

TABLA DE CONTENIDO

C	DISEÑO DEL SOFTWARE DE LA APLICACIÓN MÓVIL	1
C.1	PAQUETES DE DISEÑO	1
C.2	DIAGRAMAS DE CLASE DE DISEÑO	3
C.2.1	Caso de uso ValidarAcceso	4
C.2.2	Caso de uso ModificarContraseña	9
C.2.2	Caso de uso RealizarPedido	14
C.2.3	Caso de uso BuscarPedido	30
C.3	DIAGRAMAS INTERACCIÓN DEL SISTEMA	37
C.3.1	Caso de uso ValidarAcceso	37
C.3.2	Caso de uso ModificarContraseña	38
C.3.3	Caso de uso RealizarPedido	40
C.3.3	Caso de uso BuscarPedido	48
C.4	DIAGRAMA DE IMPLANTACIÓN	50
C.5	DISEÑO DE LA BASE DE DATOS	51

TABLA DE FIGURAS

<i>Figura C-1. Paquetes de diseño</i> _____	1
<i>Figura C-2. Estereotipo de una Client page</i> _____	2
<i>Figura C-3. Estereotipo de una Server page</i> _____	2
<i>Figura C-4. Estereotipo de un Form</i> _____	3
<i>Figura C-5. Diagrama de clases Validar Acceso, (Estereotipos)</i> _____	4
<i>Figura C-6. Diagrama de clases Validar Acceso</i> _____	5
<i>Figura C-7. Diagrama de clases ModificarContraseña, (Estereotipos)</i> _____	9
<i>Figura C-8. Diagrama de clases Modificar contraseña</i> _____	10
<i>Figura C-9. Diagrama de clases caso de uso RealizarPedido</i> _____	14
<i>Figura C-10. Diagrama de clases RealizarPedido, (Adicionar producto al carrito de compras)</i> _____	18
<i>Figura C-11. Diagrama de clases RealizarPedido, (Enviar Pedido)</i> _____	19
<i>Figura C-12. Diagrama de clases caso de uso RealizarPedido, (Eliminar producto del carrito de compras)</i> _____	20
<i>Figura C-13. Diagrama de clases Caso de uso RealizarPedido, (Cambiar cantidad de producto a pedir)</i> _____	21
<i>Figura C-14. Diagrama de clases, caso de uso BuscarPedido</i> _____	30
<i>Figura C-15. Diagrama de clases, caso de uso BuscarPedido, (búsqueda de pedido por un cliente)</i> _____	32
<i>Figura C-16. Diagrama de clases, Caso de uso BuscarPedido, (búsqueda por agente)</i> ____	33
<i>Figura C-17. Carta de secuencias de mensajes Validar acceso Cliente.</i> _____	37
<i>Figura C-18. Carta de secuencia de mensajes Validar Acceso Agente</i> _____	38
<i>Figura C-19. Carta de secuencia de mensajes Modificar Contraseña Cliente</i> _____	39
<i>Figura C-20. Carta de secuencia de mensajes Modificar Contraseña Agente</i> _____	39
<i>Figura C-21. Carta de secuencia de mensajes Realizar Pedido, Adición de productos al carrito de compras por un Cliente</i> _____	41

<i>Figura C-22. Carta de secuencia de mensajes Realizar Pedido, Adición de productos al carrito de compras por un Agente</i>	42
<i>Figura C-23. Carta de secuencia de mensajes Realizar Pedido, enviar pedido por un Cliente</i>	43
<i>Figura C-24. Carta de secuencia de mensajes Realizar Pedido, enviar pedido por un Agente</i>	44
<i>Figura C-25. Carta de secuencia de mensajes Realizar Pedido, eliminar productos del carrito de compras por un Cliente</i>	45
<i>Figura C-26. Carta de secuencia de mensajes Realizar Pedido, eliminar productos del carrito de compras por un Agente</i>	46
<i>Figura C-27. Carta de secuencia de mensajes Realizar Pedido, modificar cantidad de un producto Cliente/Agente</i>	47
<i>Figura C-28. Carta de secuencia de mensajes Buscar Pedido, Buscar pedido por un Cliente</i>	48
<i>Figura C-29. Carta de secuencia de mensajes Buscar Pedido, Buscar pedido por un Agente</i>	49
<i>Figura C-30. Diagrama de implantación</i>	50
<i>Figura C-31. Diagrama del paquete de datos, "Análisis del Software"</i>	52
<i>Figura C-32. Diagrama Entidad-Relación</i>	53
<i>Figura C-33. Diagrama Entidad-Relación, normalizado</i>	53

LISTA DE TABLAS

<i>Tabla C-1. Descripción técnica del R280d de Ericsson.</i>	51
<i>Tabla C-2. Tabla adminsx, (Administrador del sistema)</i>	54
<i>Tabla C-3. Tabla agente</i>	54
<i>Tabla C-4. Tabla Cliente</i>	55
<i>Tabla C-5. Tabla cliente_agente</i>	55
<i>Tabla C-6. Tabla cliente_empresa</i>	55
<i>Tabla C-7. Tabla empresa</i>	55
<i>Tabla C-8. Tabla pedido</i>	56
<i>Tabla C-9. Tabla producto</i>	56
<i>Tabla C-10. Tabla productopedido</i>	56
<i>Tabla C-11. Tabla transaccion</i>	56

C DISEÑO DEL SOFTWARE DE LA APLICACIÓN MÓVIL

C.1 PAQUETES DE DISEÑO

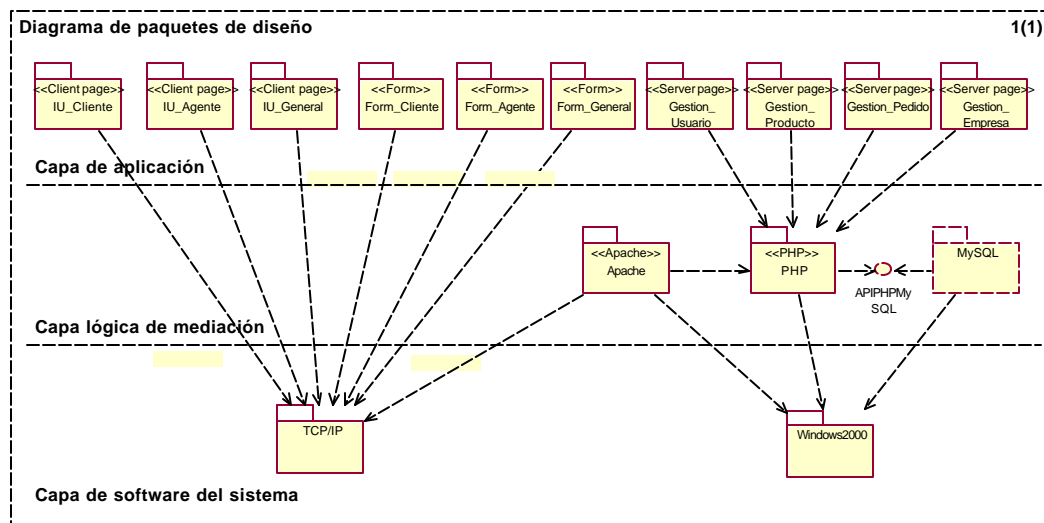


Figura C-1. Paquetes de diseño

De acuerdo al estándar de UML, en la Figura C-1, se muestran los paquetes utilizados para estructurar los elementos del modelo de diseño. A continuación se describen los estereotipos a los cuales pertenecen las clases contenidas en cada paquete.

Client page: Una página cliente, (*Cliente page*) es un documento HTML distribuido por un servidor web, utilizando el protocolo http; los navegadores, (clientes) decodifican el documento y presentan la información contenida el usuario.

La petición de una página cliente se realiza, enviando al servidor web, la dirección de la página, (URL, *Unified Resource Locator*), valiéndose, también, del protocolo http.

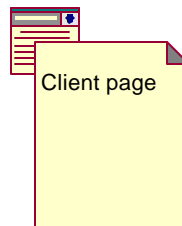


Figura C-2. Estereotipo de una Client page

Server page: Una Server page, (página servidora), se define como una secuencia de comandos procesados por el servidor, los cuales son ejecutados cuando el navegador, (cliente), solicita una URL. Mediante la ejecución de estas instrucciones, se construye una página cliente de acuerdo a información obtenida, generalmente, de una base de datos. Así un servidor está en capacidad de enviar páginas clientes construidas dinámicamente, como respuesta a peticiones de URLs de Server pages.

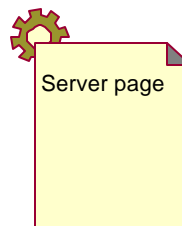


Figura C-3. Estereotipo de una Server page

Form: Se refiere a los formularios contenidos en una Client page, los cuales a su vez, contienen campos de entrada, tales como cajas de texto, botones y listas de selección, cajas de chequeo, etc. Cada formulario en una Client page, tiene asociado una Server page, la cual recibe y procesa la información enviada.

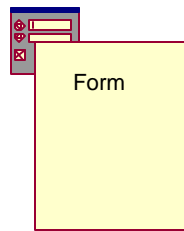


Figura C-4. Estereotipo de un Form

Apache: Es el servidor web, compatible con HTTP/1.1, el cual en conjunto con el módulo de php se utilizará como plataforma de soporte para el sistema.

PHP: Es un lenguaje scripting¹ con HTML embebido, este lenguaje, contiene un API que permite ejecutar funciones por medio de las cuales es posible realizar consultas a la base de datos del sistema, utilizando el gestor de la base de datos del sistema, (MySQL).

MySQL: Es un sistema de administración que permite, adicionar, acceder y procesar datos almacenados en una base de datos. También se podría definir como un sistema cliente/servidor que consiste en un servidor SQL multi-hilo que soporta diferentes backends, librerías y programas cliente, herramientas administrativas y algunas interfaces de programación. MySQL, será el sistema de gestión de la base de datos de este sistema.

C.2 DIAGRAMAS DE CLASE DE DISEÑO

Los estereotipos por medio de los cuales se definen la relación de asociación entre clases son los siguientes:

- **Build:** Esta relación indica la creación de una Client Page por una Server page, utilizando código PHP.
- **Link:** Simboliza la invocación de una Client page por otra Client page, a través de un hipervínculo o botón de selección.

¹ Lenguaje que indica al computador como ejecutar un procedimiento específico

- **Query:** Hace referencia a la necesidad de ejecutar operaciones que permitan realizar consultas a la base de datos.
- **Submit:** Simboliza la invocación de una Server page, por una Client page, por medio de un botón de selección o hipervínculo que implique el envío de parámetros incluidos en un Form, necesarios para la construcción de otra Client page.

C.2.1 Caso de uso ValidarAcceso

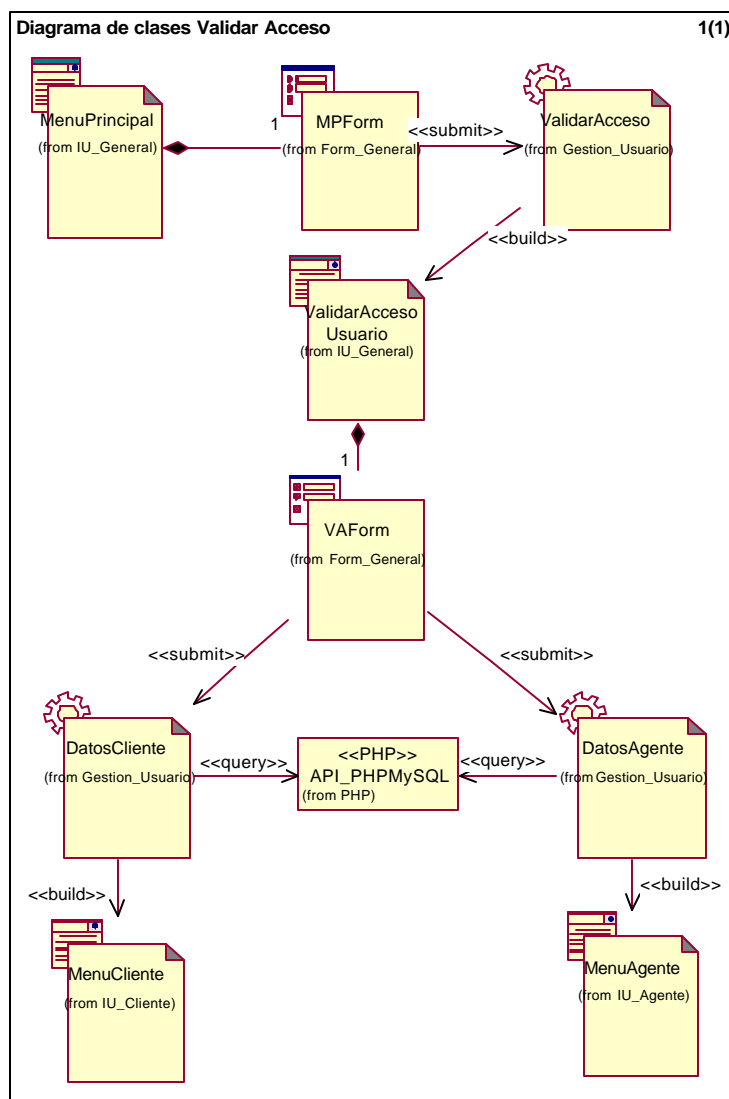


Figura C-5. Diagrama de clases Validar Acceso, (Estereotipos)

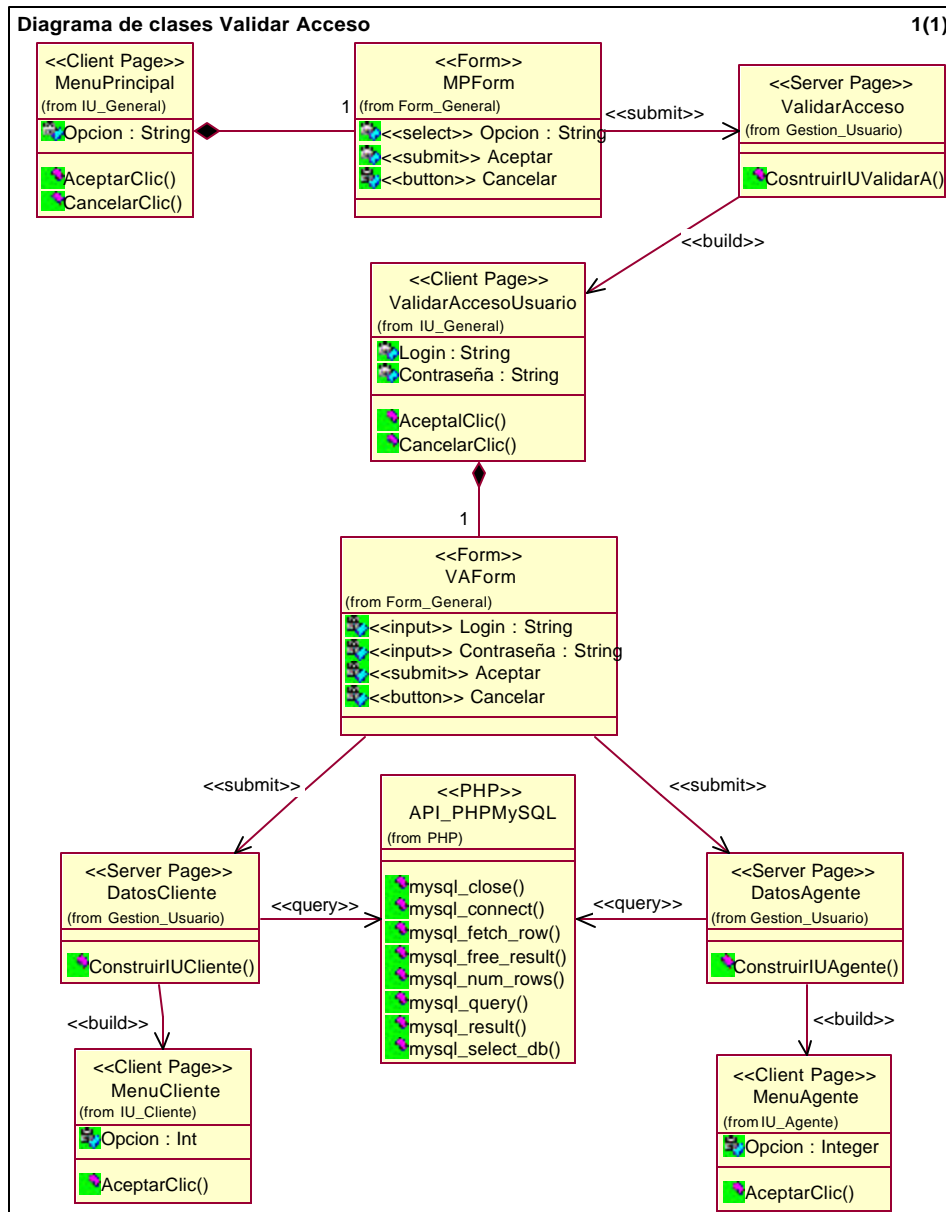


Figura C-6. Diagrama de clases Validar Acceso

Este caso de uso, el usuario invoca la Server page ValidarAcceso, realizando un submit valiéndose del formulario, MPForm, contenido en el MenuPrincipal. De esta forma la Client page ValidarAccesoUsuario, es construida.

