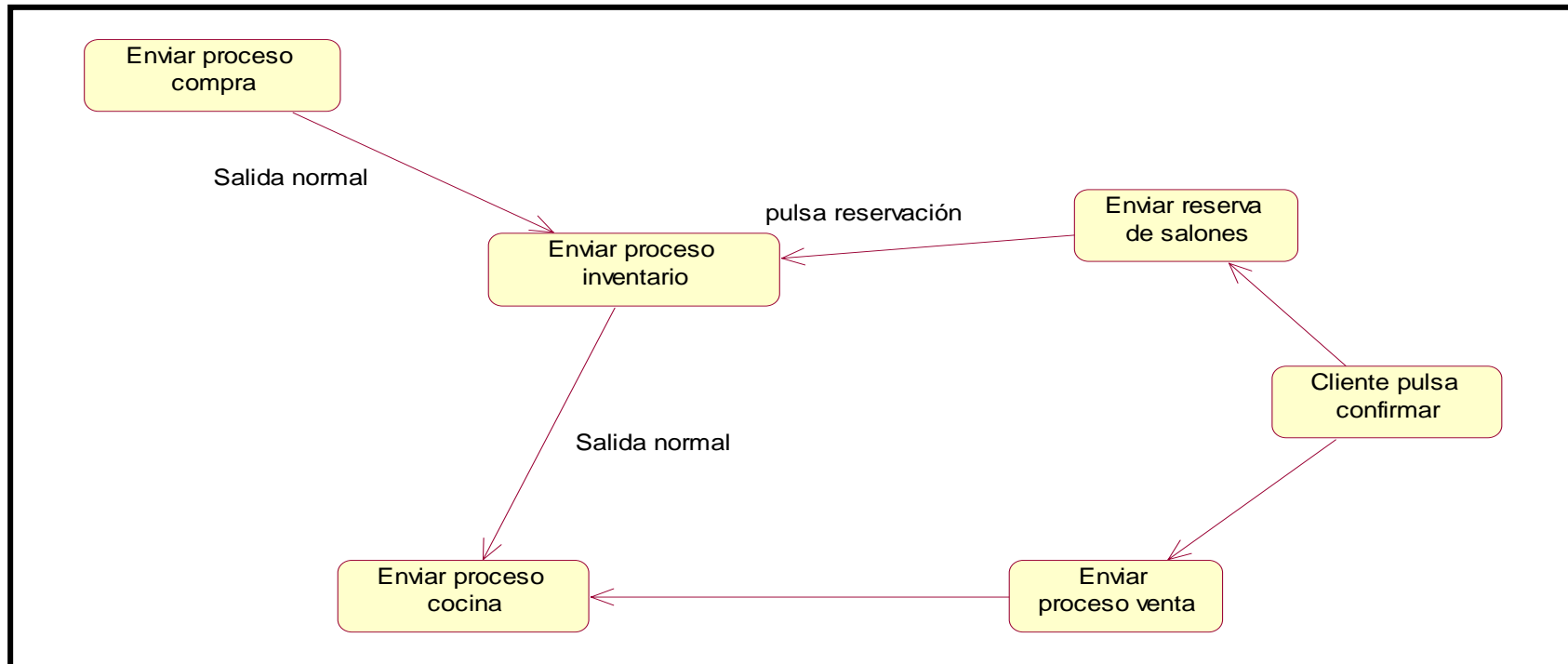


Figura No. 46 (Estado Diagrama General)

En este caso, vemos en modo general, cómo interactúan los diferentes procesos que posee el sistema. El proceso de compra tiene una entrada al proceso inventario ya que, luego de realizar una compra, ésta se debe ingresar al inventario. Del proceso inventario se produce una salida de productos a cocina y a los diversos departamentos. Tenemos el proceso reserva de salones y ventas que interactúan con el cliente ya que, ellos realizan una venta o una reserva y, estos procesos se ejecutan.



### 4.3.2.2 DIAGRAMA INVENTARIO

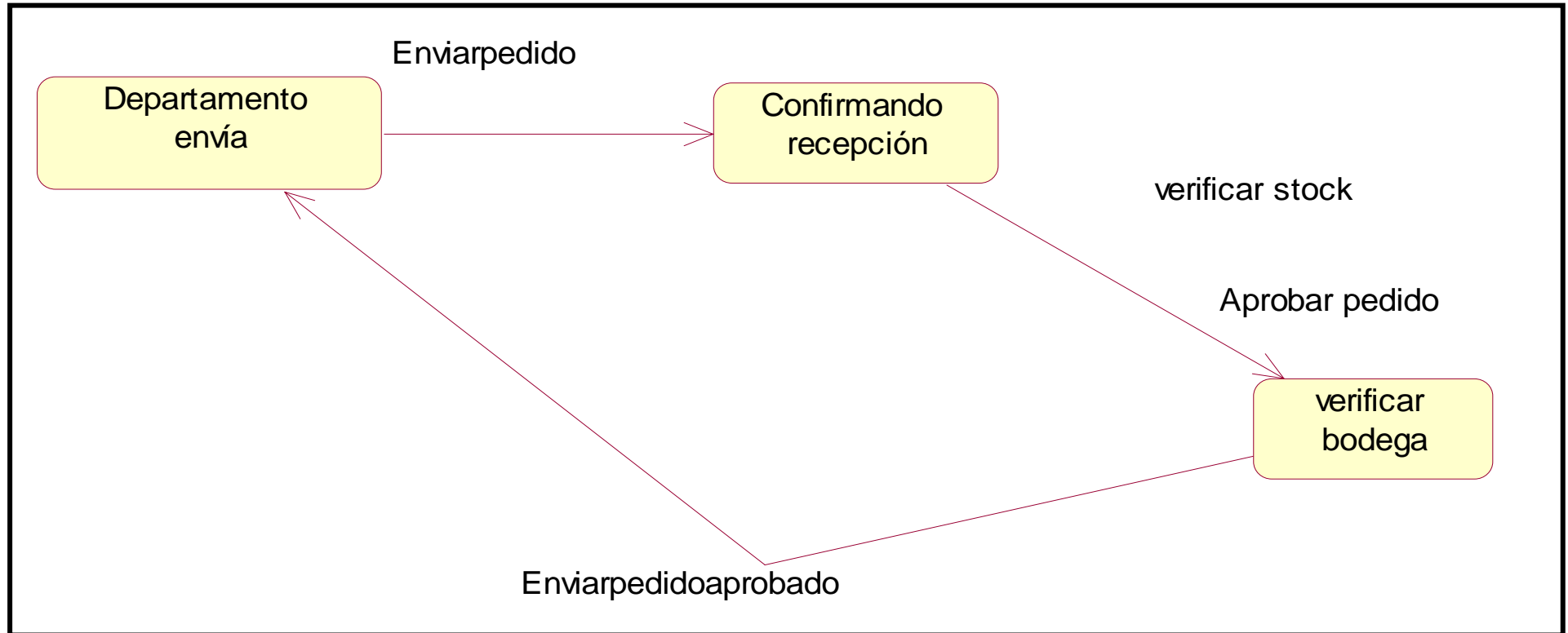


Figura No. 47 (Estado Diagrama Inventario)

Proceso inventario tiene como entradas los pedidos de cada departamento, estos pedidos se deben procesar o verificar los diversos pedidos que se han realizado en el mes; después de esto se debe confirmar el pedido. Una vez aceptado se envía a cada departamento.



