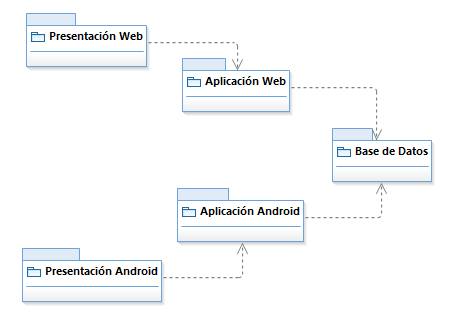
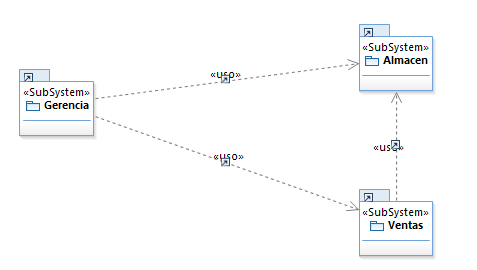
**Diseño de estructura**

[Describa el diseño desde el más alto nivel. Esto se suele hacer con un diagrama que muestra una arquitectura en capas.]



**Subsistemas**

**Ventas**



En este Subsitema se encuentran los casos de Uso:

Registro de Venta

Emitir Comprobante

Generar Proforma

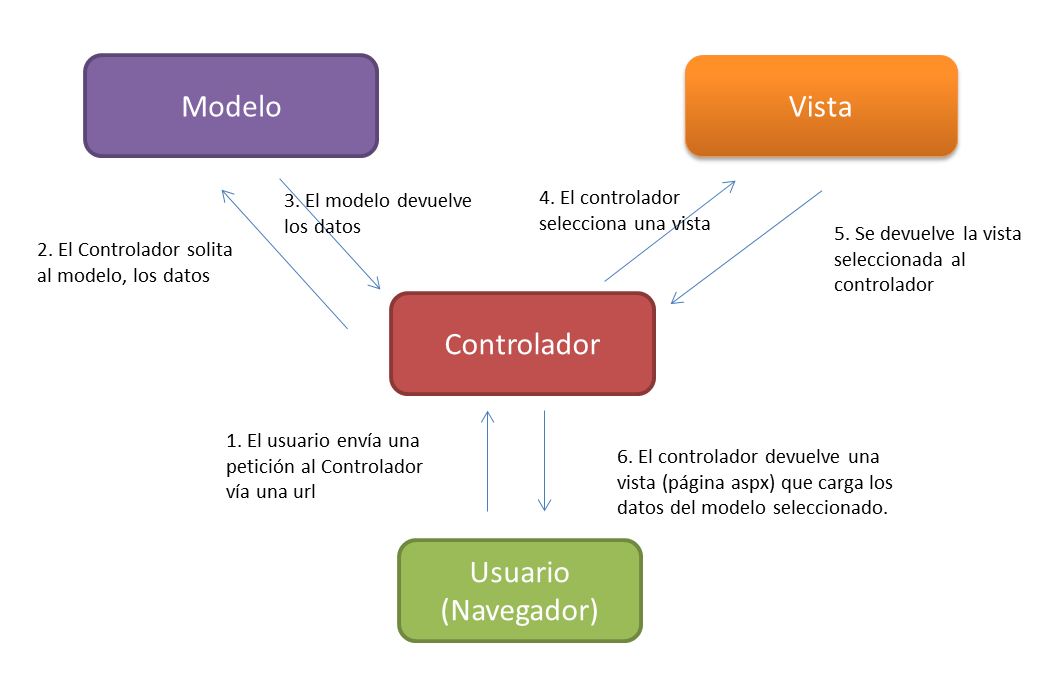
**Patrones**

**MVC**

**Descripción general**

El patrón de arquitectura MVC (Modelo Vista Controlador) es un patrón que define la organización independiente del Modelo (Objetos de Negocio), la Vista (interfaz con el usuario u otro sistema) y el Controlador (controlador del workflow de la aplicación)

**Estructura**



**Comportamiento**

* El usuario interactúa con la interfaz de usuario de alguna forma (por ejemplo, el usuario pulsa un botón, enlace, etc.)
* El controlador recibe (por parte de los objetos de la [interfaz](https://es.wikipedia.org/wiki/Interfaz_de_usuario)-vista) la notificación de la acción solicitada por el usuario. El controlador gestiona el evento que llega, frecuentemente a través de un gestor de eventos (handler) o callback.
* El controlador accede al modelo, actualizándolo, posiblemente modificándolo de forma adecuada a la acción solicitada por el usuario (por ejemplo, el controlador actualiza el carro de la compra del usuario).
* El controlador delega a los objetos de la vista la tarea de desplegar la interfaz de usuario. La vista obtiene sus datos del modelo para generar la interfaz apropiada para el usuario donde se reflejan los cambios en el modelo (por ejemplo, produce un listado del contenido del carro de la compra).
* El modelo no debe tener conocimiento directo sobre la vista. Sin embargo, se podría utilizar el patrón [Observador](https://es.wikipedia.org/wiki/Observer_%28patr%C3%B3n_de_dise%C3%B1o%29) para proveer cierta indirección entre el modelo y la vista, permitiendo al modelo notificar a los interesados de cualquier cambio.
* La interfaz de usuario espera nuevas interacciones del usuario, comenzando el ciclo nuevamente

Ejemplo

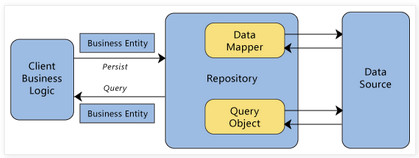
[A menudo, puede transmitir la naturaleza de la modalidad mejor con un ejemplo concreto.]