Por medio del formulario contenido en la Client page, ValidarAccesoUsuario, el usuario puede enviar sus datos, (Nombre de usuario y contraseña), con el fin de ser identificado por el sistema. Estos datos, dependiendo del tipo de usuario, son

enviados a las Server pages, Datos cliente, (en el caso de un cliente) o Datos Agente, (para un agente), éstas tienen acceso a la base de datos utilizando el API de MySQL que proporciona PHP. Finalmente, de acuerdo al resultado obtenido de la comparación de los datos ingresados por el usuario y los obtenidos en la consulta, las Server pages construyen las Client pages, MenuCliente o MenuAgente.

```
Class MenuPrincipal {  
    Var opcion;  
    AceptarClic() {  
        Switch (estaclase->opcion) {  
            Caso "Ingresar":  
                Cargar ValidarAcceso();  
                break;  
            Caso "Ayuda":  
                Cargar Ayuda();  
                break;  
            Caso "Ver empresas":  
                MostrarEmpresas();  
                break;  
        }  
    }  
}
```

```
Class ValidarAcceso {  
    Var login;  
    Var contraseña  
    ConstruirIUValidarA()  
}
```

```
Class ValidarAccesoUsuario {  
    Var Usuario;  
    AceptarClic(ValidarAcceso->login, ValidarAcceso->contraseña) {
```

```
    estaclase->Usuario=InicialLogin; //Login=InicialLogin + Codigo usuario
                                //Inicial login es: A ,(para Agente) o C,
                                //( para Cliente)
    Si estaclase->Usuario=="A" {
        Cargar DatosAgente();
    }
    Si no Si Usuario=="C"{
        Cargar DatosCliente();
    }
    Si no {
        Desplegar mensaje "Nombre de usuario o contraseña incorreta";
        Cargar ValidarAcceso();
    }
}
CancelarClic() {
    Volver a MenuPrincipal;
}
}
```

Class DatosCliente {

```
    ConstruirIUCliente(ValidarAcceso->login,ValidarAcceso->contraseña){
        Conectar a la base de datos;
        consulta="Seleccionar en la base de datos todos los usuarios de la tabla
                cliente donde login='login' y password='contraseña'";
        resultado=Ejecutar consulta;
        Amacendar resultado;
        Numero= Número de filas en el resultado;
        Si Numero==1 {
            Cargar perfil de usuario;
        }
        Si no {
            Desplegar mensaje "Nombre de usr o contraseña incorrecta";
            Cargar ValidarAcceso();
        }
    }
}
```

}

Class DatosAgente {

```
    ConstruirIUCliente(ValidarAcceso->login, ValidarAcceso->contraseña){
        Conectar a la base de datos;
        consulta="Seleccionar en la base de datos todos los usuarios de la tabla
                agente donde login='login' y password='contraseña'";
        resultado=Ejecutar consulta;
        Amacenar resultado;
        Numero= Número de filas en el resultado;
        Si Numero==1 {
            Cargar perfil de usuario;
        }
        Si no {
            Desplegar mensaje "Nombre de usr o contraseña incorrecta";
            Cargar ValidarAcceso();
        }
    }
}
```

C.2.2 Caso de uso ModificarContraseña

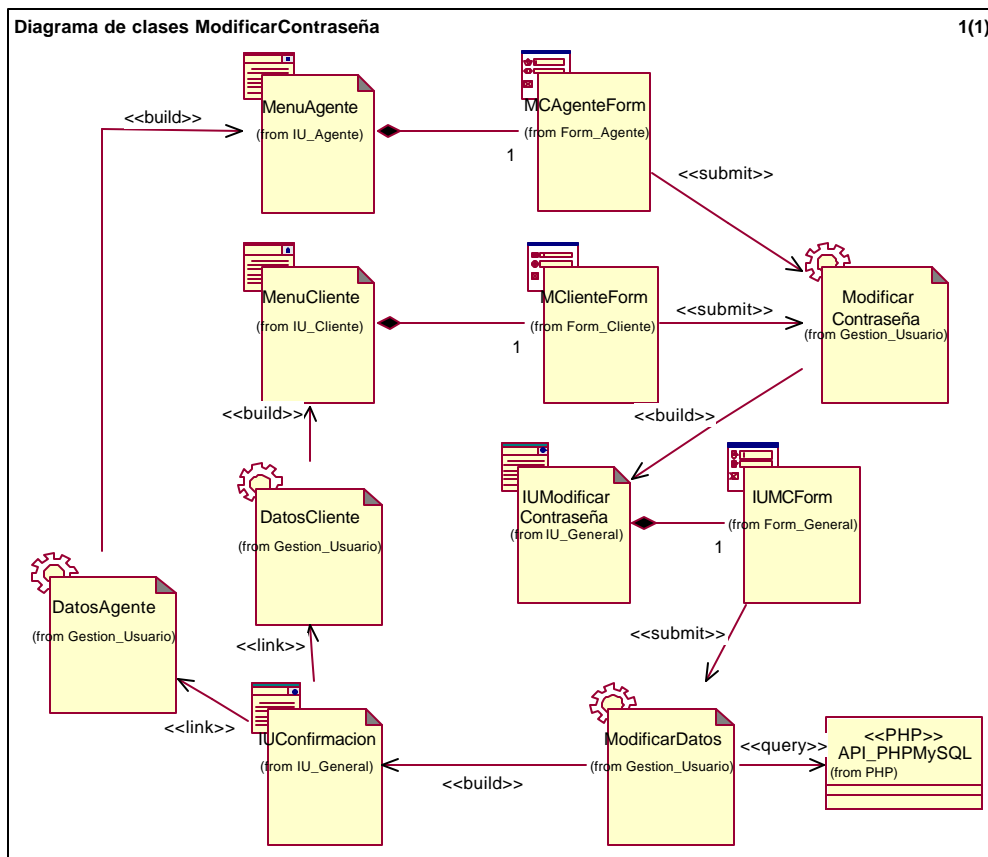


Figura C-7. Diagrama de clases ModificarContraseña, (Estereotipos)

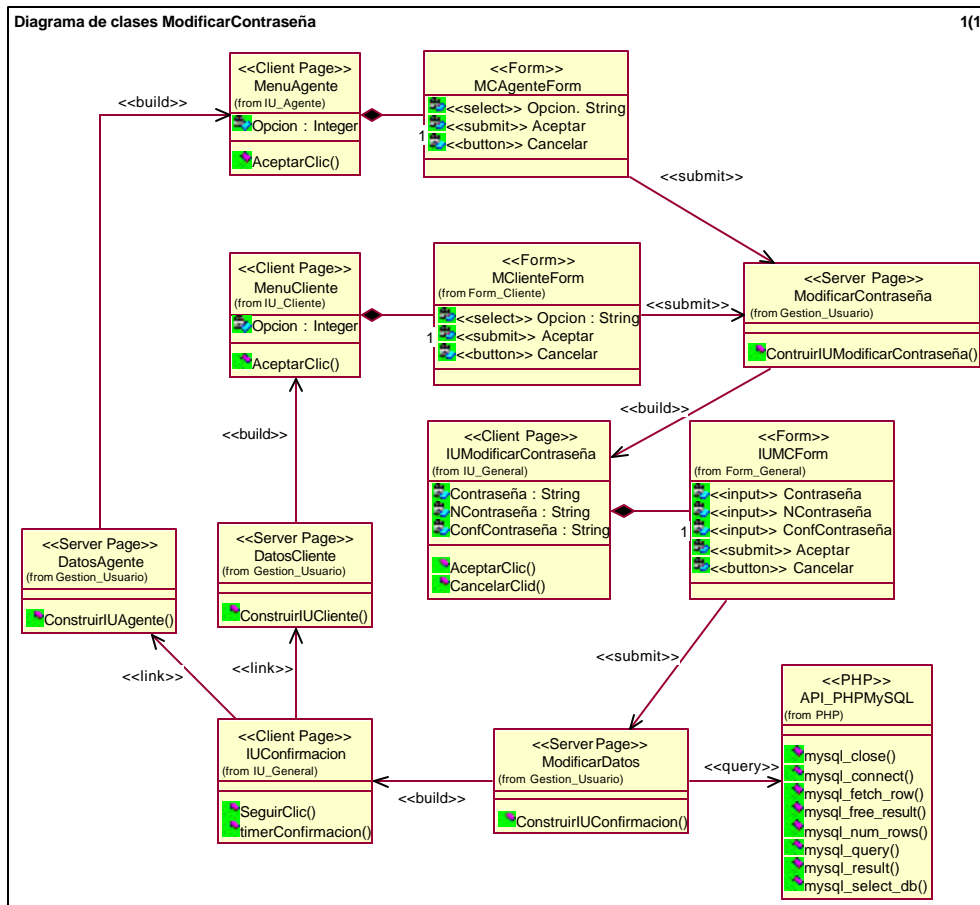


Figura C-8. Diagrama de clases Modificar contraseña

El usuario, (Cliente o Agente) inicia el caso de uso cuando realiza la invocación de la Server page `ModificarContraseña`, desde el formulario contenido en la Client page `MenuAgente` o `MenuCliente`, según el caso. La Server page, se encarga de construir la `IUModificarContraseña`, en la cual se presenta al usuario un formulario que le solicita la introducción de algunos datos, (Contraseña anterior, Nueva contraseña y la confirmación de la contraseña nueva), lo cuales se envían a la Server page `ModificarDatos` y esta se encarga de actualizar los datos en la base de datos, de acuerdo a un análisis previo de estos, por medio del cual verifica la contraseña, (realizando una consulta a la base de datos) y si la confirmación de la contraseña coincide con la nueva contraseña introducida.

Si los datos son correctos, la Server page, `ModificarDatos`, modifica lo datos en la base de datos y construye la Client page `IUConfirmacion`, la cual contiene un timer

que se encarga de invocar a la Server page Datos Cliente o Datos Agente, según el tipo de usuario.

```
Class MenuAgente {  
    Var opcion;  
    AceptarClic() {  
        Switch(this->opcion){  
            Caso "Ver producto":  
                DatosEmpresa->ConstruirListaProductos()  
                break;  
            Caso "ModificarContraseña":  
                ModificarContraseña->ModificarContraseña()  
                break;  
            Caso "Salir":  
                CerrarSesion();  
                break;  
        }  
    }  
}
```

```
Class MenuCliente {  
    Var opcion;  
    AceptarClic() {  
        Switch(this->opcion){  
            Caso "Ver empresas":  
                DatosEmpresa->ConstruirListaEmpresas()  
                break;  
            Caso "ModificarContraseña":  
                ModificarContraseña->ModificarContraseña()  
                break;  
            Caso "Salir":  
                CerrarSesion();  
                break;  
        }  
    }  
}
```

}

Class ModificarContraseña {

```
    ConstruirIUModificarContraseña() {  
        Var Contraseña;  
        Var Ncontraseña;  
        Var ConfContraseña;  
  
        AceptarClic() {  
            ModificarDatos->ConstruirIUConfirmacion();  
        }  
        CancelarClic() {  
            Volver a menu agente;  
        }  
    }  
}
```

}

Class ModificarDatos {

```
    ContraseñaAntigua=ModificarContraseña->Contraseña;  
    ContraseñaNueva=ModificarContraseña->Ncontraseña;  
    ConfirmaciónContraseña=ModificarContraseña->ConfContraseña;  
    ConstruirIUConfirmacion {  
        Si ValidarAccesoUsuario->Usuario==C {  
            TablaConsulta="Cliente"  
        }  
        Si no {  
            TablaConsulta="Agente"  
        }  
        Query="Seleccionar Contraseña de la tabla TablaConsulta donde login=  
        'ValidarAcceso->login';  
        Resultado=EjecutarConsulta;  
        Si Resultado[0]==ContraseñaAntigua {  
            Si ContraseñaNueva==ConfirmacionContraseña {  
                Query="Actualizar Contraseña de la tabla TablaConsulta  
                donde login'"ValidarAcceso->login";
```



```
        EjecutarConsulta;
    }
    Si no {
        Error=1;
    }
}
Si no {
    Error=1;
}
Si Error==1{
    Desplegar mensaje "Contraseña o confirmación de contraseña
    incorrecta"
    ModificarContraseña->ConstruirIUModificarContraseña();
}
Si no {
    Desplegar mensaje "Constraseña actualizada";
    Iniciar timer;
}
}
}
```

Class IUConfirmacion {

```
    TimerConfirmacion() { //Después de 2 segundos
        Si ValidarAccesoUsuario->Usuario==A{
            DatosAgente->ConstuirIUAgente();
        }
        Si no {
            DatosCliente->ConstuirIUCliente();
        }
    }
    SeguirClic() {
        Si ValidarAccesoUsuario->Usuario==A{
            DatosAgente->ConstuirIUAgente();
        }
        Si no {
            DatosCliente->ConstuirIUCliente();
        }
    }
}
```

C.2.2 Caso de uso RealizarPedido

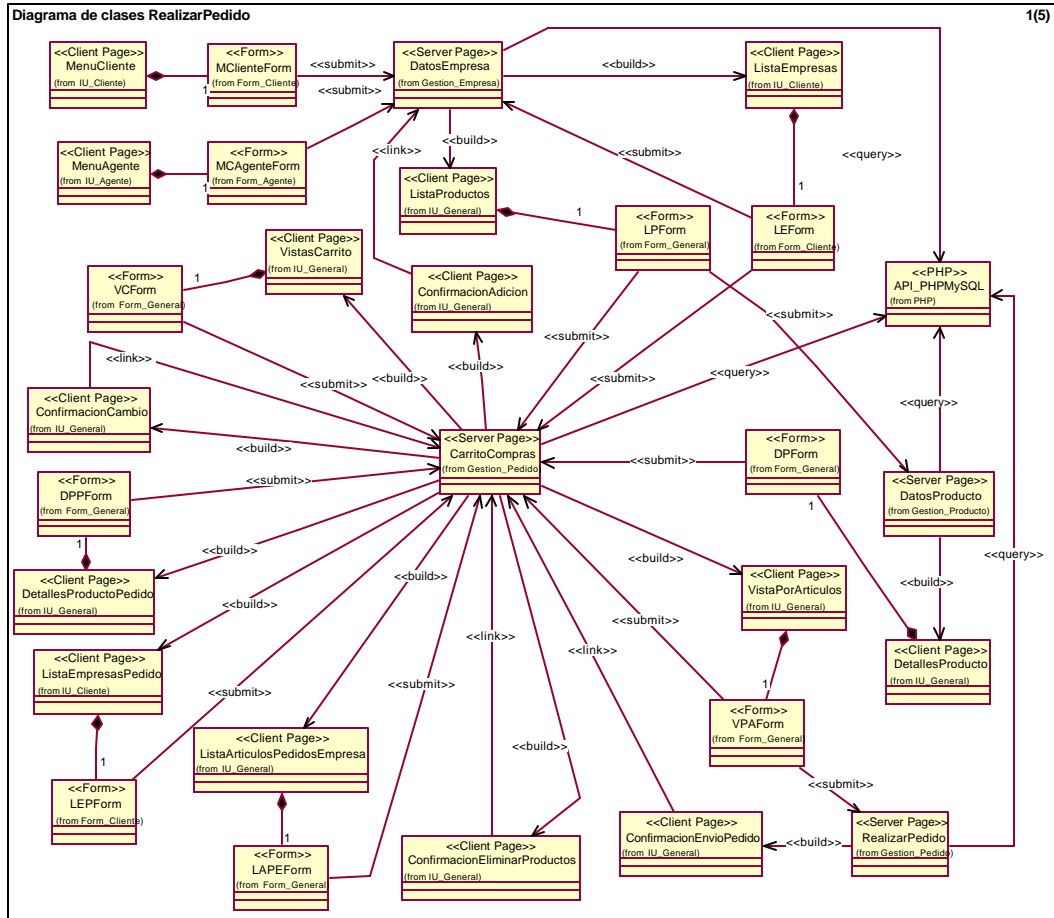


Figura C-9. Diagrama de clases caso de uso RealizarPedido

Este caso de uso, permite al usuario enviar un listado de productos, los cuales son presentados a cada empresa, proveedora, como un pedido. El caso de uso puede ser accedido tanto por Clientes como por Agentes.

Después de haberse validado en el sistema, un Cliente puede hacer la solicitud del listado a las empresas a las cuales se encuentra suscrito, es decir, para las cuales

se encuentra habilitado para realizar pedidos, escogiendo la opción “Empresas”, en la Client page MenuCliente, la cual contiene un formulario que se encarga de enviar la petición a la Server page DatosEmpresa.

La Server page Datos empresa consulta la información del usuario y los datos de las empresas a las cuales este se encuentra suscrito y construye la lista por medio de la cual el Cliente puede acceder a la lista de productos de una empresa determinada. Estos datos son desplegados al Cliente en la Client page ListaEmpresas.