### 4.3.2.3 DIAGRAMA VENTAS

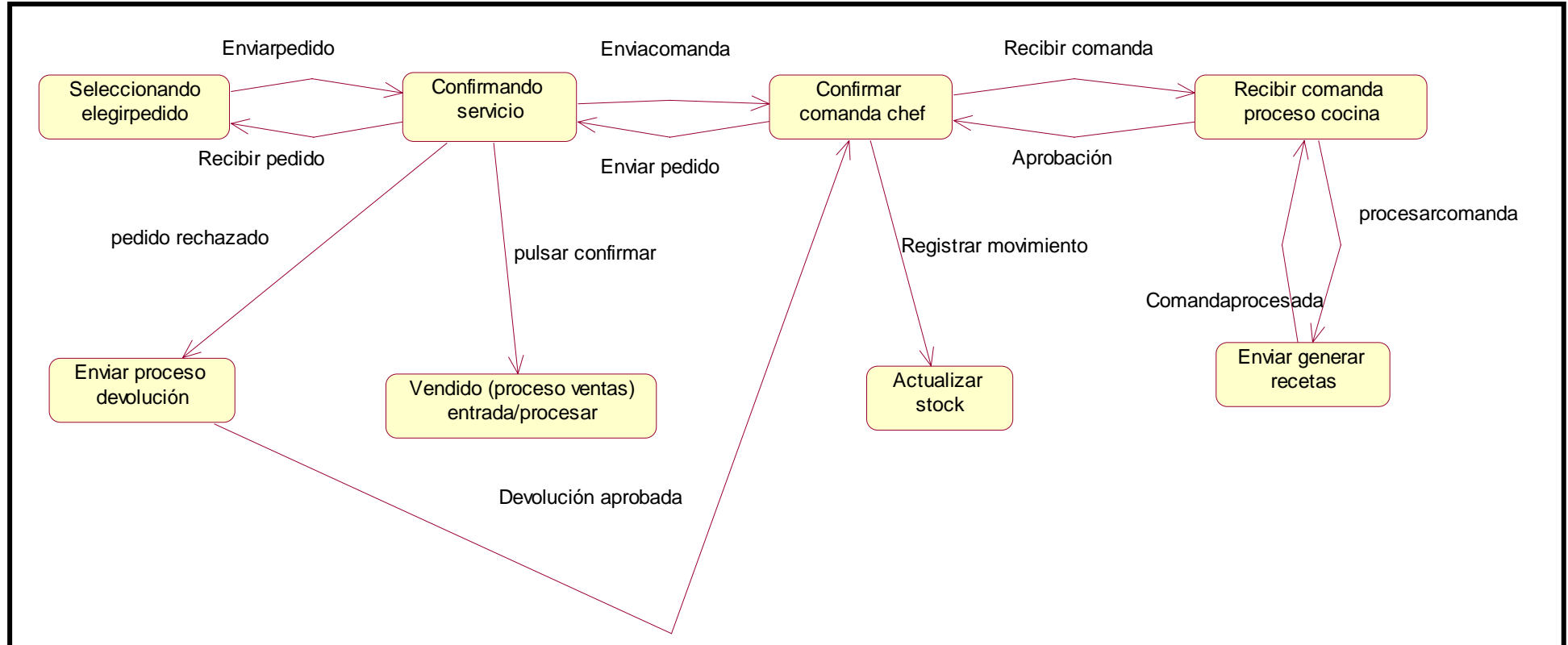


Figura No. 48 (Estado Diagrama Ventas)

En este proceso se interactúa con el cliente. Este realiza un pedido, el cual es enviado al chef; este interactúa con el proceso cocina, el cual procesa la receta. Una vez terminados estos procesos el chef da su aprobación y es entregado al cliente. Después de la aceptación del cliente, se emite una factura y se terminó la venta. En caso de que se realice una devolución del pedido, se debe enviar al chef para que apruebe y modifique el stock



#### 4.3.2.4 DIAGRAMA DEVOLUCION

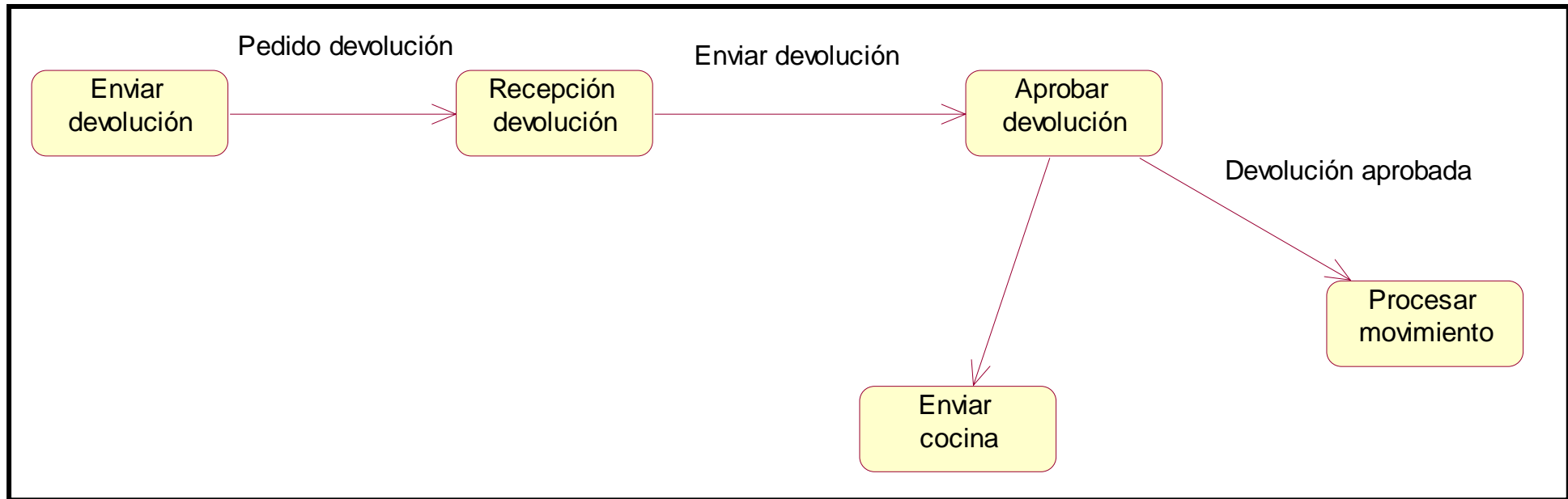


Figura No. 49 (Estado Diagrama Devolución)

En el proceso devolución se recibe la devolución, la cual es aprobada por el chef; este envía dicha devolución a la cocina y modifica el inventario.



### 4.3.2.5 DIAGRAMA GENERA RECETA

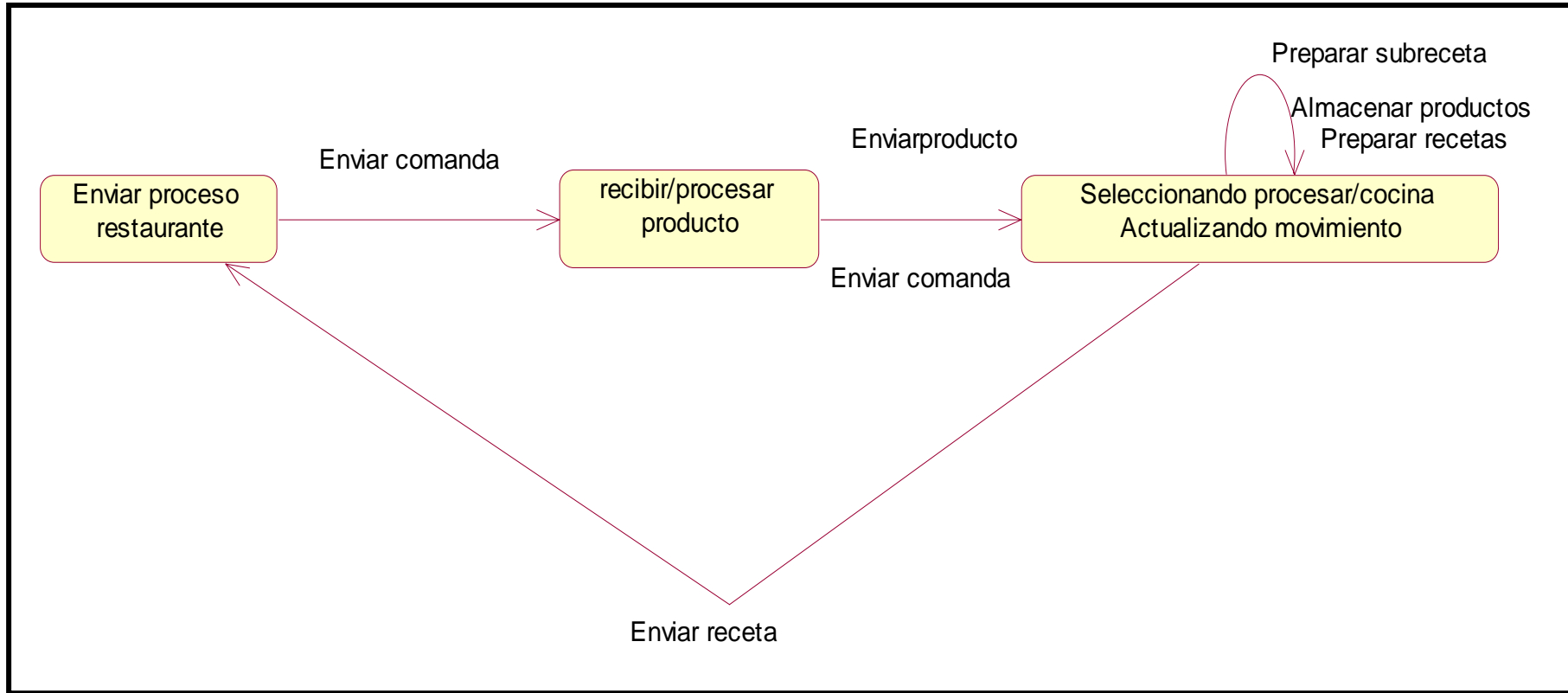


Figura No. 50 (Estado Diagrama Genera Receta)

En el proceso genera receta, primero se ingresan y procesan los productos recibidos. Una vez seleccionados, se realiza la preparación de subrecetas, se almacena los productos y se preparan recetas.