La Client page ListaEmpresas, únicamente, es mostrada a los Clientes debido a que solamente estos pueden estar suscritos en más de una empresa.

El usuario puede acceder a la lista de productos de una empresa, enviando la petición a la Server page DatosEmpresa, por medio del formulario que se encuentra contenido en dicha Client page. De esta forma la Server page, construye una lista de productos de la empresa y las despliega al cliente en la Client page ListaProductos.

Un agente solamente puede acceder a la lista de productos de la empresa a la cual representa. Esto es posible enviando la petición a la Server page DatosEmpresa, directamente desde la Client page MenuAgente.

Una vez el usuario, (Cliente o Agente), tiene la lista de productos, puede hacer la petición para que uno de estos sea mostrado en detalle, mediante el Form que se encuentra en la Client page ListaProductos, éste invoca a la Server page DatosProducto, para que esta construya la Client page Detalles producto, la cual contiene un Form, que permite al usuario adicionar el producto al carrito de compras; si decide hacerlo, datos como cantidad, valor y código del producto son enviados a la Server page CarritoCompras, la cual se encarga de construir el carrito de compras del usuario y una Client page de confirmación, la cual dispone de un timer que invoca a la Server page DatosEmpresa para que construya la lista de productos, en la cual se encontraba el usuario cuando adicionó el producto al carrito de compras.

Solamente si el usuario ha agregado, al menos, un producto al carrito de compras, es posible acceder a él, mediante la opción que se encuentra en las Client page, ListaProductos y ListaEmpresas, la cual solamente está disponible si existen productos en el carrito de compras.

La Client page VistasCarrito, en la cual se muestran varias opciones, (Vista de pedidos por artículos, vista de pedido por empresas y eliminar productos), que permiten al usuario gestionar los productos que se encuentran en él, es construida cuando se invoca la Server page CarritoCompras.

La Client page VistaPorArticulos es construida por la Server page CarritoCompras cuando el usuario, realiza la petición desde el Form, contenido en la Client page VistasCarrito. Esta Client page, muestra al usuario la información correspondiente a Referencia, Cantidad, Valor y Valor total de todos los productos contenidos en el carrito de compras; Además contiene un Form que permite al usuario invocar la Server page Carrito Compras, para ver en detalle cada producto contenido en el carrito de compras y a la Server page Realizar Pedido, para enviar la solicitud del pedido de los productos mostrados en ésta Client page. Esta solicitud es almacenada en la base de datos, en caso de que en la misma solicitud se envíen productos de diferentes empresas, se almacenarán un pedido para cada empresa.

Cuando el usuario, realiza un clic sobre la referencia de un producto mostrado en la Client page VistaPorArticulos; el Form, contenido en ella, envía la petición a la Server page CarritoCompras la cual se encarga de construir la Client page que contiene información, almacenada en la base de datos, del producto escogido, y la cantidad que el usuario quiere de ese producto. Además contiene un enlace que permite al usuario cambiar la cantidad de producto. Si decide realizar cambios, estos son enviados a la Server page CarritoCompras, por medio de un Form. Esta Server page realiza el cambio, actualiza el carrito de compras y construye una Client page que indica al usuario que el cambio ha sido realizado y reenvía después de un corto tiempo (aprox. 2 seg) al la Client page en la cual estaba antes de invocar la Client page DetalleProductoPedido.

También el usuario, utilizando el Form contenido en Client page VistasCarrito, puede enviar a la Server page CarritoCompras, la solicitud de eliminar productos del carrito de compras, En este caso, la Server page construye la Client page, ListaEmpresasPedido, que contiene una lista de las empresas a las cuales se les va a realizar el pedido, por medio de esta Client page el usuario puede acceder a la lista de productos que va a pedir de una determinada empresa, enviando la invocación, desde el Form contenido en la Client page. Esta solicitud es enviada a la Server page CarritoCompras, la cual construye la lista de productos y la despliega al usuario en la Client page, ListaArticulosPedidosEmpresa. Por medio del Form contenido en esta Client page, el usuario puede hacer la petición de eliminación de un producto del carrito de compras. Esta petición es enviada a la Server page CarritoCompras la cual envía una Client page al usuario solicitándole la confirmación de la eliminación del producto del carrito de compras. Si el usuario decide eliminar el producto, la Server page reconstruye el carrito de compras y envía al usuario la Client page ListaArticulosPedidosEmpresa, si aún existen artículos en el carrito de compras de esa empresa, en caso de que no existan, envía al usuario la Client page inmediatamente superior, que en este caso se trata de la Client page ListaEmpresasPedido. En caso de que el carrito de compras llegara a quedar vacío la Server page enviaría al usuario una aviso informándole que el carrito de compras se encuentra vacío e inmediatamente después a la Client page desde la cual se envió la solicitud de despliegue de la Client page VistasCarrito, (puede ser ListaEmpresas o ListaProductos).

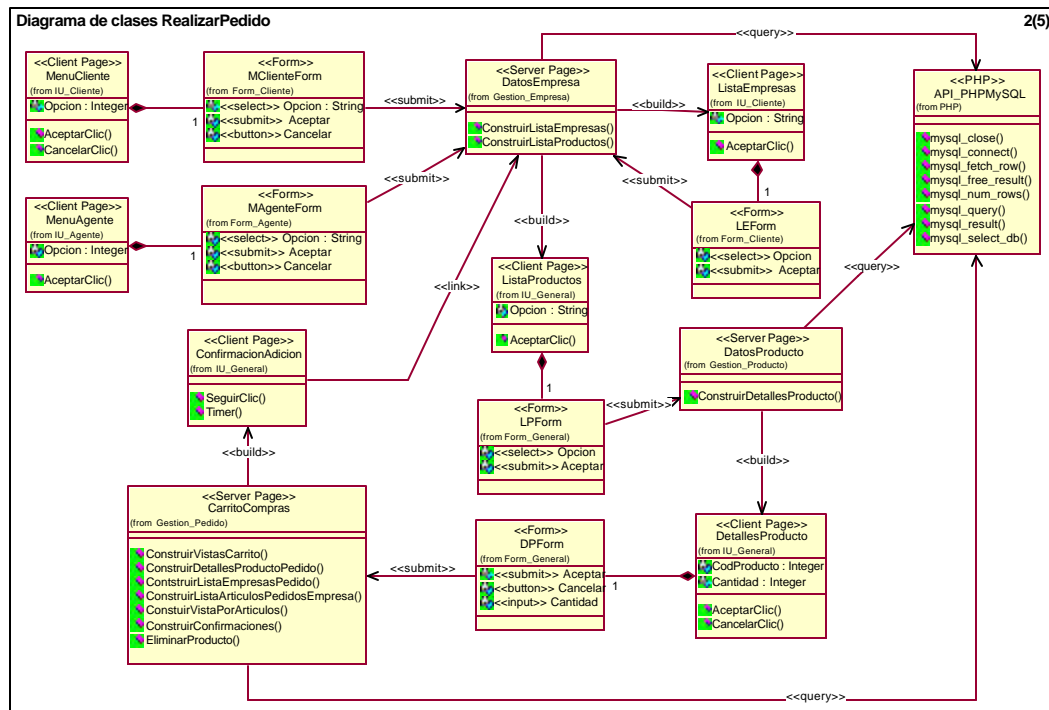


Figura C-10. Diagrama de clases RealizarPedido, (Adicionar producto al carrito de compras)

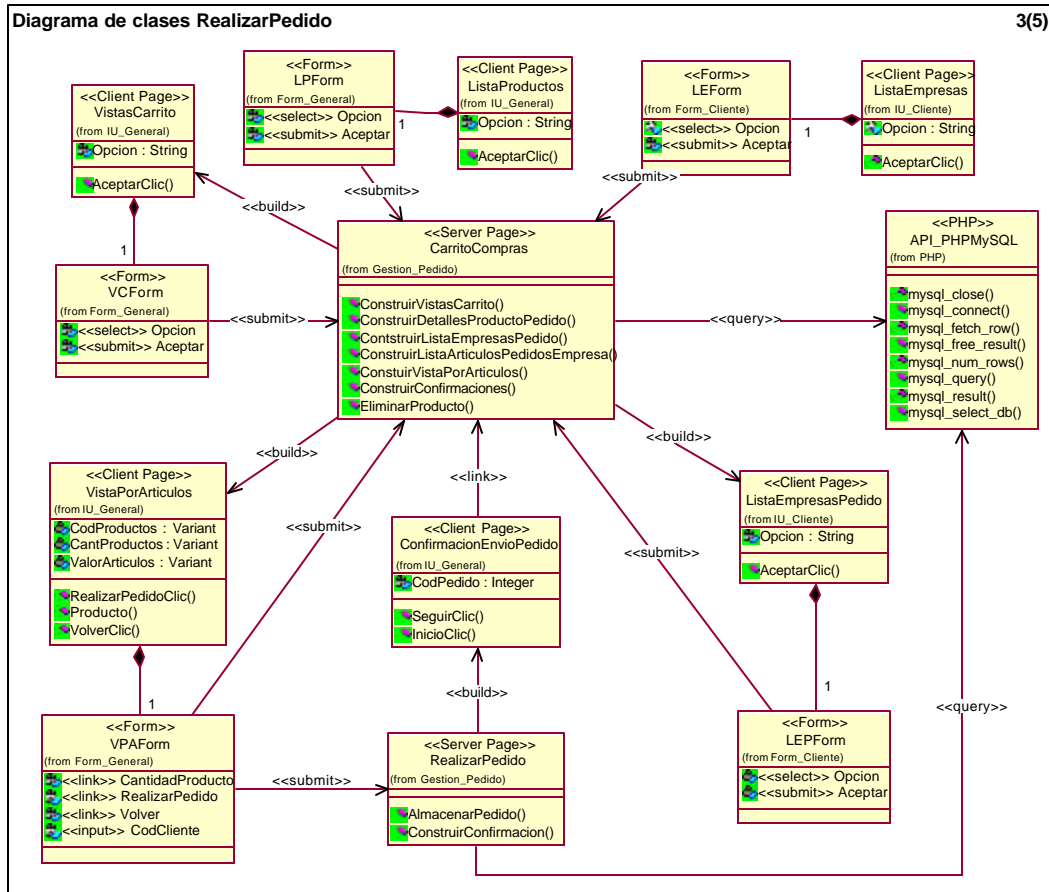


Figura C-11. Diagrama de clases RealizarPedido, (Enviar Pedido)

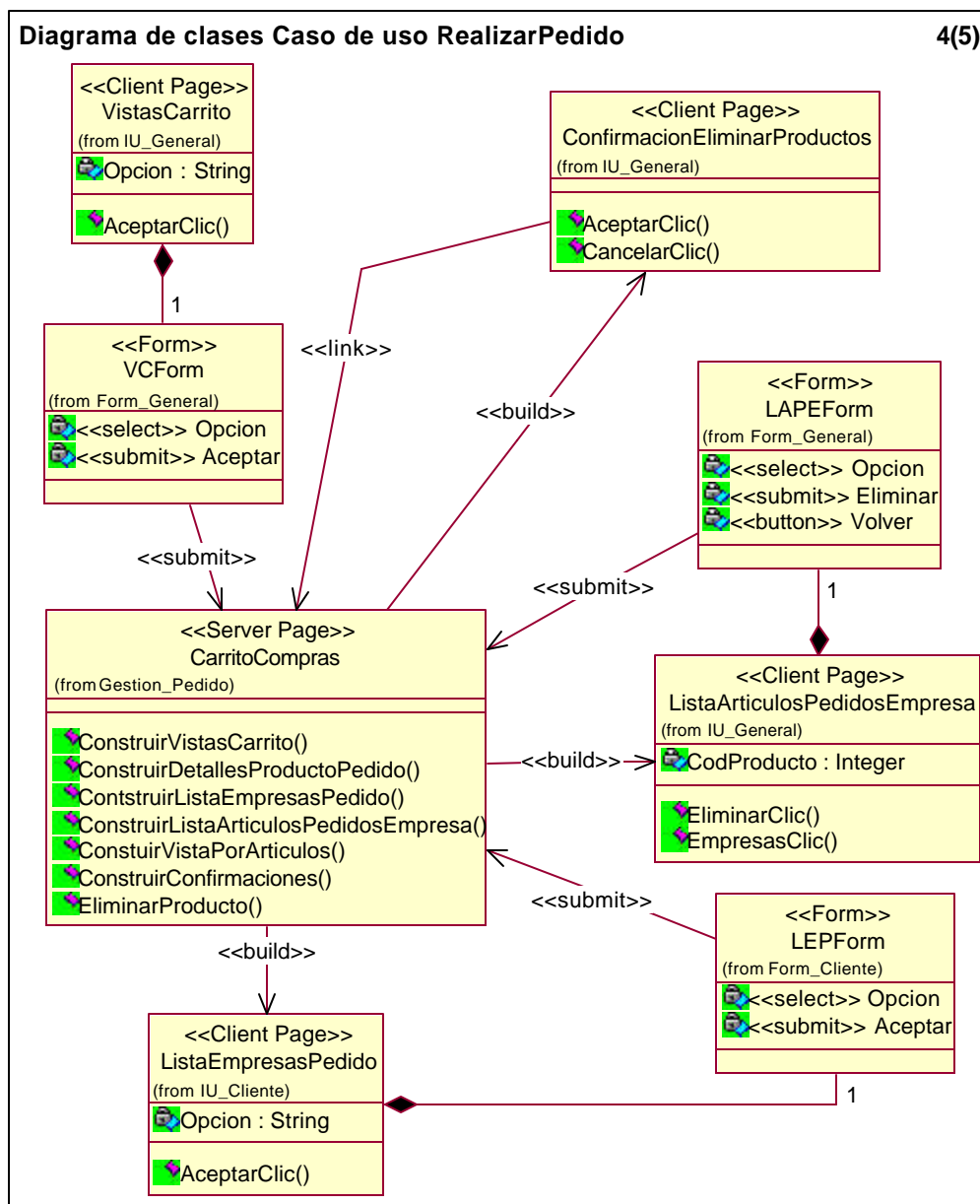


Figura C-12. Diagrama de clases caso de uso RealizarPedido, (Eliminar producto del carrito de compras)

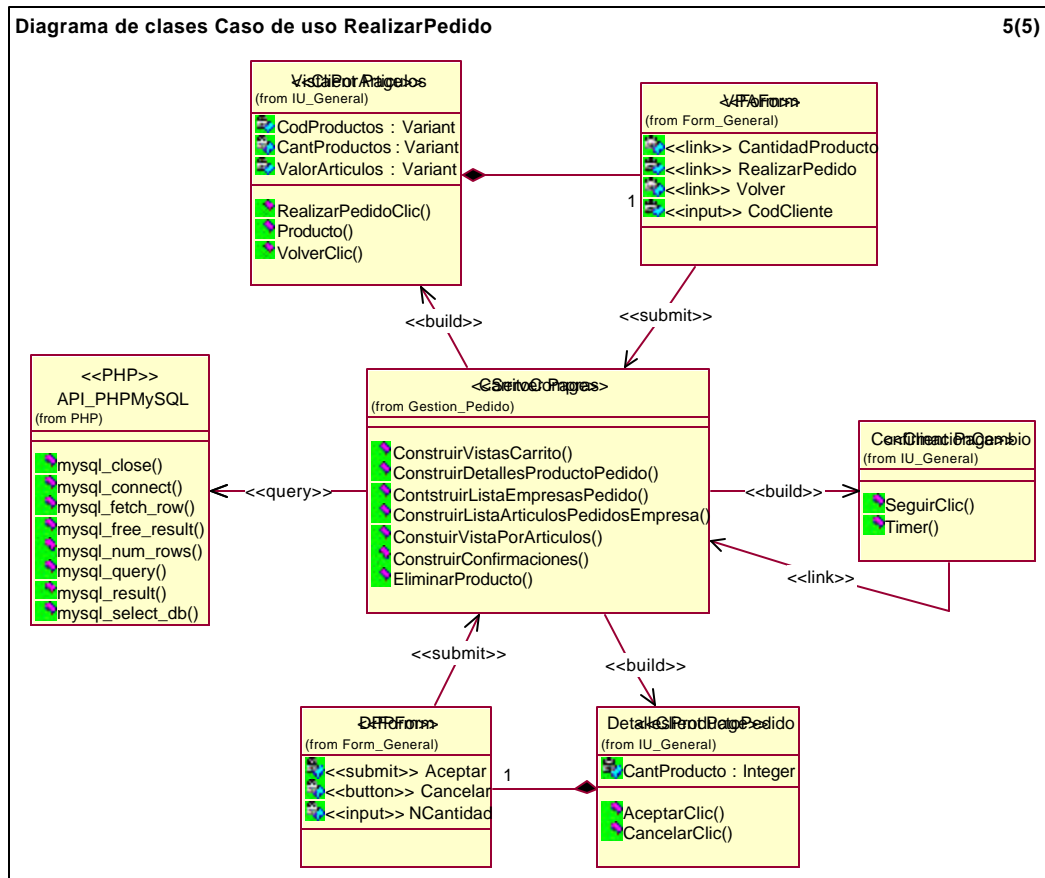


Figura C-13. Diagrama de clases Caso de uso RealizarPedido, (Cambiar cantidad de producto a pedir)

Class DatosEmpresa {

ConstruirListaEmpresas() {

Query="Consultar lista de empresas de la tabla Cliente en donde el cliente se encuentre registrado";

Resultado="Ejecutar consulta(query)";

Desplegar lista resultado;

}

Construir ListaProductos(CodigoEmpresa) {

Query="Seleccionar Nombre, codigo, referencia de la tabla productos donde codigo empresa='CodigoEmpresa'";

Resultado="Ejecutar consulta(query)";

Desplegar lista resultado;

}

```
}
```

Class ListaEmpresas{

```
    Var opcion;  
    AceptarClic() {  
        Switch (estaclase->opcion) {  
            Caso "VerCarritoCompras"  
                CarritoCompras ->ConstuirVistasCarrito;  
                Break;  
            Caso "VerProductos":  
                DatosEmpresa->ConstruirListaProductos(estaclase  
                                                            ->opcion);  
                Break;  
        }  
    }  
}
```

Class ListaProductos {

```
    Var opcion;  
    AceptarClic() {  
        Switch (estaclase->opcion) {  
            Caso "VerCarritoCompras"  
                CarritoCompras ->ConstuirVistasCarrito;  
                Break;  
            Caso "VerDetalleProducto":  
                DatosProducto->ConstruirDetallesProducto(estaclase  
                                                            ->opcion);  
                Break;  
            Caso "volver":  
                DatosEmpresa->ConstruirListaEmpresas(),  
                Break;  
        }  
    }  
}
```

Class DatosProducto {

```
    ConstruirDetallesProducto() {  
        Query="Consultar Nombre, Cantidad, Empresa, Valor de la tabla productos  
donde Codigo='ListaProductis->opcion";  
        Resultado=EjecutarConsulta(Query);  
        Desplegar datos producto;  
    }  
}
```

Class DetallesProducto {

```
    AceptarClic() {  
        Adicionar producto al carrito de compras;  
        CarritoCompras ->TipoConfirmacion=Adicion;  
        CarritoCompras ->ConstruirConfirmacion();  
    }  
    CancelarClic() {  
        Volver a la lista de productos;  
    }  
}
```

Class ConfirmacionAdicion {

```
    TimerConfirmacion() { //Después de 2 segundos  
        DatosEmpresa->ConstruirListaProductos();  
    }  
    SeguirClic() {  
        DatosEmpresa->ConstruirListaProductos();  
    }  
}
```

Class CarritoCompras{

```
    Var TipoConfirmacion;  
    ConstruirVistasCarrito() {
```

```
        Desplegar opciones de vistas del carrito; //por artículos, por empresas,
                                                //eliminar productos, volver a
                                                //lista de empresas o lista de
                                                //productos.
    }

    ConstruirDetallesProductoPedido(codigo) {
        Query=consultar todos los datos de la tabla productos donde
        codigo='codigo';
        Resultado=Ejecutar consulta;
        Desplegar resultados;
    }

    ConstruirListaEmpresasPedido(codigo) {
        Para todos los codigos de las empresas del carrito de compras{
            Query=Consultar nombre, codigo de la tabla empresas donde
            codigo='codigo';
            Resultado=Ejecutar consulta;
            Almacenar resultado;
        }
        Desplegar resultados;
    }

    ConstruirListaArticulosPedidosEmpresa(codigo) {
        Para todos los codigos de los productos del carrito de compras donde codigo
        = 'codigo' {
            Query consultar nombre, codigo de la tabla productos donde
            codigo='codigo';
            Resultado=Ejecutar consulta;
            Almacenar resultado;
        }
        Desplegar resultado;
    }

    ConstruirVistaPorArticulos(CodigoEmpresa) {
        Si CodigoEmpresa="" {
            Desplegar los datos de los productos almacenados en el carrito de
            compras;
            Calcular valores totales; //Valor total del pedido;
            Desplegar valores totales;
        }
    }
```

```
}
ConstruirConfirmaciones() {
    Switch (estaclase->TipoConfirmacion) {
        Caso "Adicion":
            Desplegar "Articulo adicionado";
            IniciarTimer;
        Break;
        Caso "Cambio":
            Desplegar "Cambio realizado";
            IniciarTimer;
        Break;
        Caso "Eliminar":
            Desplegar "Esta seguro de que desea eliminar el (los)
            producto(s) del carrito de compras?";
            Desplegar Boton Aceptar;
            Desplegar Boton Cancelar;
        Break;
    }
}
EliminarProducto(codigo) {
    Borrar de la lista de productos el articulo de codigo = codigo;
}
}
```

Class VistaPorArticulos {

Var CodProductos;

Var CantProductos;

Var ValorArticulos;

Var CodigoP;

RealizarPedidoClic(){

RealizarPedido->AlmacenarPedido();

RealizarPedido->ConstuirConfirmacion();

}

ProductoClic() {

```
CarritoCompras ->ConstuirDetallesProductoPedido(estaclase->CodigoP);
}

VolverClic() {
    Si PaginaAnterior==Empresas {
        DatosEmpresa->ConstuirListaEmpresas();
    }
    Si no {
        DatosEmpresa->ConstuirListaProductos();
    }
}
}

Class ListaEmpresasPedido {
    Var Opcion;
    AceptarClic(tipo) {
        Si tipo==eliminar {
            CarritoCompras
            ->ConstuirListaArticulosPedidoEmpresa(estaclase->Opcion);
        }
        else {
            CarritoCompras ->ConstruirVistaPorArticulos(estaclase->Opcion);
        }
    }
}

Class ConfirmacionEnvioPedido {
    Si CarritoCompras <> Vacio {
        SeguirClic() {
            CarritoCompras ->ConstruirVistasCarrito();
        }
    }
    else {
        Si ValidarAccesoUsuario ->Login == Cliente {
            DatosCliente->ConstruirIUCliente();
        }
    }
}
```

```
    }  
    Else{  
        DatosAgente->ConstruirIUAgente();  
    }  
}  
}
```

Class RealizarPedido {

```
    AlmacenarPedido() {  
        CodigosProductos= VistaPorArticulos->CodProductos;  
        CantProductos= VistaPorArticulos->CantProductos;  
        ValorProductos=VistaPorArticulos->ValorArticulos;  
        Query="Almacenar en la base de datos ProductoPedido CodigosProductos,  
            CantidadProductos, ValorProductos,CodigoCliente, CodigoAgente,  
            Fecha";  
        EjecutarConsulta;  
        EstaClase->ConstruirConfirmacion();  
    }  
    ConstruirConfirmacion() {  
        Desplegar "Pedido realizado";  
        Query= "Seleccionar codigo del ultimo item insertado, por este usuario,  
            en la tabla pedido";  
        Resultado=EjecutarConsulta;  
        Almacenar resultado;  
        Desplegar "Codigo del pedido: 'Resultado[0]'";  
    }  
}
```

Class DetallesProductoPedido {

```
    AceptarClic(CodigoProducto) {  
        ActualizarCantidadProductoCarritoCompras;  
        CarritoCompras ->CosntruirConfirmaciones("Cambio");  
    }  
    CancelarClic() {  
        CarritoCompras ->ConstruirDetallesProductoPedido();  
    }  
}
```

```
}
```

Class ConfirmacionCambio {

```
    TimerConfirmacion() { //Después de 2 segundos
        CarritoCompras -> ConstruirDetallesProductoPedido();
    }
    SeguirClic() {
        CarritoCompras -> ConstruirDetallesProductoPedido();
    }
}
```

Class ListaArticulosPedidosEmpresa {

```
    EliminarClic() {
        CarritoCompras -> EliminarProducto();
    }
    EmpresasClic() {
        CarritoCompras -> ConstruirListaEmpresasPedido;
    }
}
```

Class ListaEmpresasPedido {

```
    Var Opcion;
    AceptarClic() {
        CarritoCompras -> ConstruirListaArticulosPedidosEmpresa(estaclase
            ->opcion);
    }
}
```

Class ConfirmacionEliminarProductos {

```
    AceptarClic() {
        Si ya no hay mas articulos de la empresa actual {
            CarritoCompras -> ConstruirListaArticulosPedidosEmpresa();
        }
        Si no si hay articulos todavia{
            CarritoCompras -> ConstruirListaEmpresasPedido();
        }
    }
}
```



```
    }
    Si no {
        Si ValidarAccesoUsuario ->Login == Cliente {
            DatosCliente->ConstruirIUCliente();
        }
        Else {
            DatosAgente->ConstruirIUAgente();
        }
    }
}
CancelarClic() {
    CarritoCompras ->ConstruirListaEmpresasPedido();
}
}
```

C.2.3 Caso de uso BuscarPedido

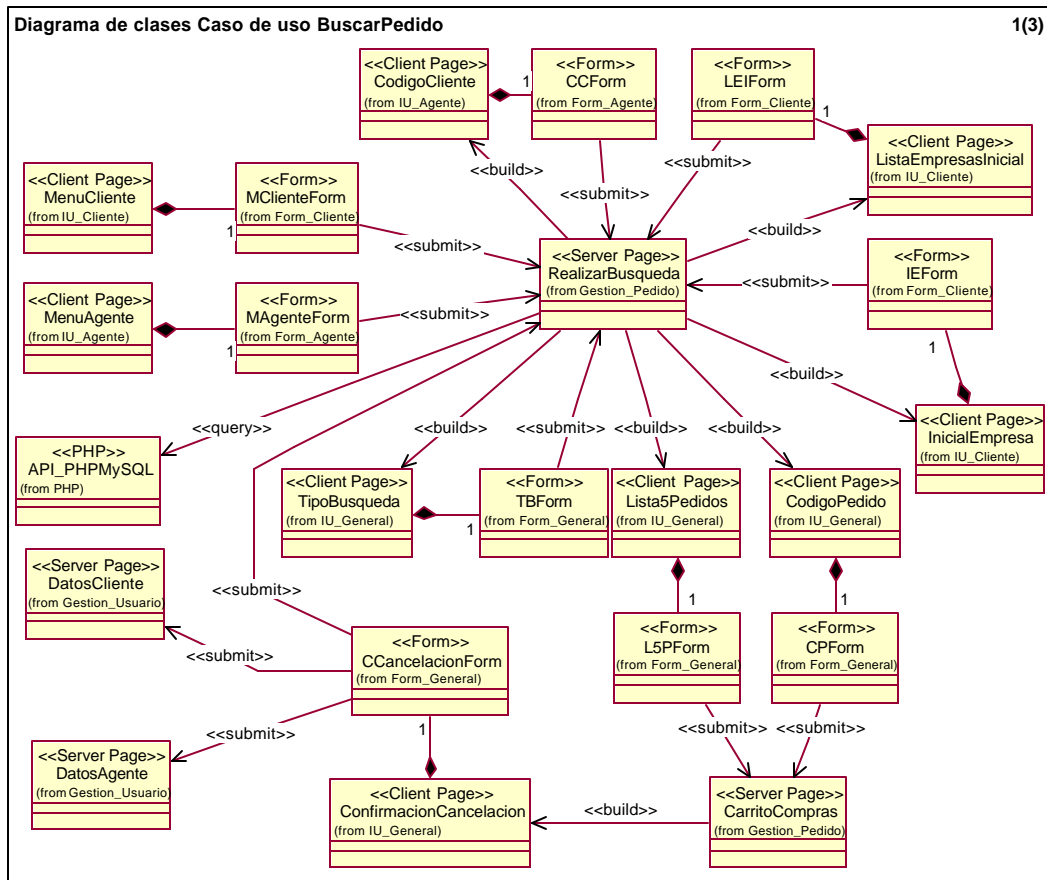


Figura C-14. Diagrama de clases, caso de uso BuscarPedido

Cuando el Cliente o Agente, desde su respectiva Client page (MenuCliente, MenuAgente) invoca la interfaz de búsqueda de pedido, la server page RealizarBusqueda, atiende la solicitud.

Si se trata de un agente, la Server page RealizarBusqueda, construira la Client page CodigoCliente, la cual contiene un Form, para enviar el codigo del cliente, a nombre quien se encuentra el pedido que desea buscar. Cuando el Agente retorna el código del cliente a la server page, esta se encarga de construir otra Client page, TipoBusqueda, por medio de la cual el usuario puede escoger el criterio de búsqueda de pedido, (por código o listar los 5 últimos pedidos), esto es enviado a la Server page utilizando el Form, Tbfom, contenido en la Client page. Si escoge la opción de búsqueda por código. La Server page, construye la Client page

CodigoPedido, que contiene un Form, por medio del cual el Agente puede enviar a la Server page CarritoCompras el código del pedido que desea ver, de esta forma, se muestra el pedido en la Client page VistaPorArticulos. De la misma forma si el Agente escoge la opción, listar los últimos 5 pedidos, la server page RealizarBusqueda construira, la Client page Lista5Pedidos, la cual contiene una lista de los últimos 5 pedidos realizados a la empresa. Por medio del Form contenido en esta Client page el Agente puede enviar el Código del pedido que desea ver a la Server page CarritoCompras, para que construya la Client page VistaPorArticulos.

En el caso de Cliente la Server page RealizarBusqueda, construye la Client page, InicialEmpresa, por medio de la cual y valiéndose del Form contenido en ella, el cliente debe introducir un caracter o caracteres, con los que inicia la empresa a la cual realizó el pedido que desea ver. Esta inicial es enviada de nuevo a la server page para que esta construya la Client page ListaEmpresasInicial, la cual contiene una lista de las empresas a las cuales el cliente esta suscrito y que su nombre empieza por el(los) caracter(es) introducido(s). Además esta Client page contiene un Form, por medio del cual el Cliente envía a la Server page la inicial de la empresa, es decir solamente se buscarán pedidos de la empresa seleccionada. A continuación la Server page construye la Client page TipoBusqueda y el procedimiento a seguir es exactamente el mismo que el del agente, desde esta Client page en adelante.

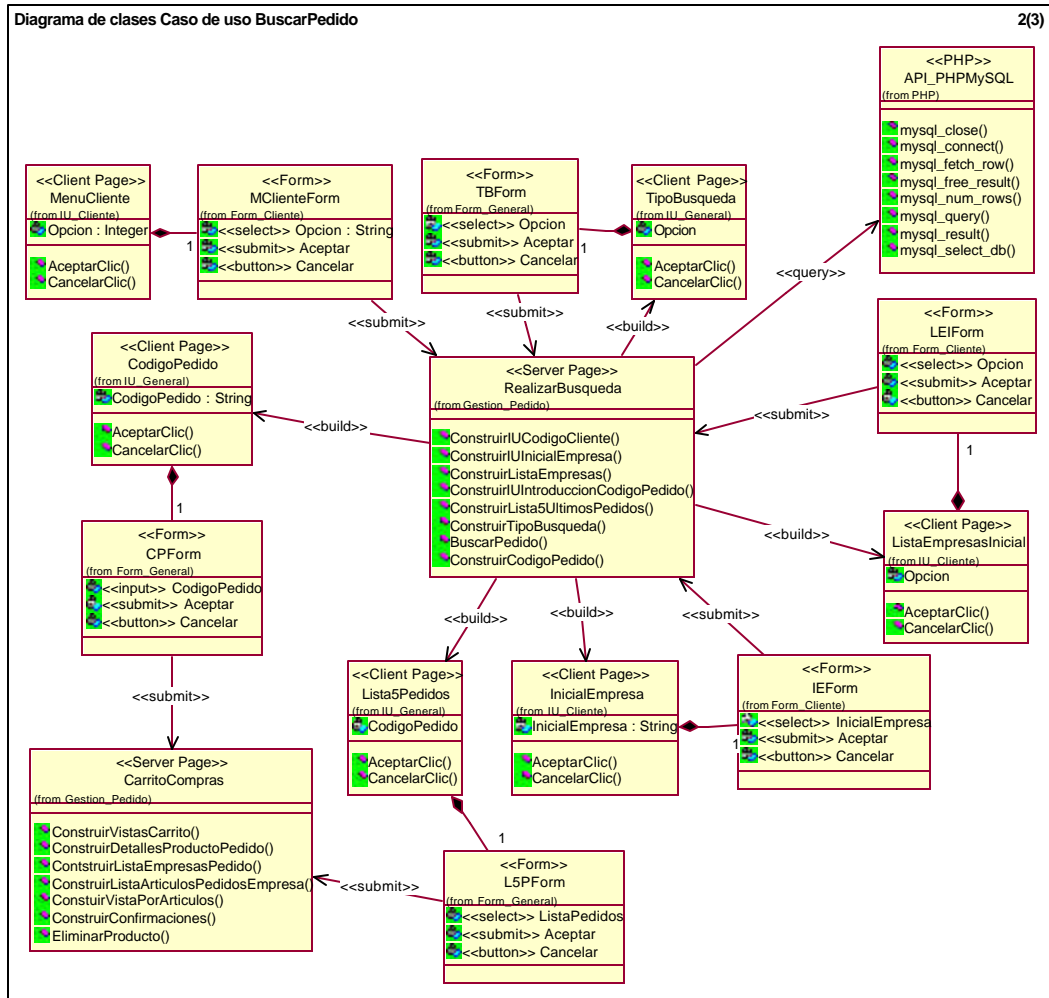


Figura C-15. Diagrama de clases, caso de uso BuscarPedido, (búsqueda de pedido por un cliente)