#### 4.3.2.6 DIAGRAMA COCINA

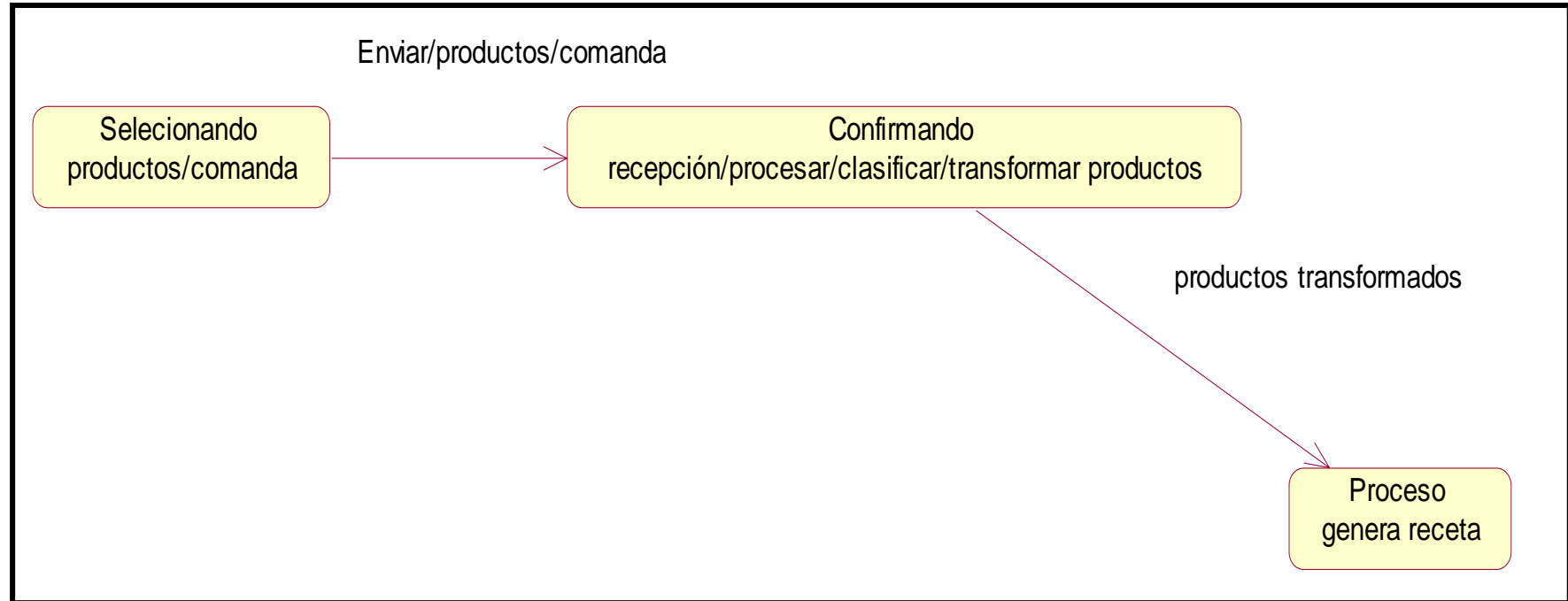


Figura No.51 (Estado Diagrama Cocina)

En el proceso cocina se seleccionan los productos y se recibe comanda. Una vez recibidos, se clasifican y transforman; con esto, se envía al proceso genera receta para su término.



### 4.3.2.7 DIAGRAMA COMPRAS

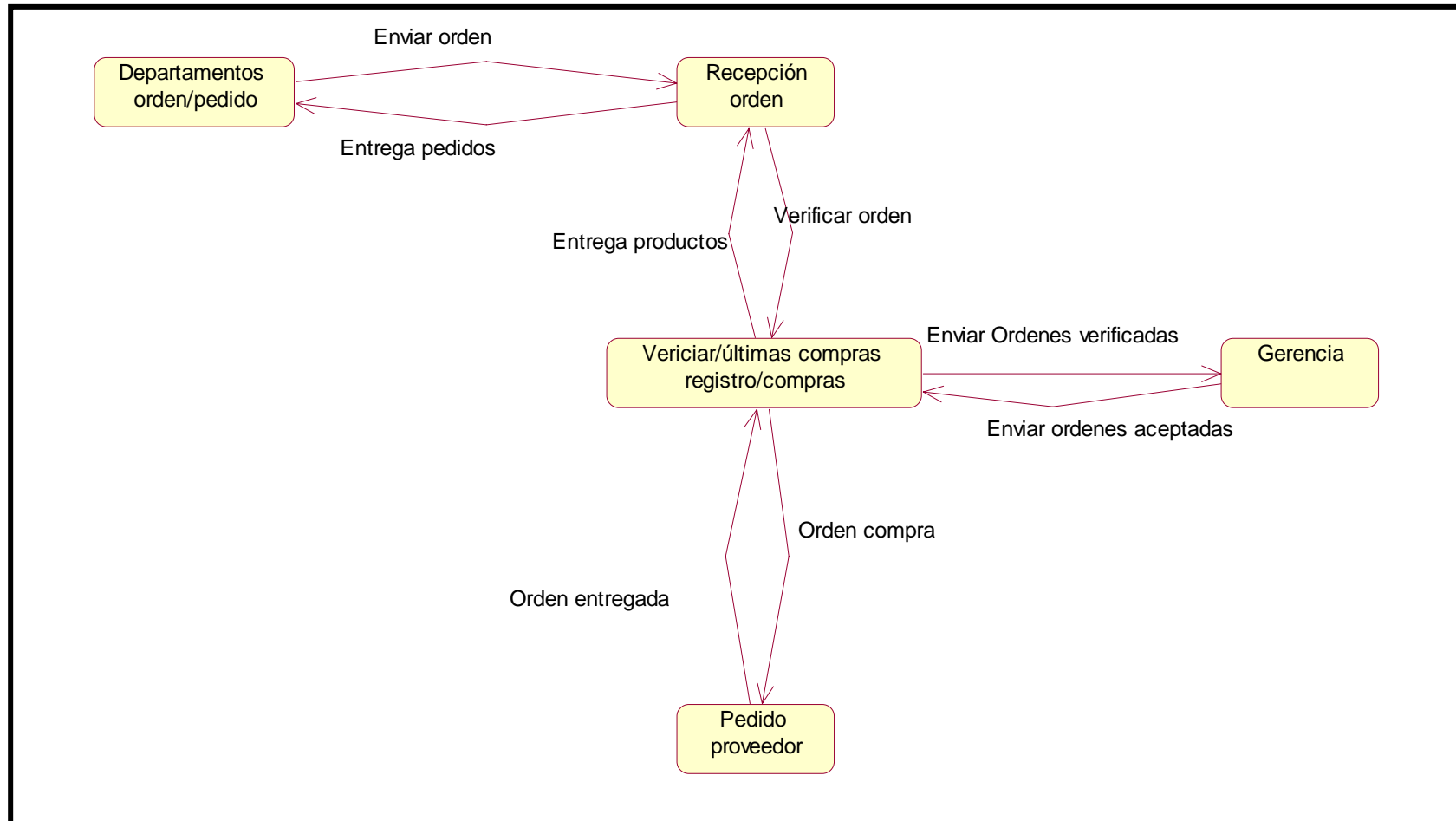


Figura No.52 (Estado Diagrama Compras)

Cada departamento hace una orden de pedido. El proceso compras realiza una recepción de estos pedidos, verifica si no ha existido compras anteriores, envía estas órdenes a gerencia para que sean aprobadas, caso de no haber tenido compras anteriores realiza un pedido al proveedor. Se deberá realizar un pedido a proveedor, el cual entrega y este proceso se registra. Terminando se entrega cada orden a los diversos departamentos.





### 4.3.2.8 DIAGRAMA RESERVA SALONES

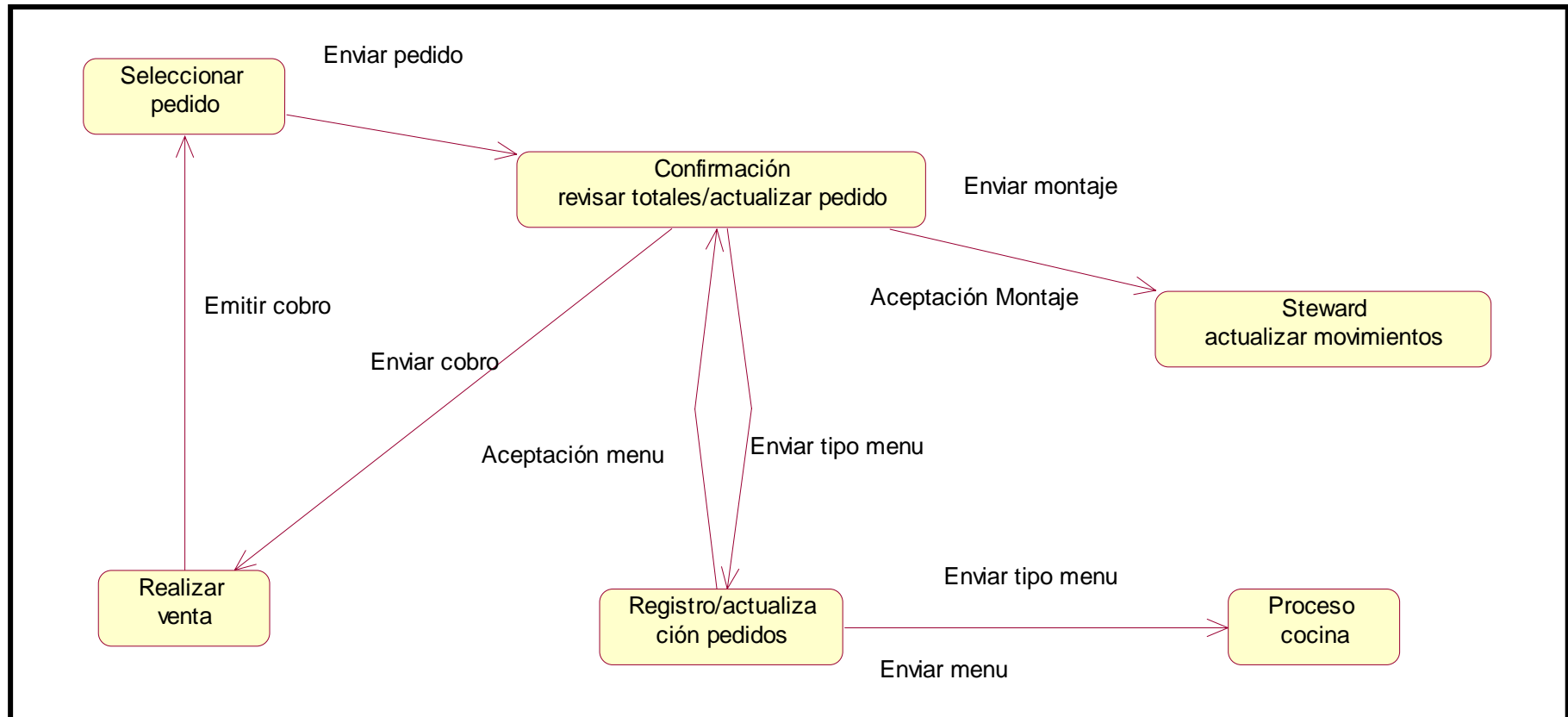


Figura No.53 (Estado Diagrama Reserva salones)

El cliente selecciona el tipo de salón y montaje que desea para su reservación, aquí se realiza un cronograma de actividades de cada salón, se confirma la reservación y se envía a Steward y cocina los tipos de menús y montajes que se deben colocar en el salón, actualizan movimiento y se envía al proceso de venta.



## **4.4 MODELAMIENTO FUNCIONAL**

Expresa qué valores de salida se derivan de qué valores de entrada. Consta de múltiples DFD's que muestran el flujo de valores desde las entradas externas, pasando por las operaciones y almacenes internos, hasta las salidas externas.

### **4.4.1 IDENTIFICACIÓN DE VALORES DE ENTRADA Y DE SALIDA**



### 4.4.1.1 DIAGRAMA GENERAL

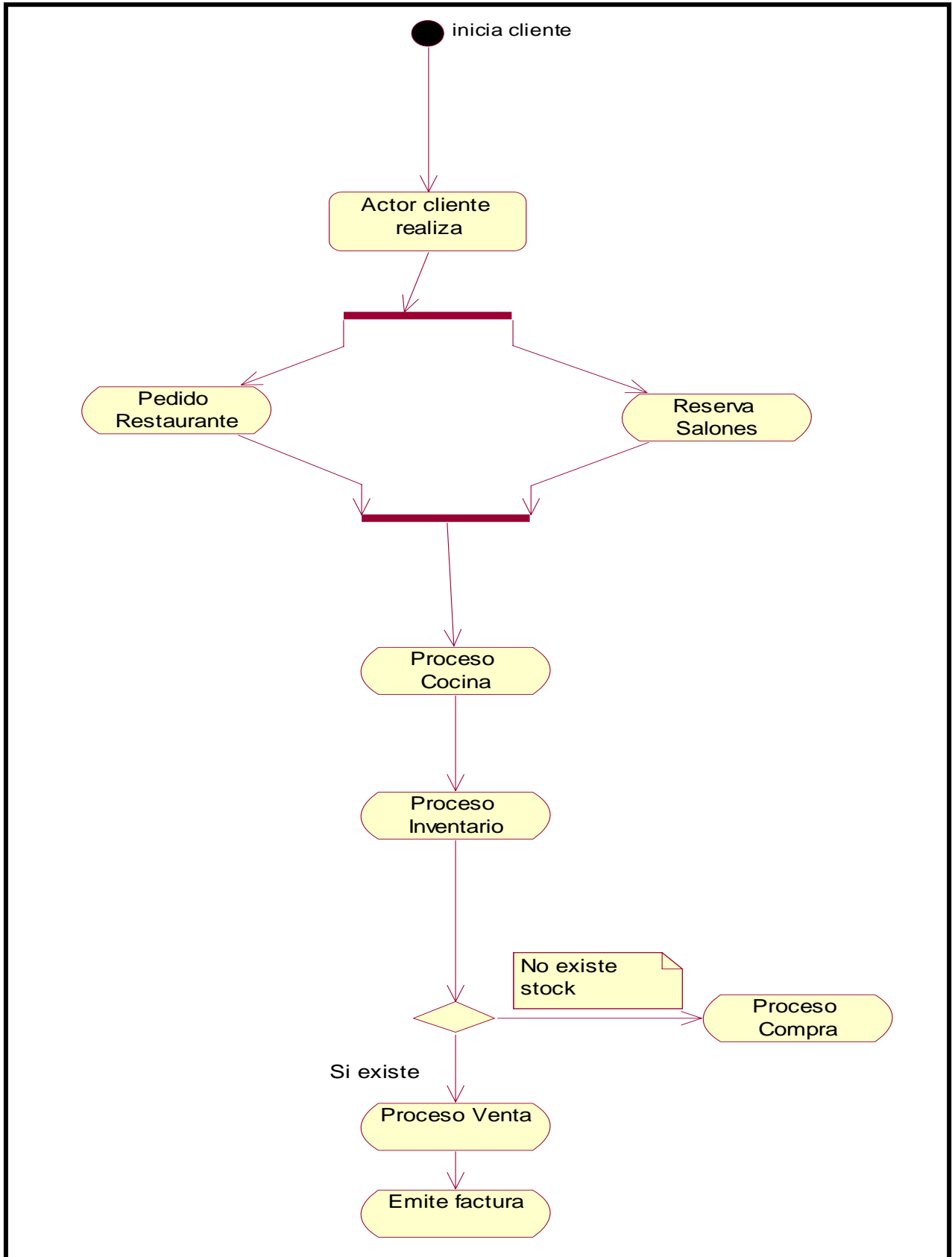


Figura No.54 (Funcional Diagrama General)