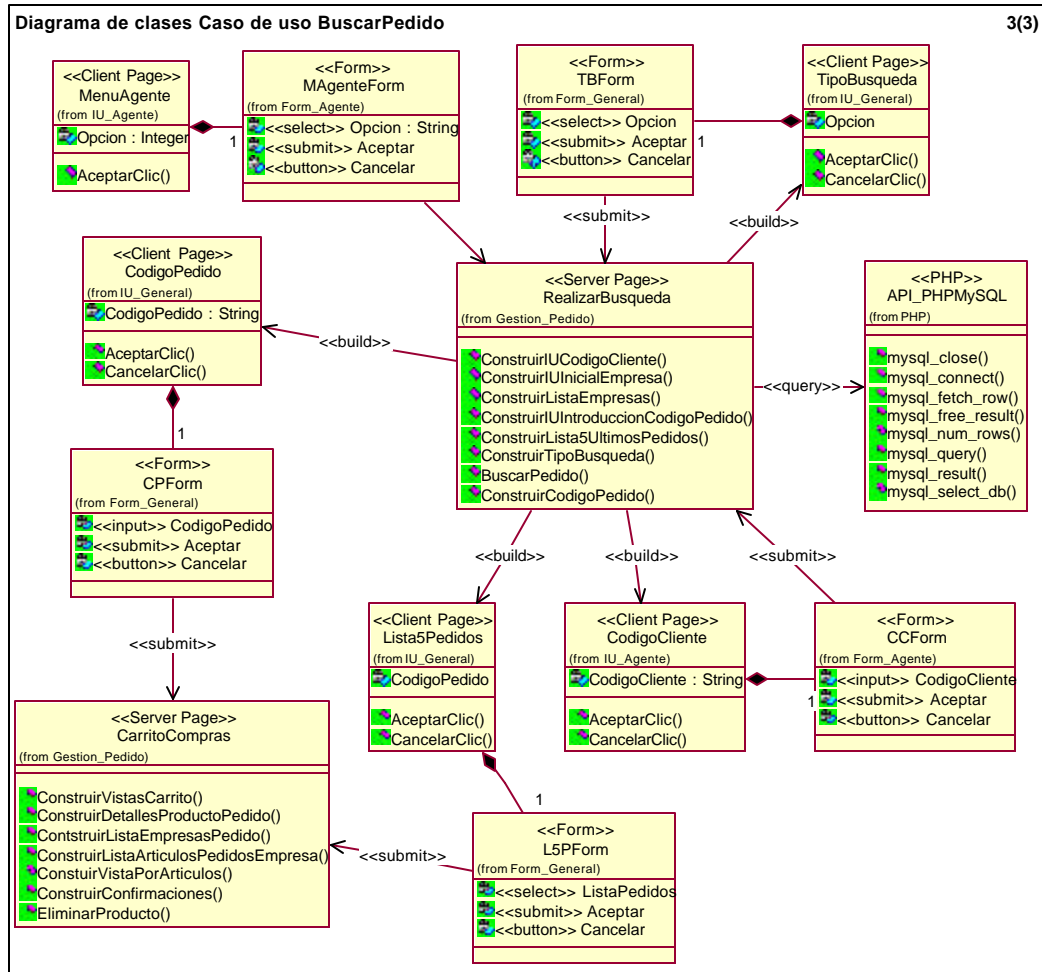


Figura C-16. Diagrama de clases, Caso de uso BuscarPedido, (búsqueda por agente)

```

Class CodigoCliente {
    Var CodigoCliente;

    AceptarClic() {
        RealizarBusqueda->ConstruirTipoBusqueda();
    }
    CancelarClic() {
        DatosAgente->ConstruirIUAgente();
    }
}

```

Class InicialEmpresa {

```
Var InicialEmpresa;  
    AceptarClic() {  
        RealizarBusqueda->ConstruirListaEmpresas(estaclase->InicialEmpresa);  
    }  
    CancelarClic() {  
        DatosCliente->ConstruirIUCliente();  
    }  
}
```

ClassListaEmpresasInicial {

```
Var Opcion;  
    AceptarClic(){  
        RealizarBusqueda->ConstruirTipoBusqueda(estaclase->Opcion);  
    }  
    CancelarClic() {  
        RealizarBusqueda->ConstruirIUIncialEmpresas();  
    }  
}
```

Class TipoBusqueda {

```
Var Opcion;  
    AceptarClic() {  
        Si estaclase->Opcion == "BusquedaPorCodigo" {  
            RealizarBusqueda->ConstruirUIIntroduccionCodigoPedido();  
        }  
        Si no {  
            RealizarBusqueda->ConstruirLista5UltimosPedidos();  
        }  
    }  
    CancelarClic() {  
        Si ValidarAccesoUsuario->Login="Cliente" {
```

```
        RealizarBusqueda->ConstruirListaEmpresas;
    }
    Si no {
        RealizarBusqueda->ConstruirIUCodigoCliente;
    }
}
}
```

Class CodigoPedido {

```
    Var CodigoPedido;
    AceptarClic() {
        Query="Consultar todos lo datos de la tabla pedido donde
              codigo='codigo'";
        Resultado=Ejecutar consulta (Query);
        AlmacenarResultado;
        CarritoCompras ->ConstruirVistaPorArticulos(Resultado);
    }
    CancelarClic() {
        RealizarBusqueda->CosntruirTipoBusqueda();
    }
}
}
```

Class Lista5Pedidos {

```
    Var CodigoPedido;
    AceptarClic() {
        Query="Consultar todos lo datos de la tabla pedido donde
              codigo='codigo'";
        Resultado=Ejecutar consulta (Query);
        AlmacenarResultado;
        CarritoCompras ->ConstruirVistaPorArticulos(Resultado);
    }
    CancelarClic() {
        RealizarBusqueda->CosntruirTipoBusqueda();
    }
}
}
```

Class RealizarBusqueda {

 ConstruirIUCodigoCliente() {

 Desplegar input;

 Desplegar boton Aceptar;

 Desplegar boton Cancelar;

 }

 ConstruirUIInicialEmpresa() {

 Desplegar input

 Desplegar boton Aceptar;

 Desplegar boton Cancelar;

 }

 ConstruirListaEmpresas(CodigoCliente, Inicial) {

 Query="Seleccionar todas las empresas en las que se encuentre
 registrado el cliente con codigo='codigo' y empiecen por='Inicial';

 Resultado=Ejecutar consulta(query);

 AlmacenarResultado;

 Desplegar resultado;

 Desplegar boton Aceptar;

 Desplegar boton Cancelar;

 }

 CosntruirUIIntroduccionCodigoPedido() {

 Desplegar input;

 Desplegar boton Aceptar;

 Desplegar boton Cancelar;

 }

 ConstruirLista5UltimosPedidos(CodigoCliente, CodigoEmpresa) {

 Query="Seleccionar codigo de los ultimos 5 registro de la tabla pedido
 donde CodigoCliente = 'CodigoCliente' y CodigoEmpresa
 ='CodigoEmpresa";

 Resultado=Ejecutar consulta(Query);

 Desplegar resultado;

 Desplegar boton Aceptar;

 Desplegar boton Cancelar;

 }

 ConstruirTipoBusqueda() {


```

Desplega select;
Desplegar boton Aceptar;
Desplegar boton Cancelar;
}
    
```

C.3 DIAGRAMAS INTERACCIÓN DEL SISTEMA

C.3.1 Caso de uso ValidarAcceso

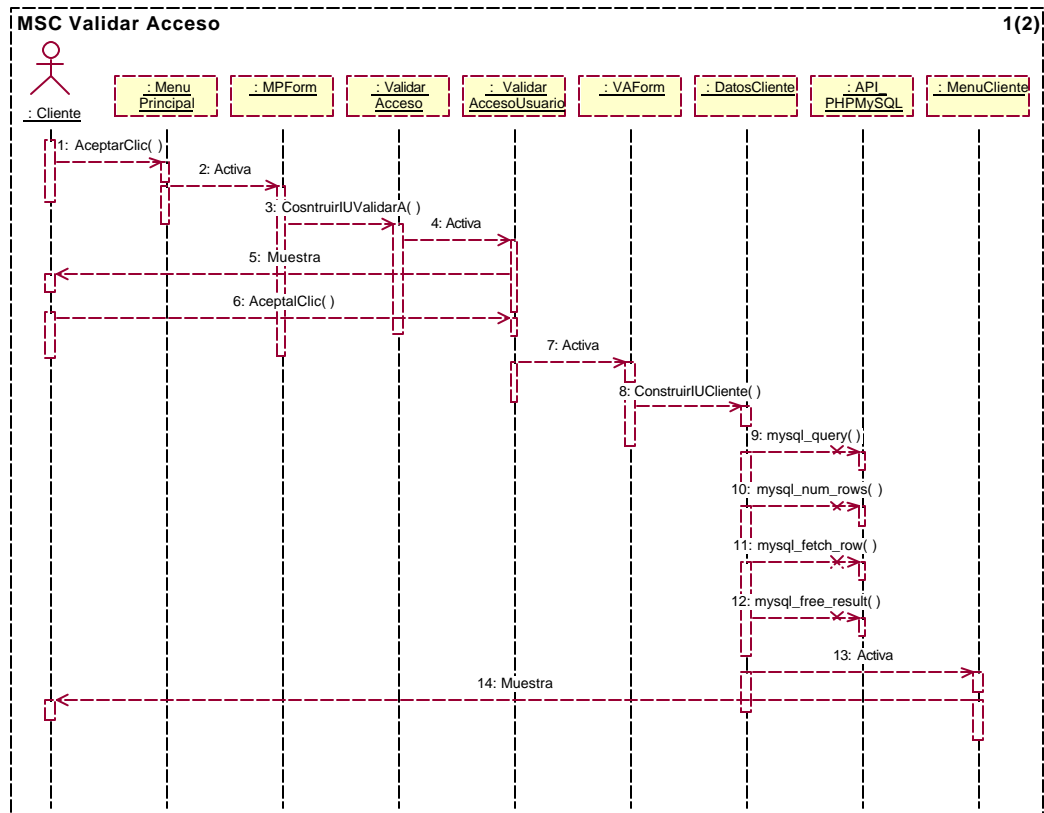


Figura C-17. Carta de secuencias de mensajes Validar acceso Cliente.

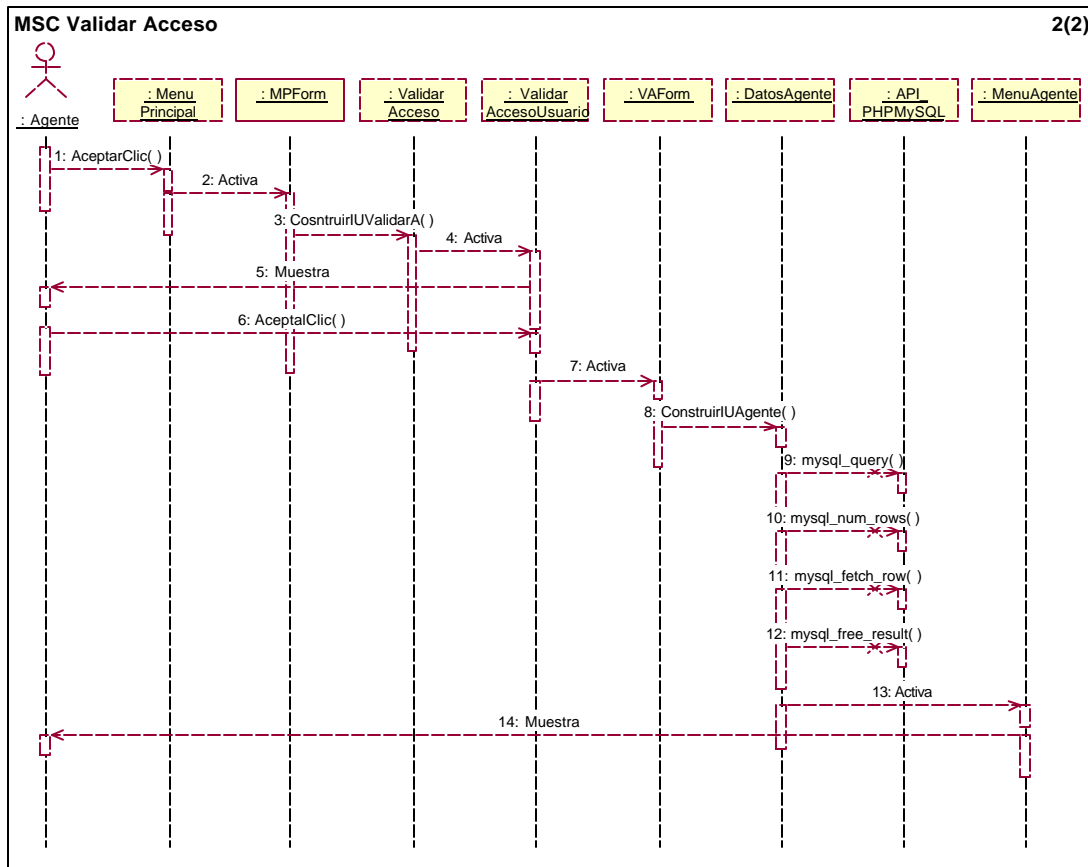


Figura C-18. Carta de secuencia de mensajes Validar acceso, Agente

C.3.2 Caso de uso ModificarContraseña

Ver página siguiente...

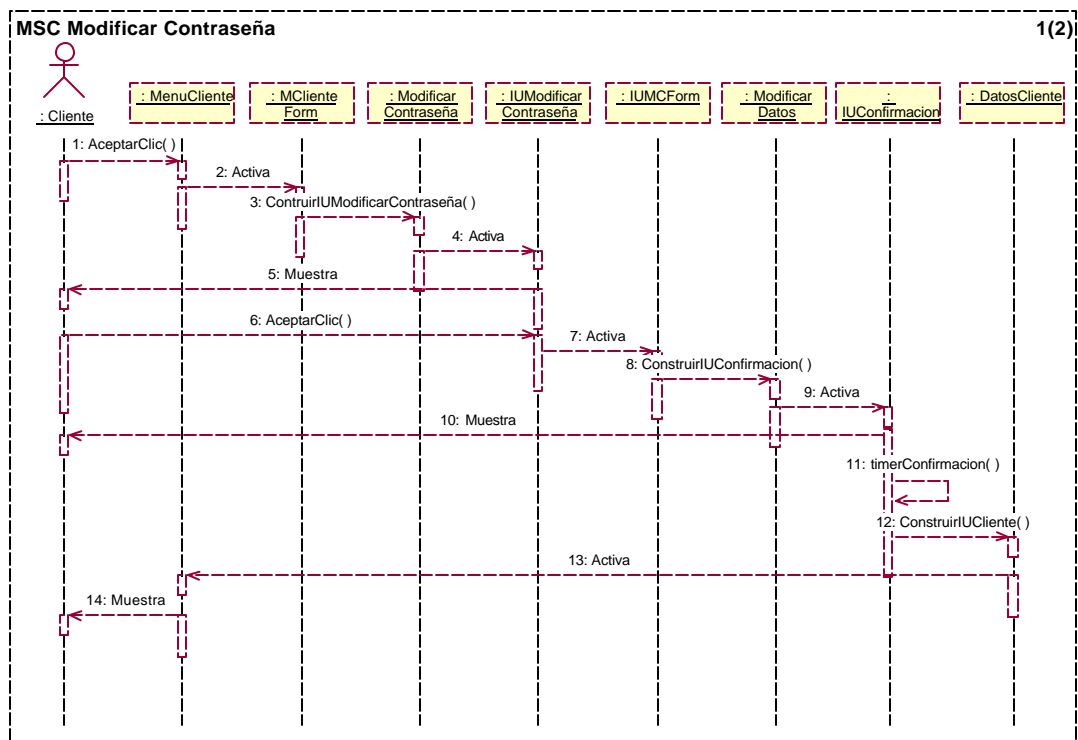


Figura C-19. Carta de secuencia de mensajes Modificar Contraseña Cliente

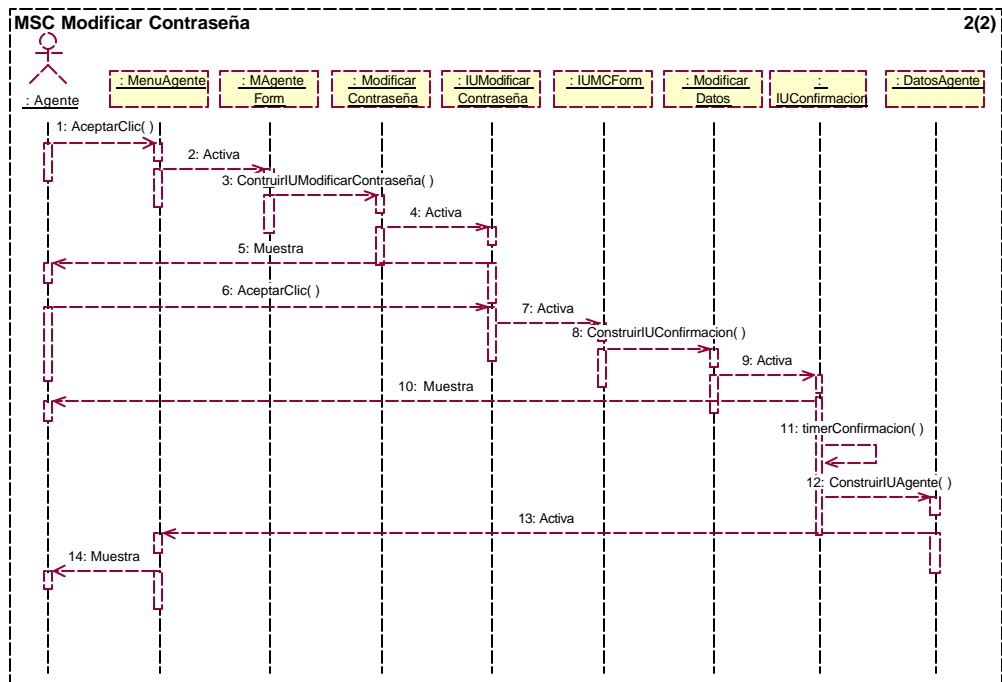


Figura C-20. Carta de secuencia de mensajes Modificar Contraseña Agente

C.3.3 Caso de uso RealizarPedido

Ver página siguiente...