### 4.4.1.2 DIAGRAMA INVENTARIO

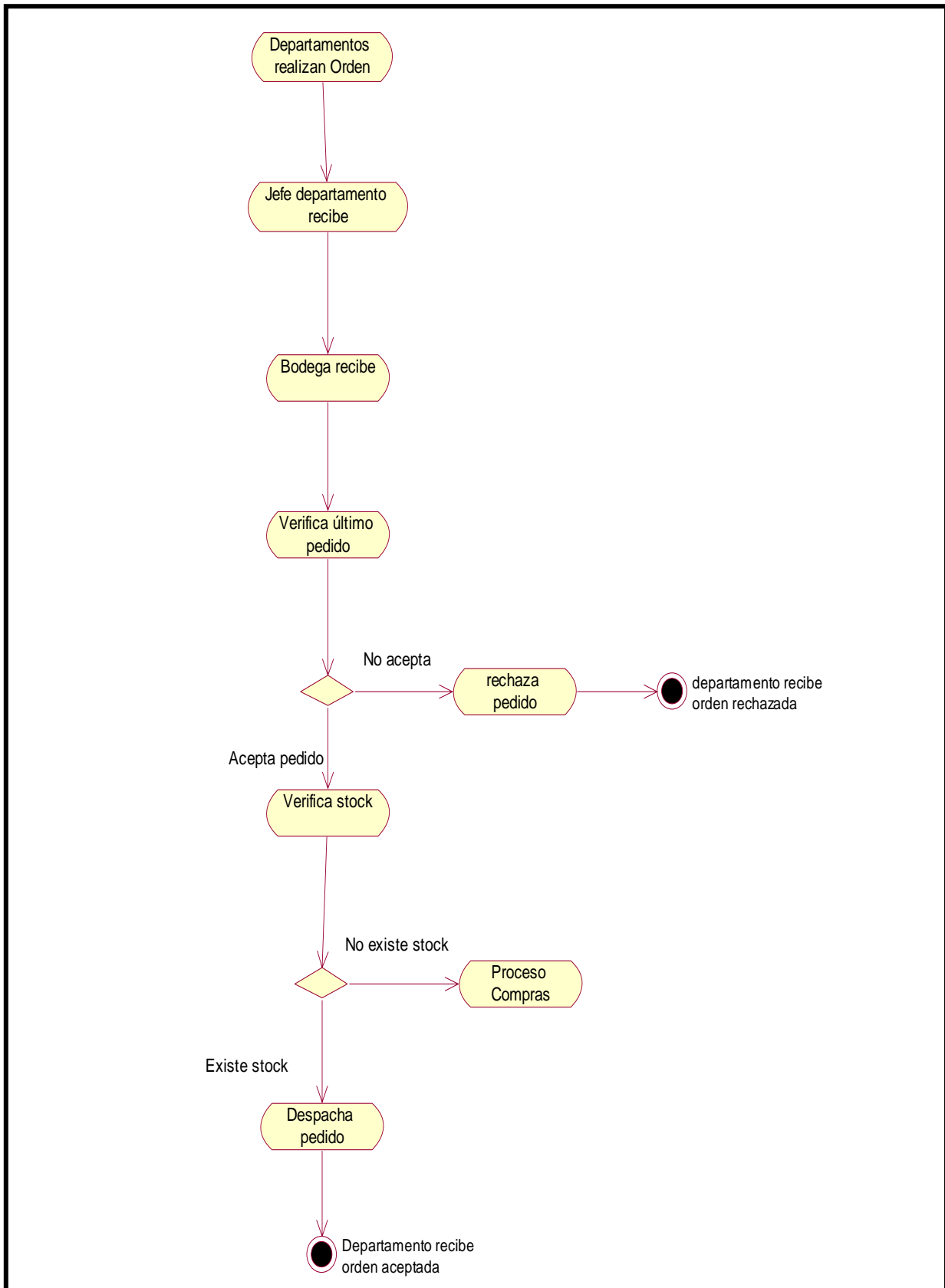


Figura No.55 (Funcional Diagrama Inventario)



### 4.4.1.3 DIAGRAMA RESERVA SALONES

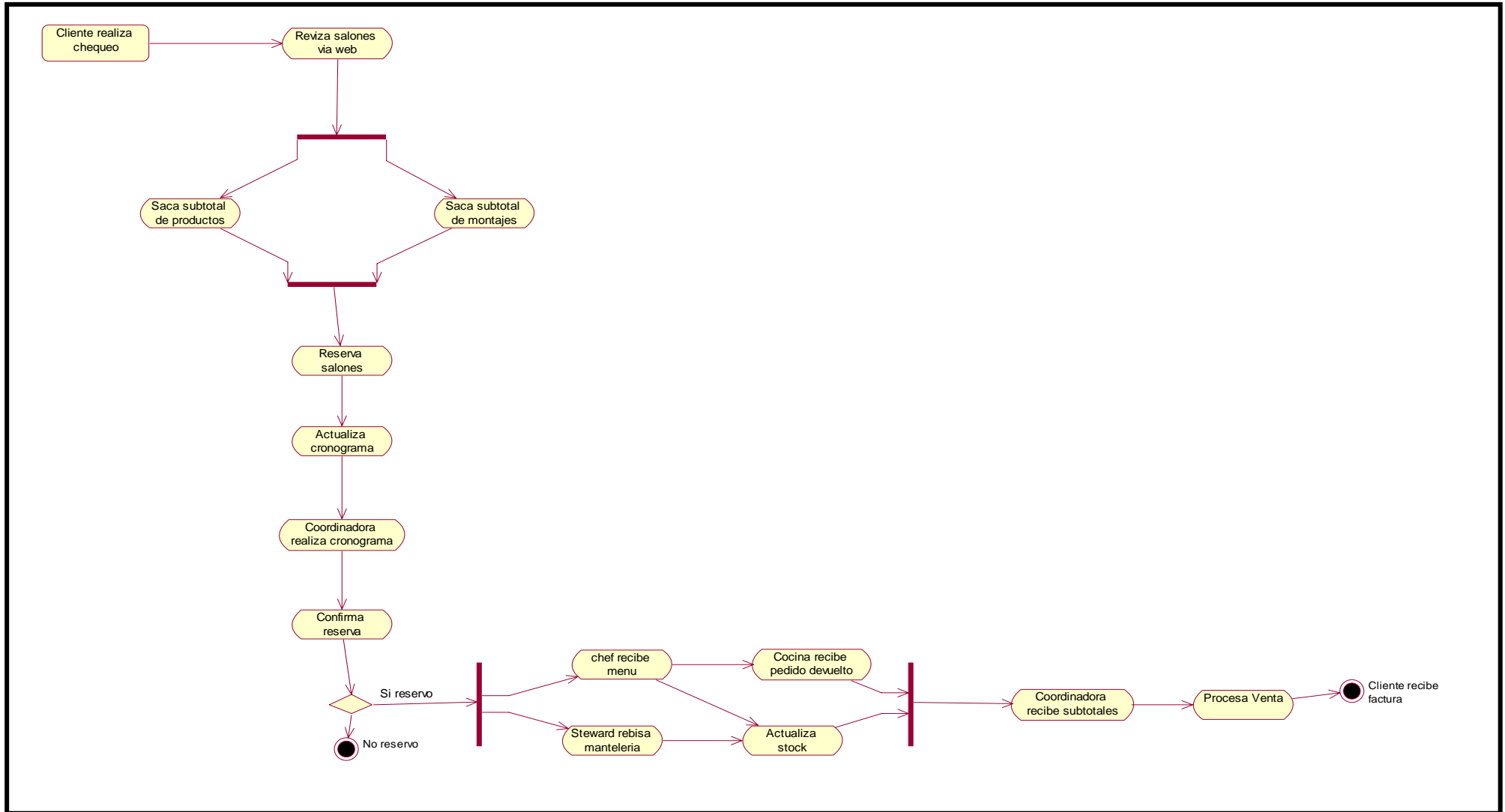


Figura No.56 (Funcional Diagrama Reserva Salones)



#### 4.4.1.4 DIAGRAMA GENERA RECETA

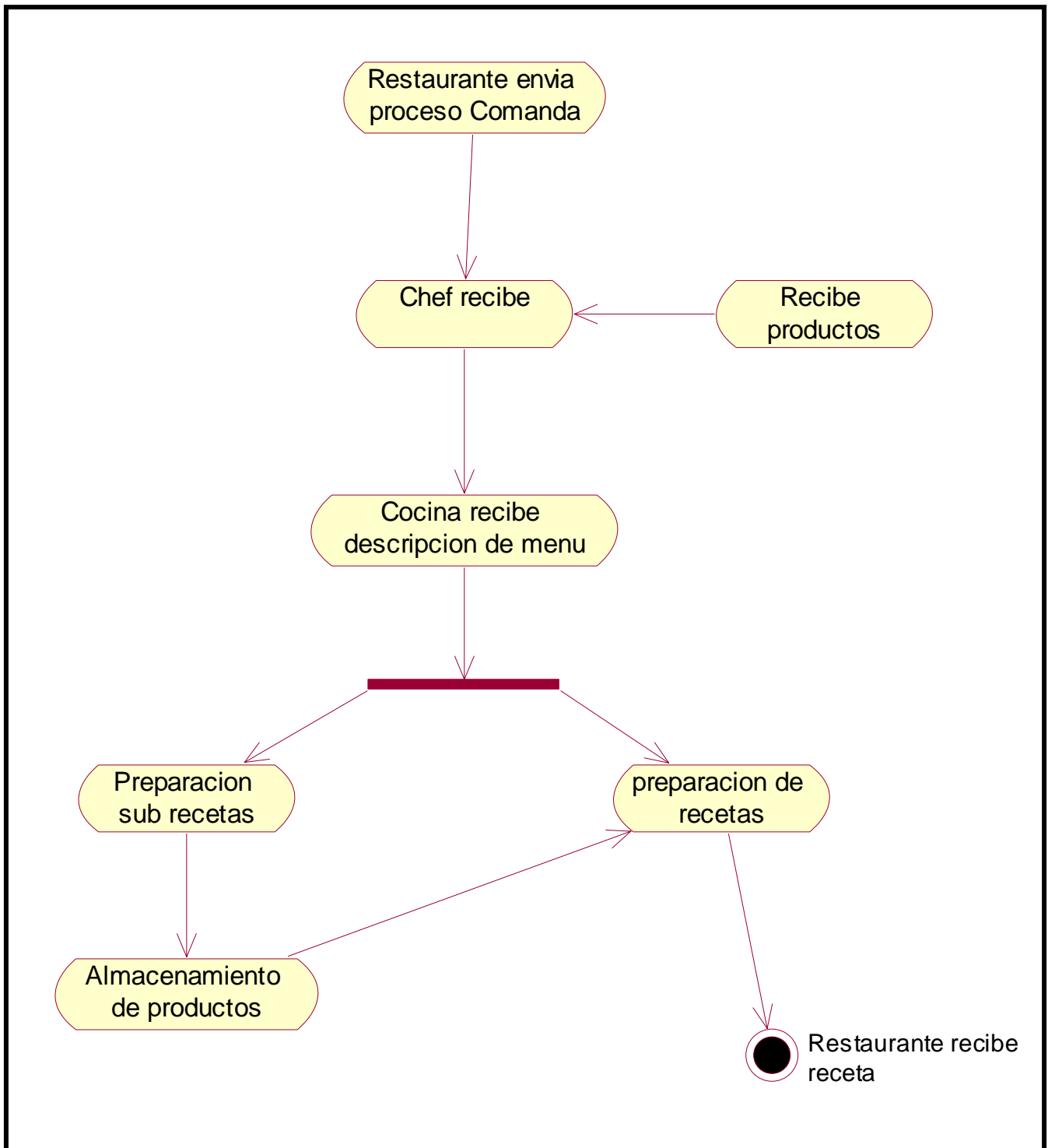


Figura No.57 (Funcional Diagrama Genera Receta)



#### 4.4.1.5 DIAGRAMA COCINA

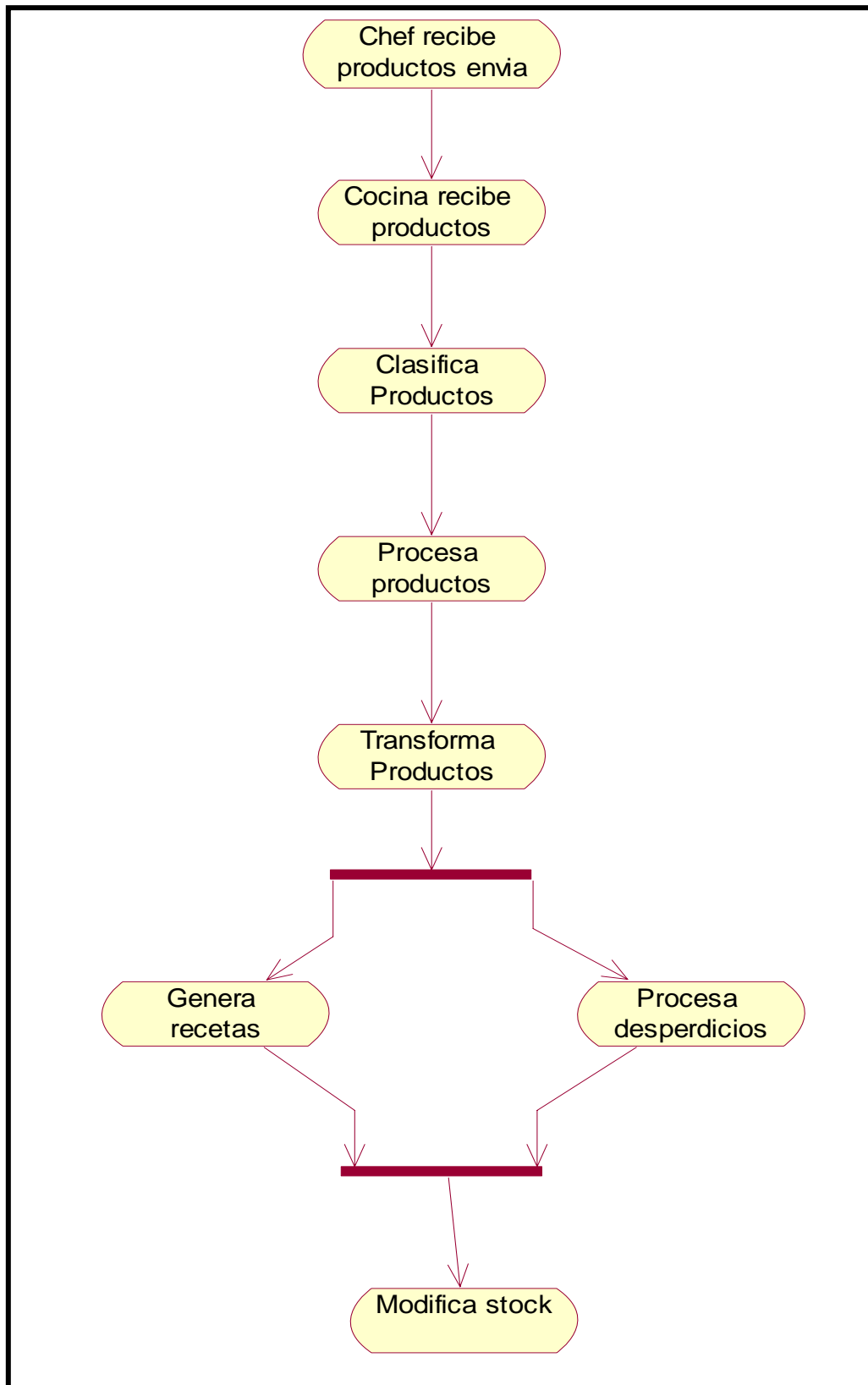


Figura No.58 (Funcional Diagrama Cocina)