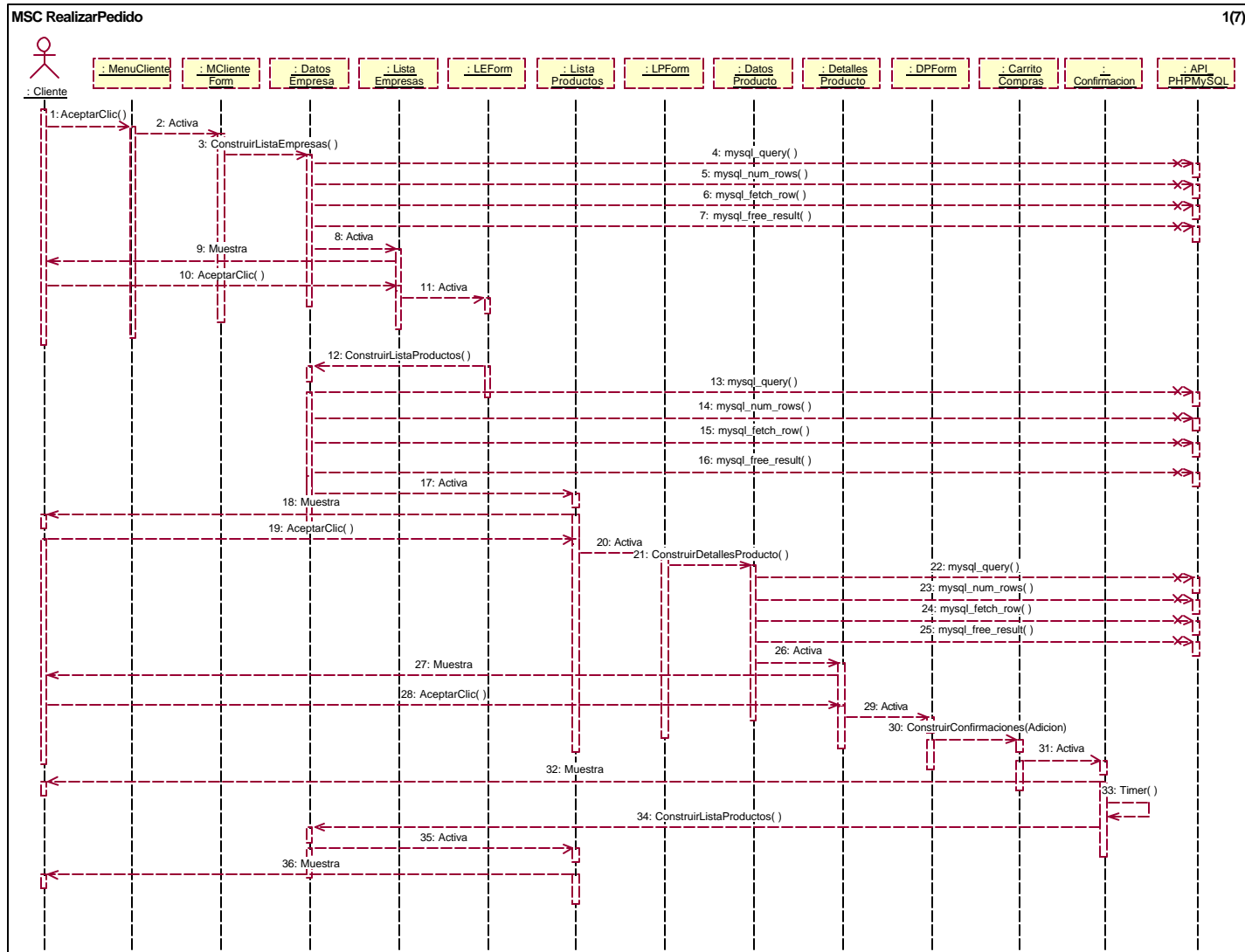


Figura C-21. Carta de secuencia de mensajes Realizar Pedido, Adición de productos al carrito de compras por un Cliente

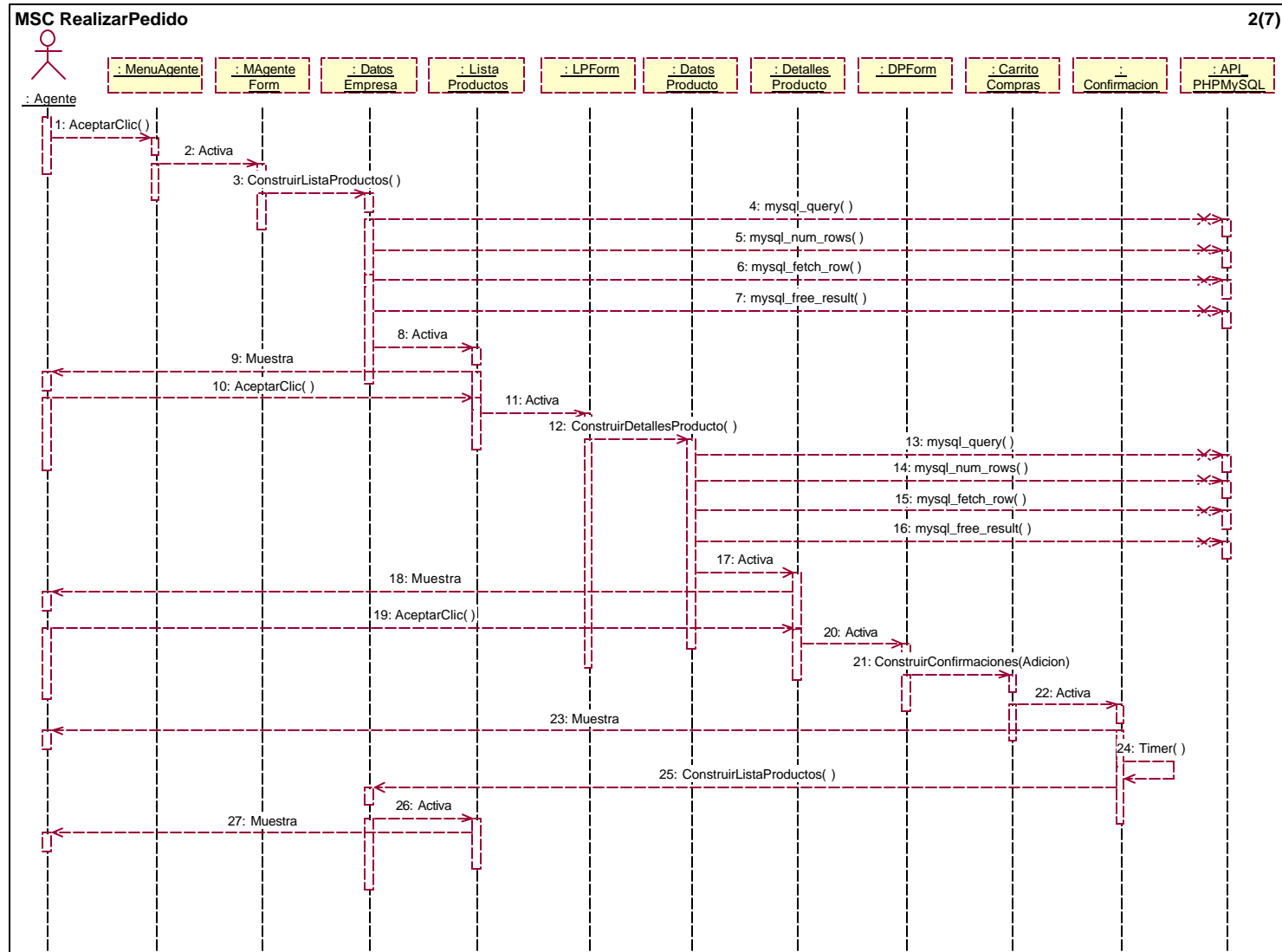


Figura C-22. Carta de secuencia de mensajes Realizar Pedido, Adición de productos al carrito de compras por un Agente

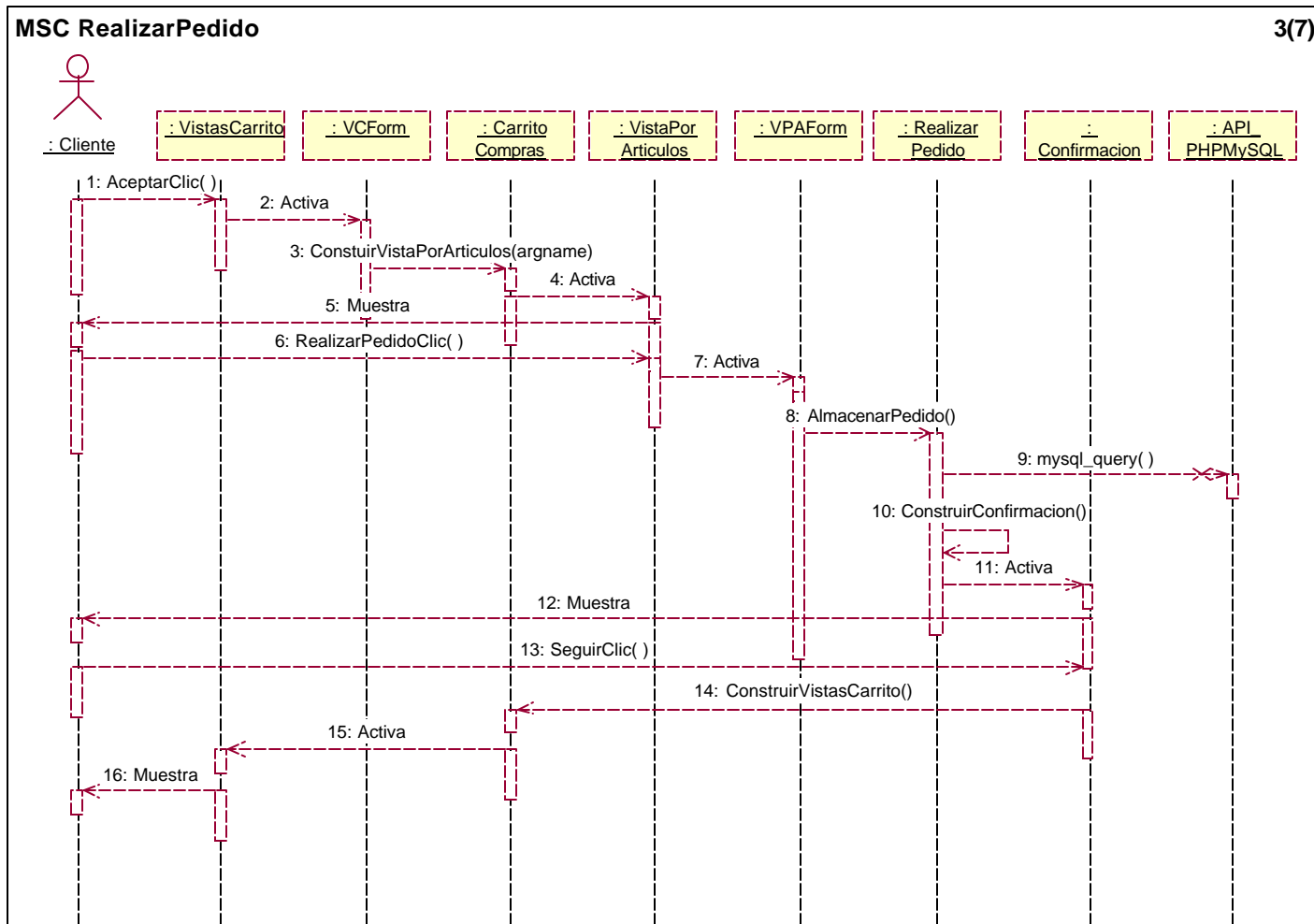


Figura C-23. Carta de secuencia de mensajes Realizar Pedido, enviar pedido por un Cliente

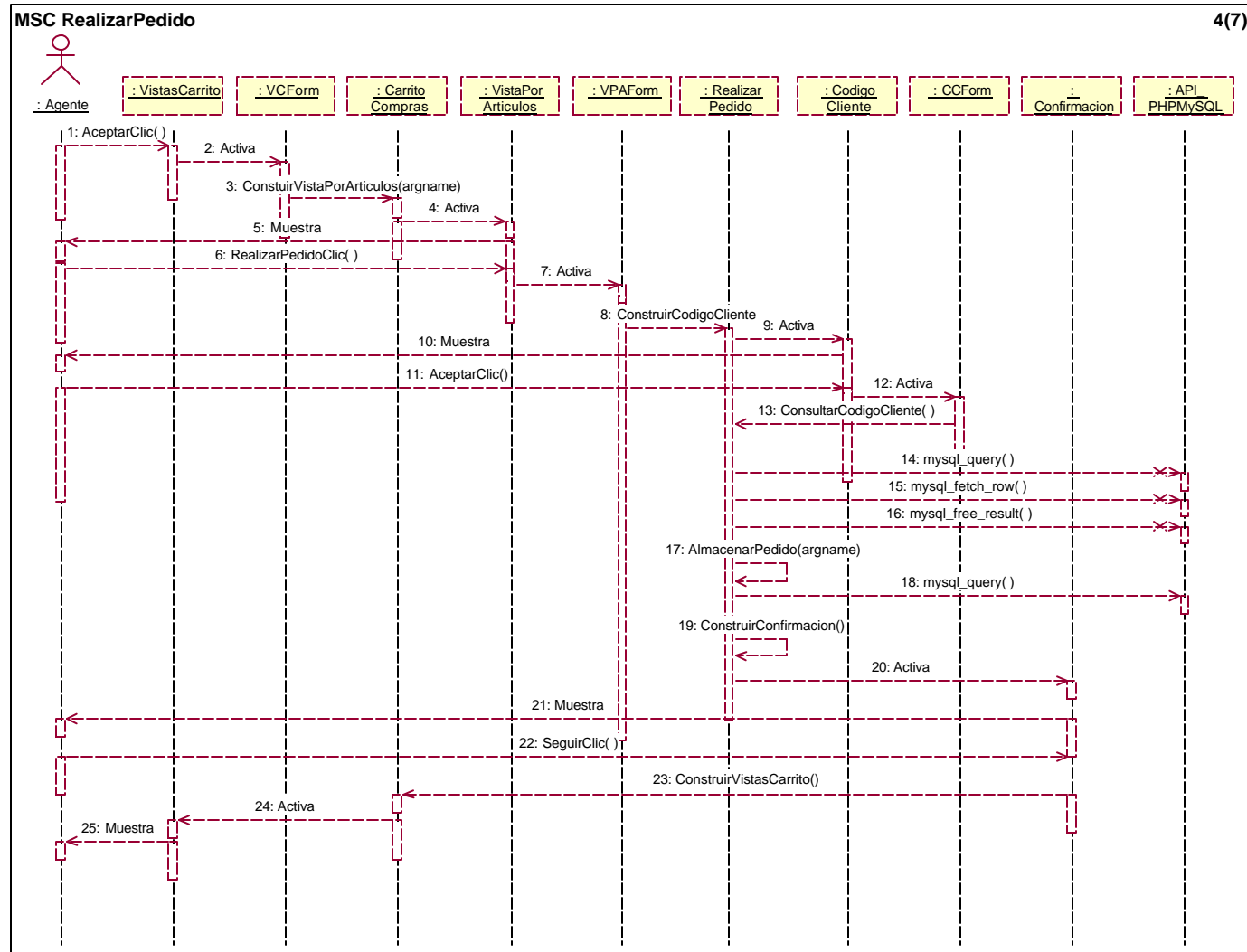


Figura C-24. Carta de secuencia de mensajes Realizar Pedido, enviar pedido por un Agente