### 4.4.1.6 DIAGRAMA COMPRAS

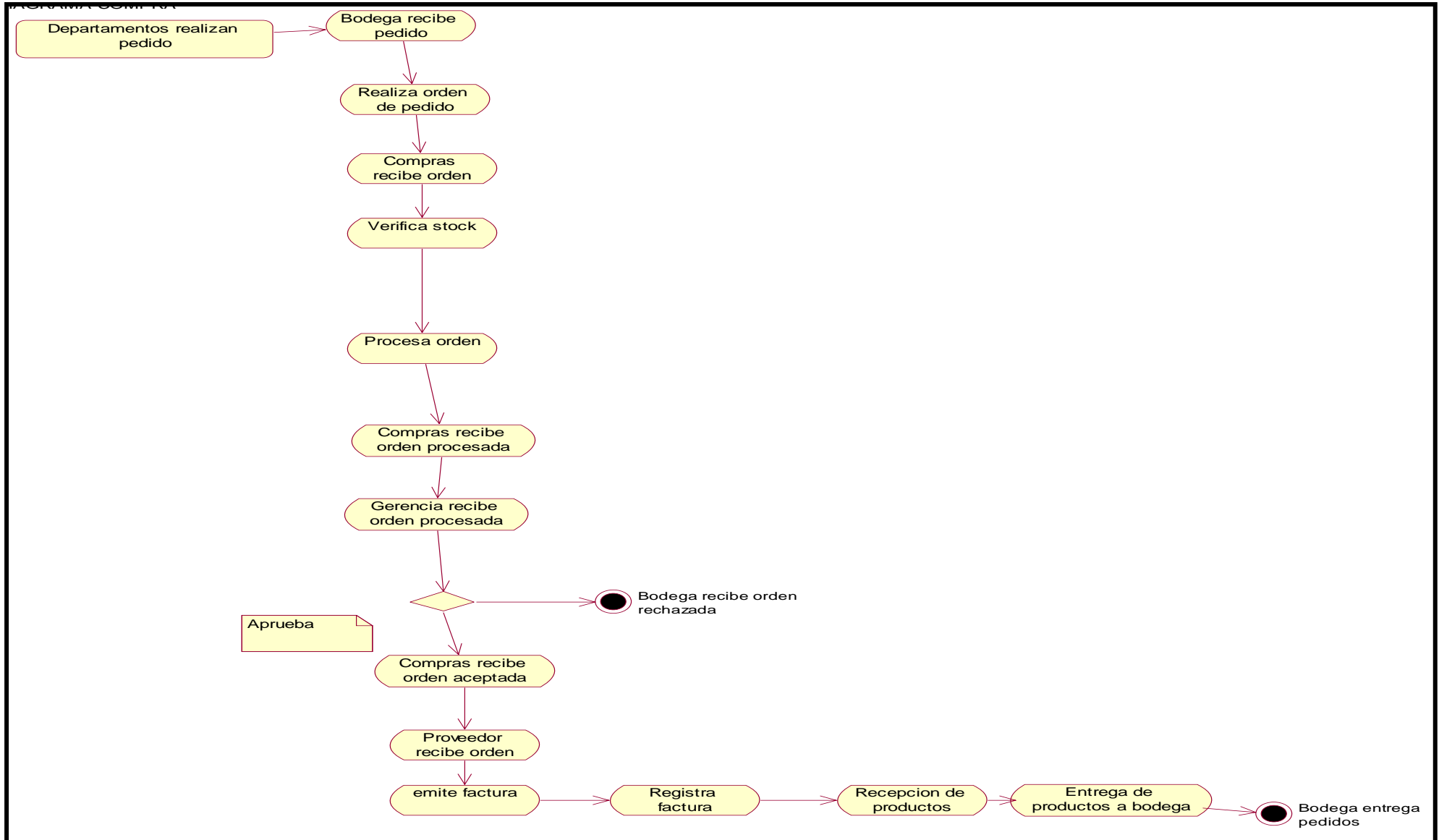


Figura No.59 (Funcional Diagrama Compras)





#### 4.4.1.7 DIAGRAMA DEVOLUCION

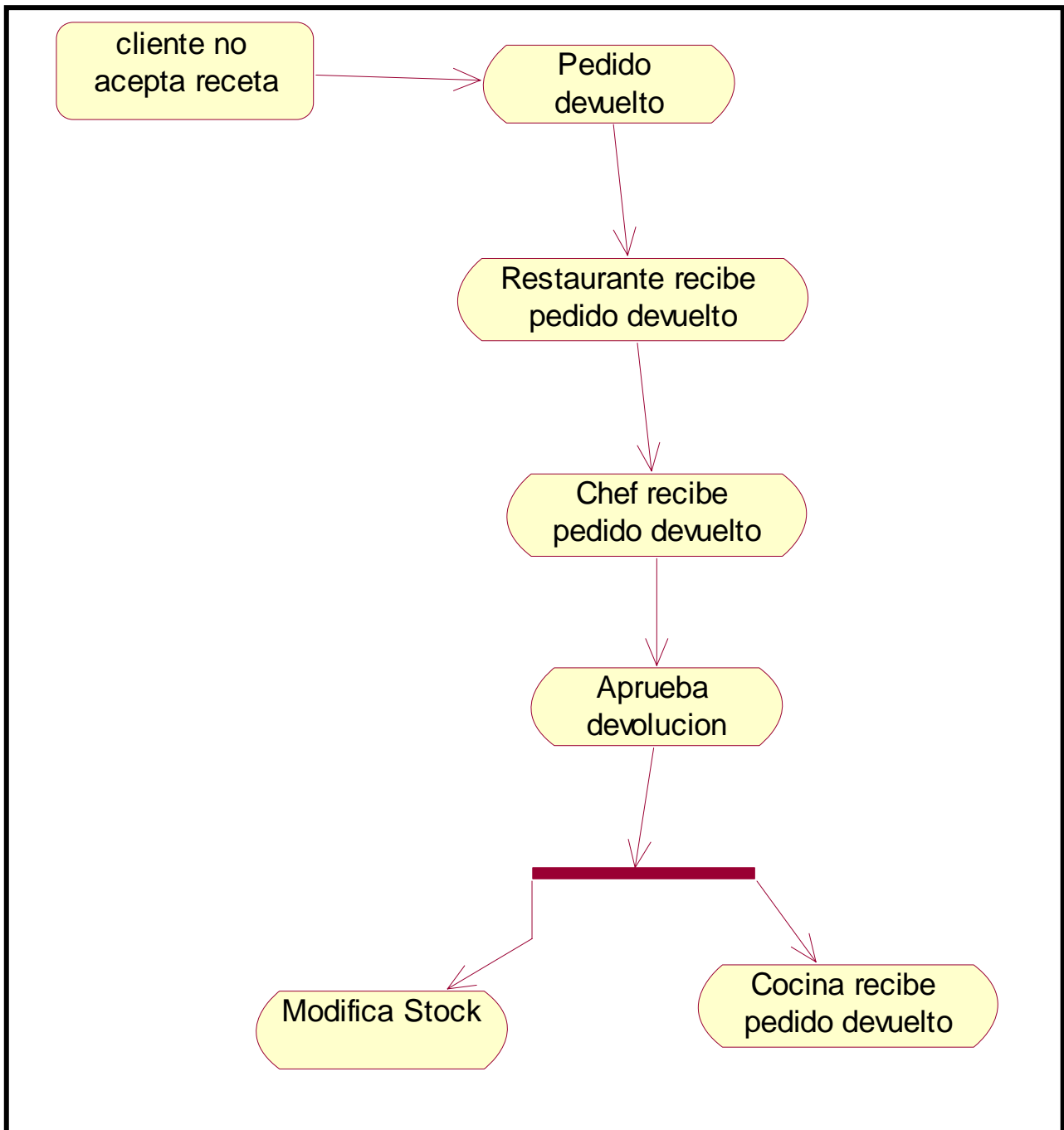


Figura No.60 (Funcional Diagrama Devolución)



### 4.4.1.7 DIAGRAMA VENTAS

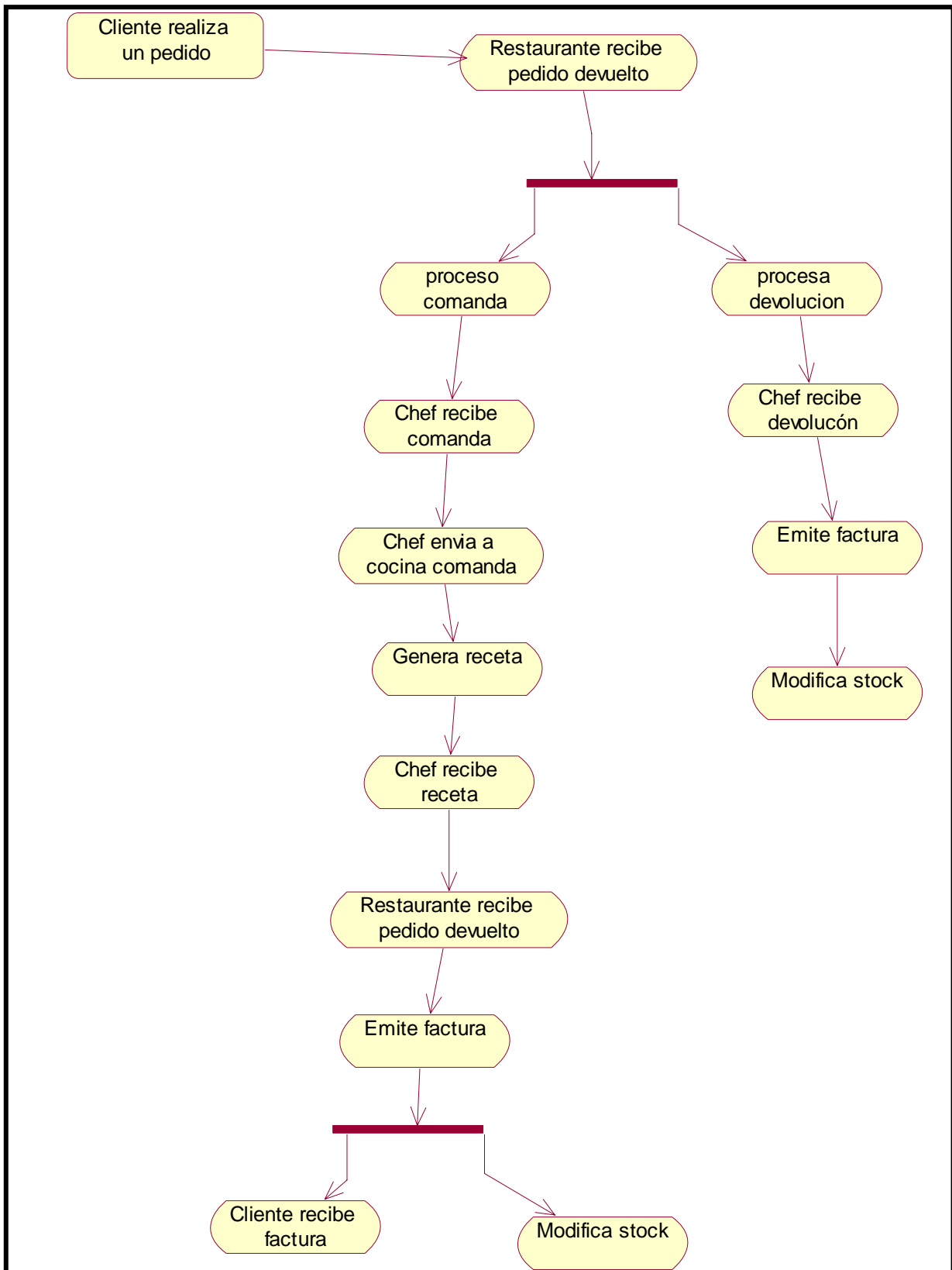


Figura No.61 (Funcional Diagrama Ventas)



## 4.5 ESPECIFICACIÓN DE CRITERIOS DE OPTIMIZACIÓN

### 4.5.1 DIAGRAMA GENERAL

Tabla 4.9 (Criterio de Optimización Diagrama General)

<b>OPERACIÓN:</b>
Explosión de los diferentes procesos
<b>RESPONSABILIDADES:</b>
Generar los enlaces para cada proceso
<b>ENTRADAS:</b>
Donde comenzar los diferentes procesos.
<b>SALIDAS:</b>
Número de enlaces generados o error si no se cumplen las condiciones que fueron impuestas por el usuario.
<b>OBJETOS MODIFICADOS:</b>
Las pantallas de mantenimiento
<b>PRECONDICIONES:</b>
El usuario ha seleccionado alguna pantalla
<b>POSTCONDICIONES:</b>
Verificación de las pantallas



### 4.5.2 DIAGRAMA INVENTARIO

Tabla 4.10 (Criterio de Optimización Diagrama Inventario)

<b>OPERACIÓN:</b>
Creaciones de productos y pedidos
<b>RESPONSABILIDADES:</b>
Generar nuevos productos y pedidos
<b>ENTRADAS:</b>
Uno o varios pedidos.
<b>SALIDAS:</b>
Número de pedidos aceptados.
<b>OBJETOS MODIFICADOS:</b>
Movimientos, pedidos aceptados y rechazados
<b>PRECONDICIONES:</b>
El usuario selecciona opciones de pedido
<b>POSTCONDICIONES:</b>
Tener pedidos de los departamentos.



### 4.5.3 DIAGRAMA RESERVA SALONES

Tabla 4.11 (Criterio de Optimización Diagrama Reserva Salones)

<b>OPERACIÓN:</b>
El cliente realiza una reservación vía web o en el hotel
<b>RESPONSABILIDADES:</b>
Generar montajes y menú que el cliente eligió
<b>ENTRADAS:</b>
Montajes, menú y costos de la factura
<b>SALIDAS:</b>
Emisión de las facturas
<b>OBJETOS MODIFICADOS:</b>
Movimientos de inventarios
<b>PRECONDICIONES:</b>
El usuario ha seleccionado salón, montajes y menú
<b>POSTCONDICIONES:</b>
Elección de salón, montajes y menú.



#### 4.5.4 DIAGRAMA GENERA RECETA

Tabla 4.12 (Criterio de Optimización Diagrama Genera Receta)

<b>OPERACIÓN:</b>
Generación de recetas para el hotel
<b>RESPONSABILIDADES:</b>
Generar nuevas recetas para el hotel
<b>ENTRADAS:</b>
La comanda que envía el Chef
<b>SALIDAS:</b>
La generación de la receta
<b>OBJETOS MODIFICADOS:</b>
Modificación del inventario
<b>PRECONDICIONES:</b>
El usuario ha seleccionado un tipo de plato
<b>POSTCONDICIONES:</b>
Listas de búsqueda de los diferentes platos.



### 4.5.5 DIAGRAMA COCINA

Tabla 4.13 (Criterio de Optimización Diagrama Cocina)

<b>OPERACIÓN:</b>
Ingreso de productos y creación de platos
<b>RESPONSABILIDADES:</b>
Clasificación, procesamiento, transformación de los productos
<b>ENTRADAS:</b>
Ingreso de productos e ingreso de comanda.
<b>SALIDAS:</b>
Transformación de los productos y preparación de comanda
<b>OBJETOS MODIFICADOS:</b>
Actualización de inventario
<b>PRECONDICIONES:</b>
Ingreso de productos, comanda
<b>POSTCONDICIONES:</b>
Pedido de productos, comanda



### 4.5.6 DIAGRAMA COMPRAS

Tabla 4.14 (Criterio de Optimización Diagrama Compras)

<b>OPERACIÓN:</b>
Clasificación de pedidos y realización de compras
<b>RESPONSABILIDADES:</b>
Generar la aceptación de los pedidos y compras a los proveedores
<b>ENTRADAS:</b>
Ingreso de los pedidos, productos
<b>SALIDAS:</b>
Salida de pedidos a los diferentes departamentos
<b>OBJETOS MODIFICADOS:</b>
Modificación del inventario
<b>PRECONDICIONES:</b>
Realización de pedidos
<b>POSTCONDICIONES:</b>
Necesidades de cada departamento





### 4.5.7 DIAGRAMA DEVOLUCION

Tabla 4.15 (Criterio de Optimización Diagrama Devolución)

<b>OPERACIÓN:</b>
Ingreso de la devolución de una receta
<b>RESPONSABILIDADES:</b>
Generar las bajas de los productos
<b>ENTRADAS:</b>
Ingreso de la receta.
<b>SALIDAS:</b>
Receta devuelta
<b>OBJETOS MODIFICADOS:</b>
Modificación del inventario
<b>PRECONDICIONES:</b>
El usuario no acepta el plato recibido
<b>POSTCONDICIONES:</b>
El usuario realiza un pedido.



### 4.5.8 DIAGRAMA VENTAS

Tabla 4.16 (Criterio de Optimización Diagrama Ventas)

<b>OPERACIÓN:</b>
Realización de una venta
<b>RESPONSABILIDADES:</b>
Generar la factura y cobro de la venta
<b>ENTRADAS:</b>
Pedido de restaurante, habitación, reservación
<b>SALIDAS:</b>
Generación de la venta
<b>OBJETOS MODIFICADOS:</b>
Modificación del inventario
<b>PRECONDICIONES:</b>
Un cliente realiza un pedido o reservación
<b>POSTCONDICIONES:</b>
El cliente ingresa al hotel.