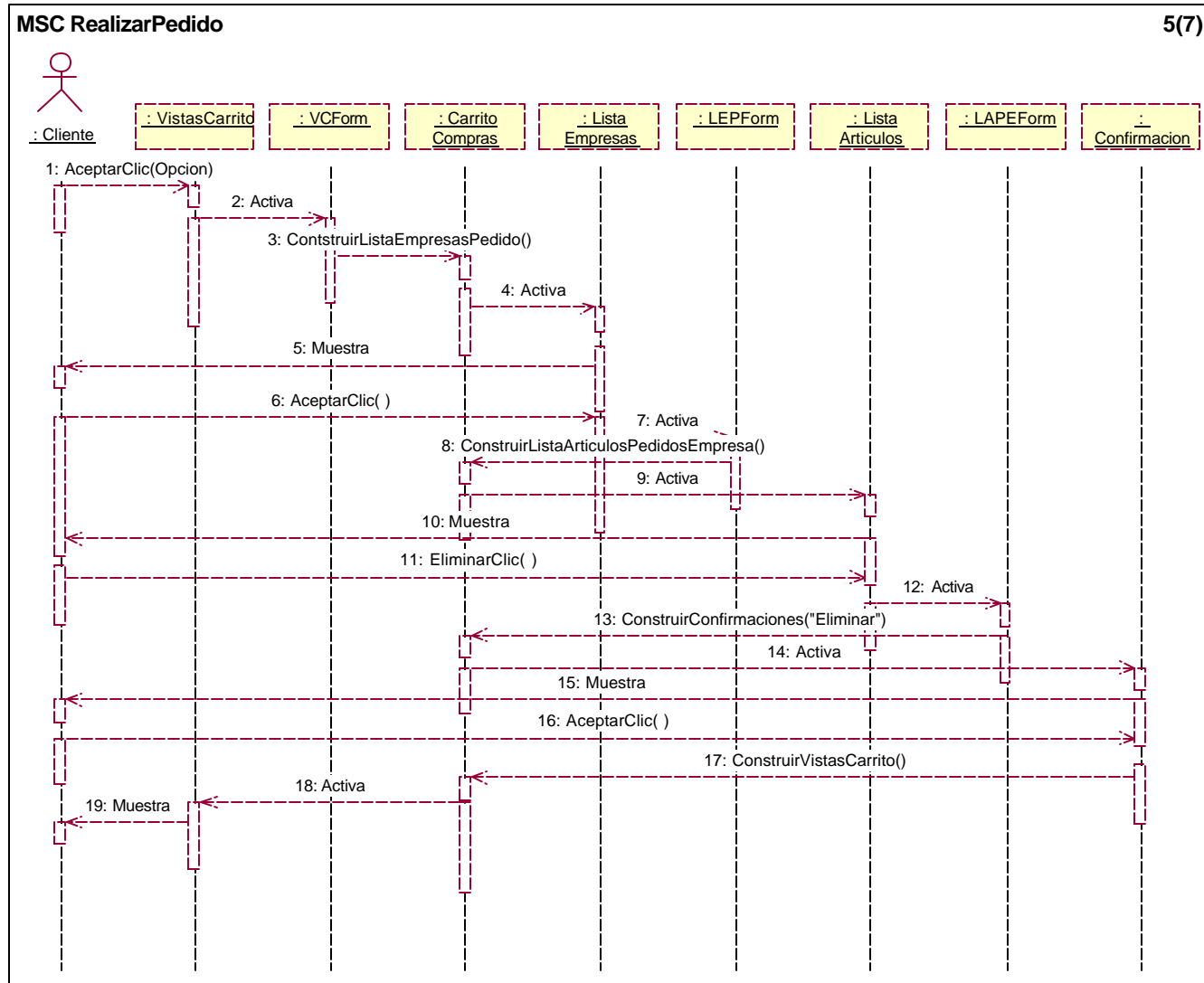


Figura C-25. Carta de secuencia de mensajes Realizar Pedido, eliminar productos del carrito de compras por un Cliente

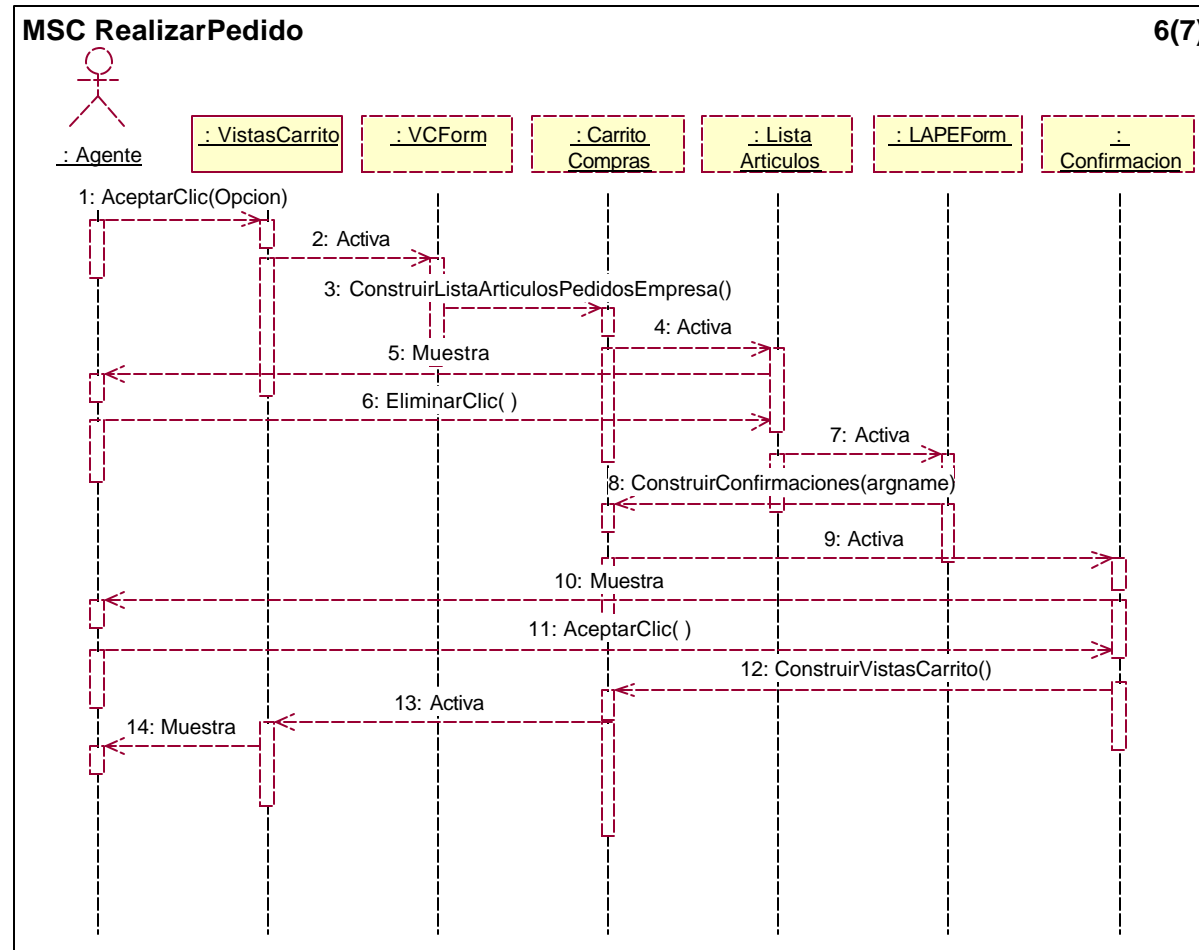


Figura C-26. Carta de secuencia de mensajes Realizar Pedido, eliminar productos del carrito de compras por un Agente

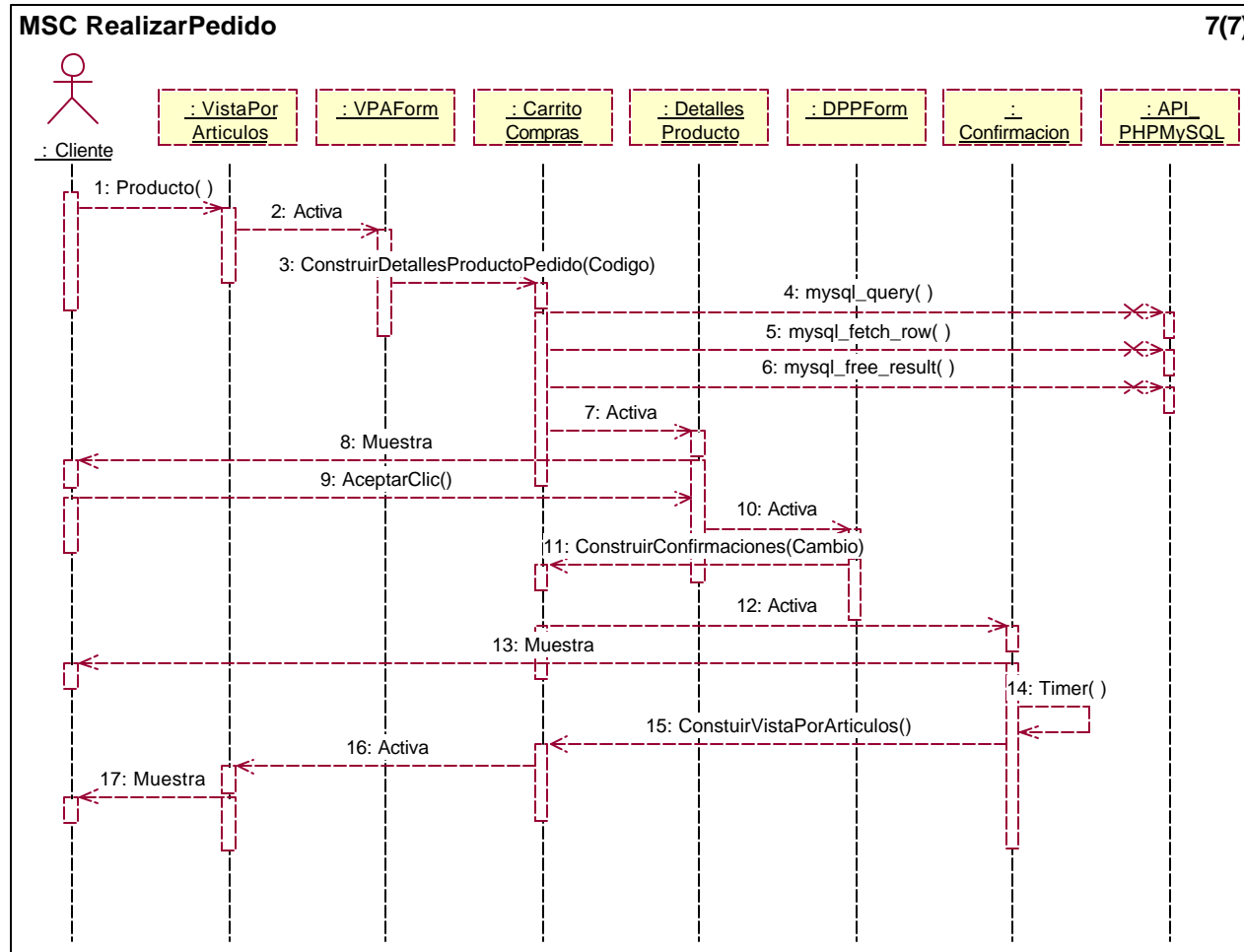


Figura C-27. Carta de secuencia de mensajes Realizar Pedido, modificar cantidad de un producto Cliente/Agente

C.3.3 Caso de uso BuscarPedido

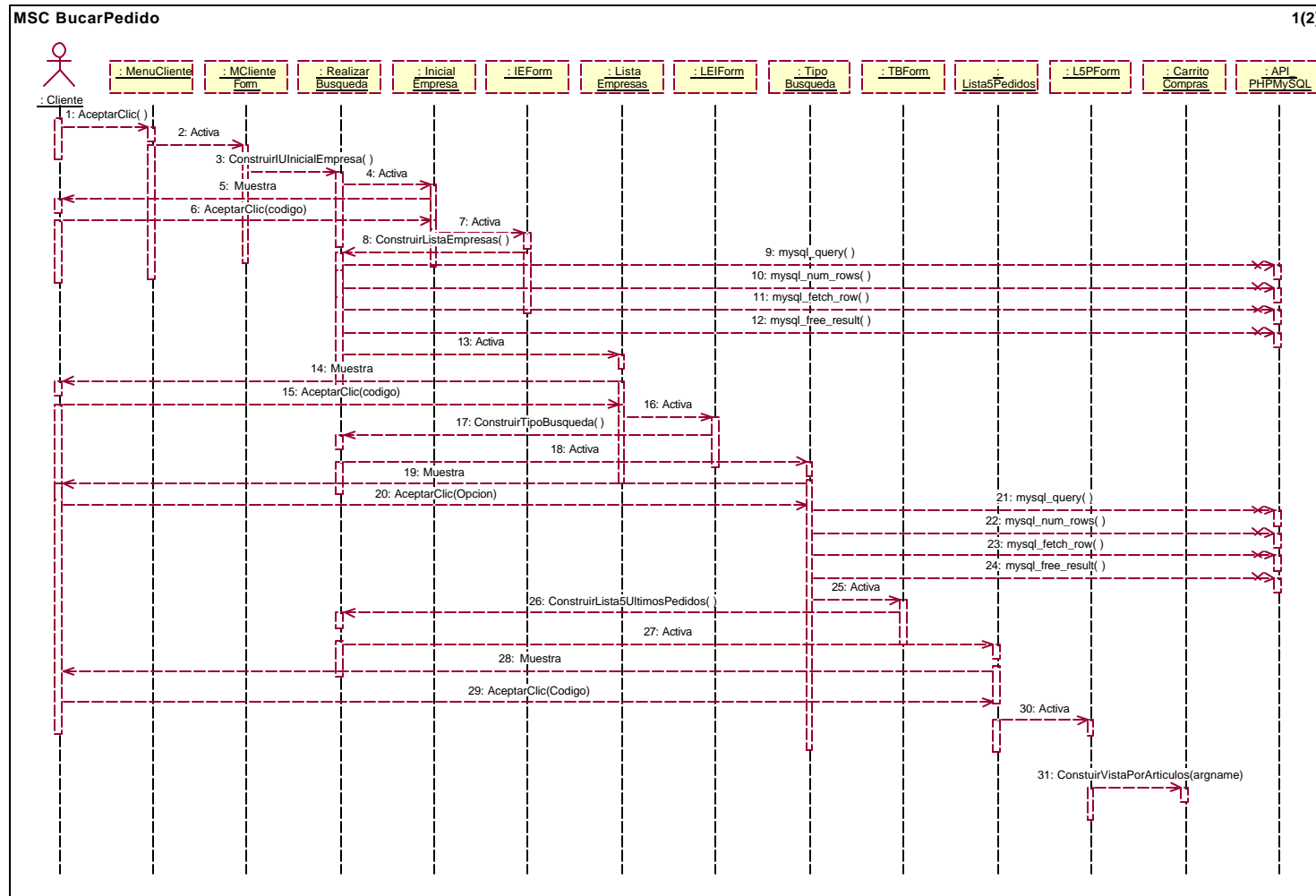


Figura C-28. Carta de secuencia de mensajes Buscar Pedido, Buscar pedido por un Cliente

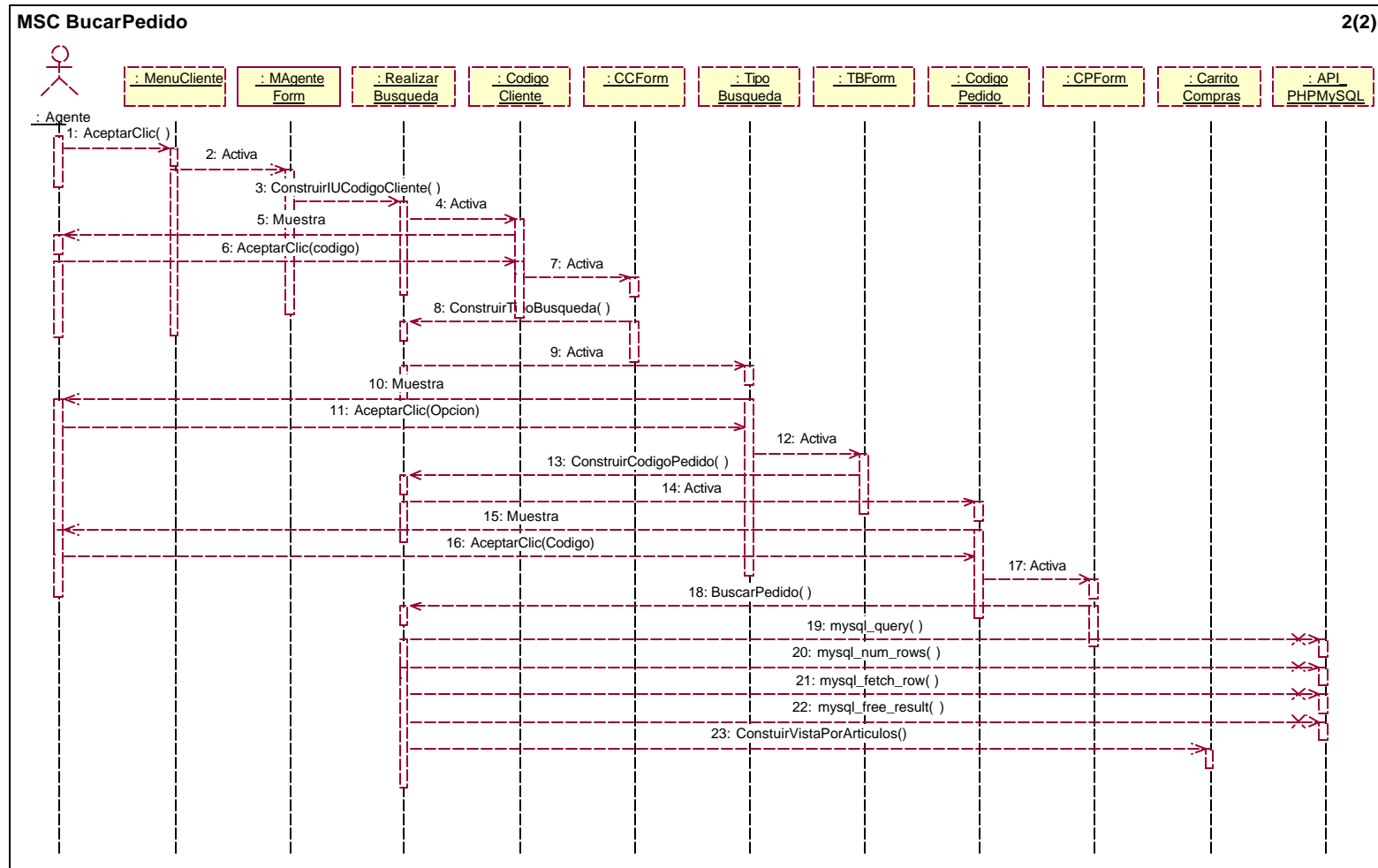


Figura C-29. Carta de secuencia de mensajes Buscar Pedido, Buscar pedido por un Agente

C.4 DIAGRAMA DE IMPLANTACIÓN

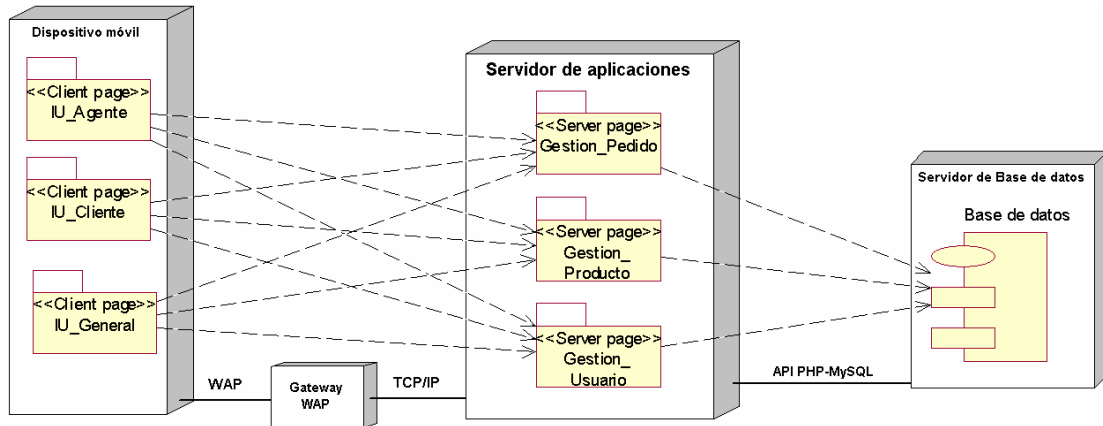


Figura C-30. Diagrama de implantación

Este diagrama ilustra la distribución física de los componentes del sistema. En primer lugar están las Client page desplegadas en el dispositivo móvil, mediante invocaciones realizadas al servidor de aplicación; en segundo lugar se encuentra una gateway WAP la cual hace posible el intercambio de información entre la red fija y la red móvil. Seguidamente está el servidor de aplicaciones el cual alberga las Server page encargadas de construir las Client page. Finalmente se encuentra el servidor de base de datos el cual podría estar implementado en el mismo servidor de aplicación; sin embargo se ilustra de forma separada para describir el caso más general. A continuación se describe en detalle cada uno de estos componentes.

Dispositivo móvil: El dispositivo para el cual fue desarrollada la aplicación, es el R280d de Ericsson, la descripción técnica del dispositivo es la siguiente:

Tipo de acceso	CDPD, TDMA ² .
Versión del navegador	Openwave 3.1

² En Colombia se hace a través de CDPD.

Resolución de la pantalla	101 x 40 pixels
Tamaño del despliegue	3 filas x 15 columnas
Máximo tamaño de. PDU, (Protocol Data Unit)	1492

Tabla C-1. Descripción técnica del R280d de Ericsson.

Servidor de aplicaciones y de base de datos: Es un PC con sistema operativo windows2000 professional, con Apache Web Server, como servidor web; MySQL, como gestor de base de datos y PHP, como lenguaje *scripting* de WML embebido.

En el diagrama también indican los protocolos de comunicación entre componentes hardware. Entre el dispositivo móvil y la gateway se utiliza WAP, (Wireless Application Protocol) y entre la gateway y el servidor de aplicaciones se utiliza TCP/IP. Debido a que el servidor de base de datos se encuentra en el mismo servidor de aplicaciones se utiliza el API de PHP para MySQL, para la comunicación entre la aplicación y la base de datos.

C.5 DISEÑO DE LA BASE DE DATOS

A partir de las entidades descritas en el “Análisis del software”, se construyó el siguiente diagrama de entidad relación:

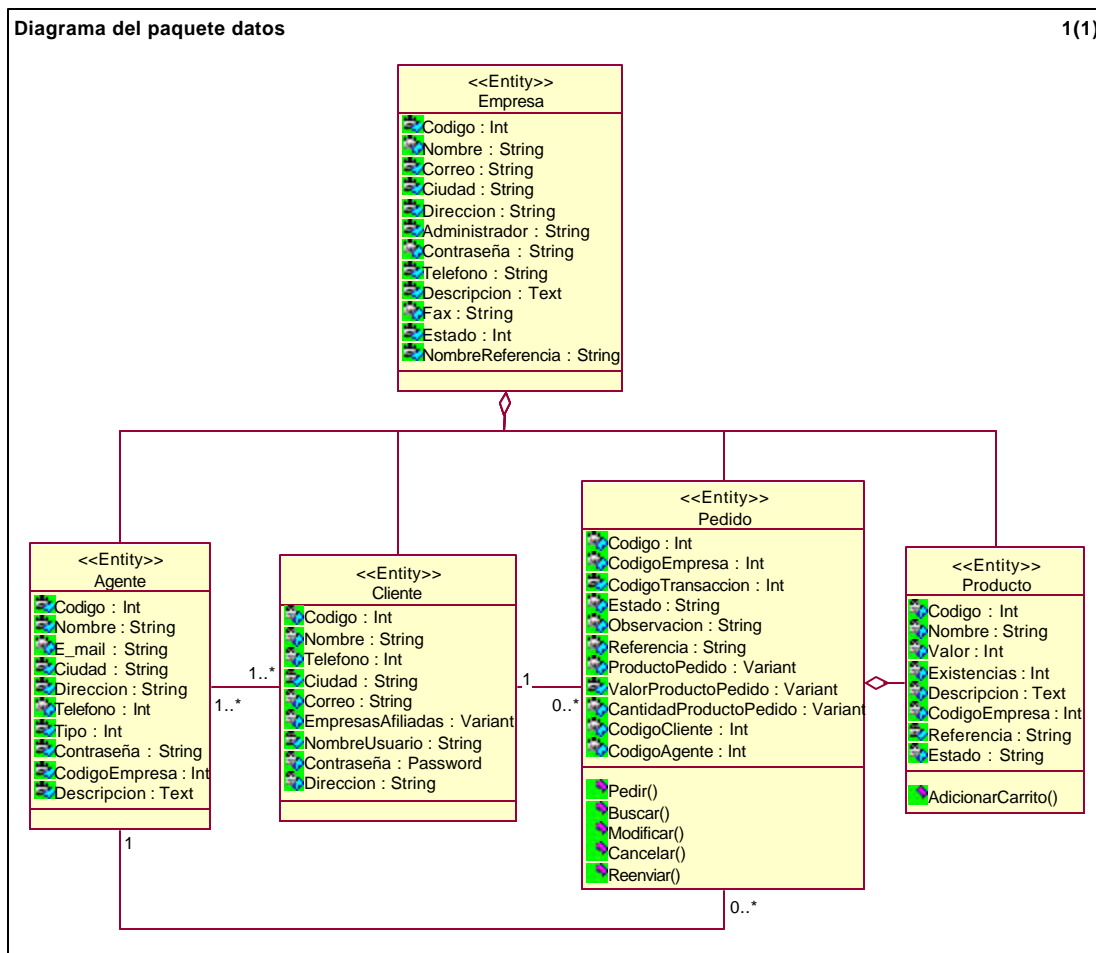


Figura C-31. Diagrama del paquete de datos, "Análisis del Software"

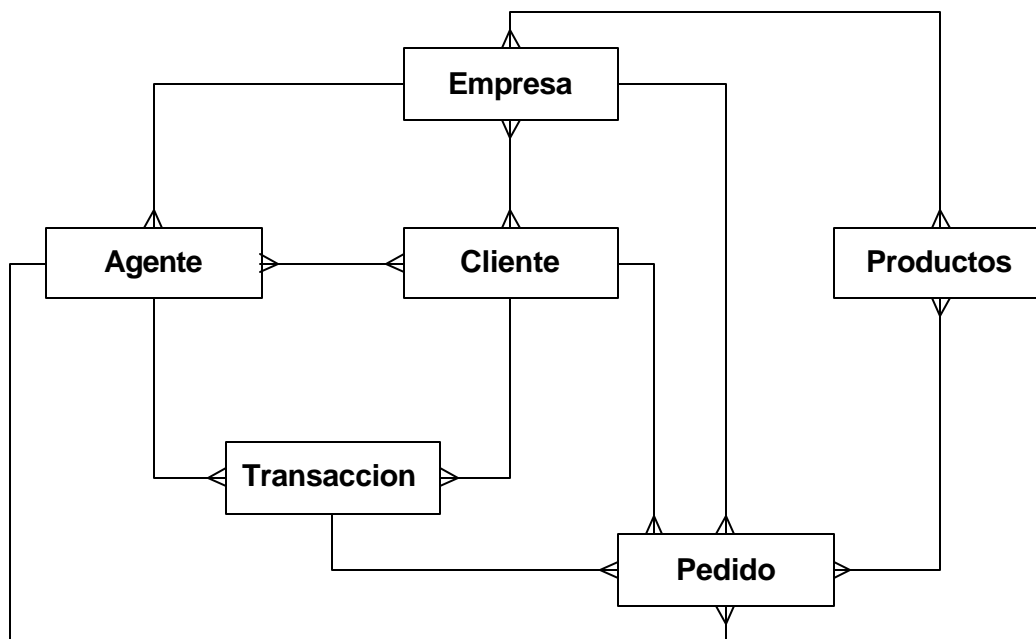


Figura C-32. Diagrama Entidad-Relación

Destruyendo las relaciones muchos a muchos finalmente la base de datos es como sigue:

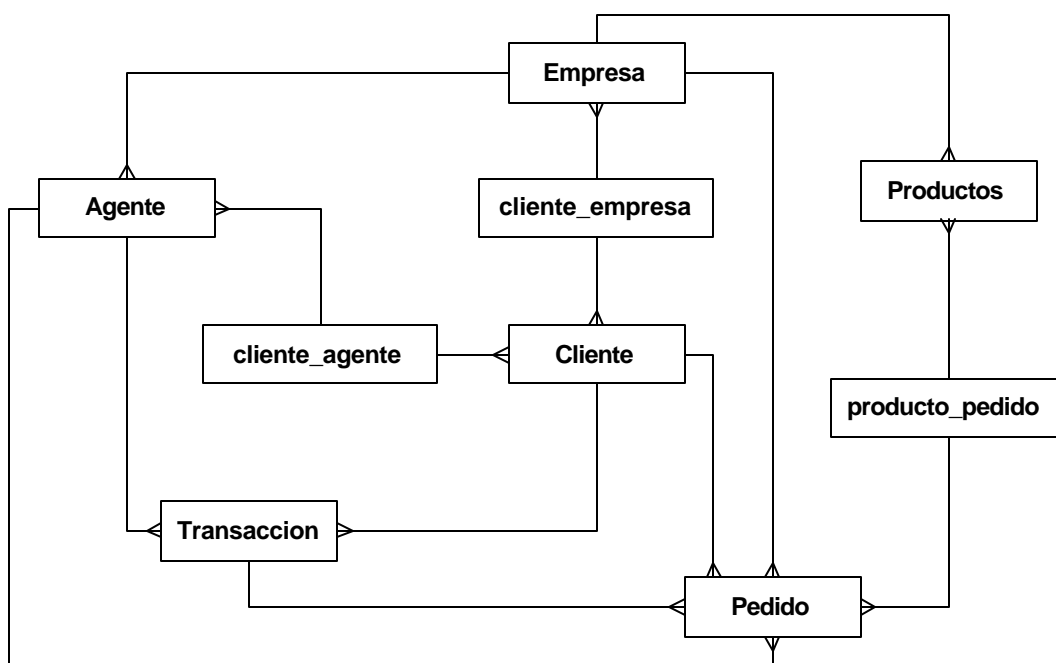


Figura C-33. Diagrama Entidad-Relación, normalizado

Los atributos para cada tabla son, (los campos subrayados corresponden a la clave primaria de la tabla):

adminsx

Campo	Tipo
<u>Codigo</u>	int(11)
Nombre	varchar(100)
IdCiudad	int(11)
Login	varchar(15)
Password	varchar(15)
Direccion	varchar(100)
Telefono	varchar(30)
E_mail	varchar(50)
Estado	int(11)

Tabla C-2. Tabla adminsx, (Administrador del sistema)

agente

Campo	Tipo
<u>Codigo</u>	int(11)
CodigoEmpresa	int(11)
IdCiudad	int(11)
Nombre	char(50)
Login	char(30)
Password	char(30)
Tipo	enum('Visitador', 'Repartidor')
Direccion	char(30)
Telefono	char(20)
E_mail	char(50)

Tabla C-3. Tabla agente

Cliente

Campo	Tipo
<u>Codigo</u>	int(11)
IdCiudad	int(11)
Nombre	char(50)
Login	char(30)
Password	char(30)
Direccion	char(30)
Telefono	char(20)
E_mail	char(50)

Tabla C-4. Tabla Cliente

cliente_agente

Campo	Tipo
CodigoCliente	int(11)
CodigoAgente	int(11)

Tabla C-5. Tabla cliente_agente

cliente_empresa

Campo	Tipo
CodigoCliente	int(11)
CodigoEmpresa	int(11)
<u>Id</u>	tinyint(4)

Tabla C-6. Tabla cliente_empresa

empresa

Campo	Tipo
<u>Codigo</u>	int(11)
Nombre	varchar(100)
Administrador	varchar(80)
IdCiudad	int(11)
Password	varchar(15)
Descripcion	text
Direccion	varchar(100)
Telefono	varchar(30)
E_mail	varchar(50)
Fax	varchar(30)
Estado	int(11)
NombreRef	varchar(4)

Tabla C-7. Tabla empresa

pedido

Campo	Tipo
<u>Codigo</u>	int(11)
CodigoEmpresa	int(11)
CodigoTransaccion	int(11)
Estado	int(11)
Observacion	text
Referencia	varchar(30)

Tabla C-8. Tabla pedido

productopedido

Campo	Tipo
CodigoProducto	int(11)
CodigoPedido	int(11)
Cantidad	int(11)
Precio	int(11)

Tabla C-10. Tabla productopedido

producto

Campo	Tipo
<u>Codigo</u>	int(11)
CodigoEmpresa	int(11)
Referencia	varchar(30)
Nombre	varchar(30)
Descripcion	text
Valor	int(11)
Existencias	int(11)
Estado	int(11)

Tabla C-9. Tabla producto

transaccion

Campo	Tipo
<u>Codigo</u>	Int(11)
CodigoCliente	Int(11)
Fecha	date
CodigoAgente	Int(11)

Tabla C-11. Tabla transaccion