**REPOSITORY**

**Descripción general**

El patrón repositorio o repository está íntimamente relacionado con el acceso a datos y nos permite tener una abstracción de la implementación de acceso a datos en nuestras aplicaciones, de modo que nuestra lógica de negocio no conozca ni esté acoplada a la fuente de datos. En pocas palabras esto quiere decir que el repositorio actúa como un intermediario entre nuestra lógica de negocio y nuestra lógica de acceso a datos para que se centralice en un solo punto, y de esta forma se logre evitar redundancia de código.

**Estructura**



**Comportamiento**

* Uno de los escenarios donde más se suele usar el patrón repositorio, es cuando tenemos múltiples fuentes de datos.
* obtenemos gran parte de la información de una base de datos relacional Sql Server, obtenemos otros datos desde un servicio web y obtenemos otros desde una base de datos No Sql por ejemplo. en este contexto el patrón repositorio nos ayudara a centralizar el acceso a cada una de estas fuentes de datos para que nuestros aplicativos cliente no tengan que lidiar con este engorroso trabajo

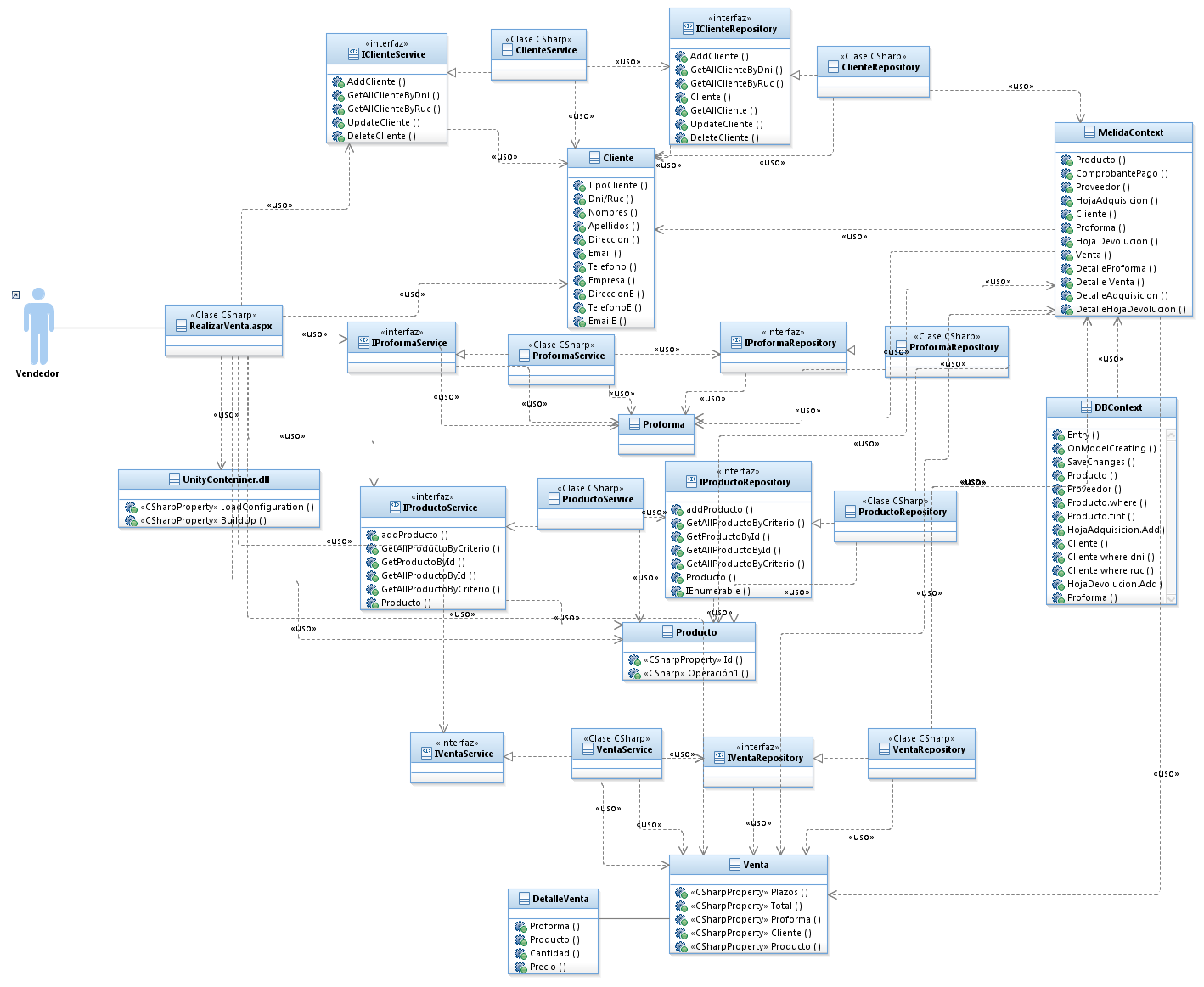
**Requisito realizaciones**

**Realizar Venta**

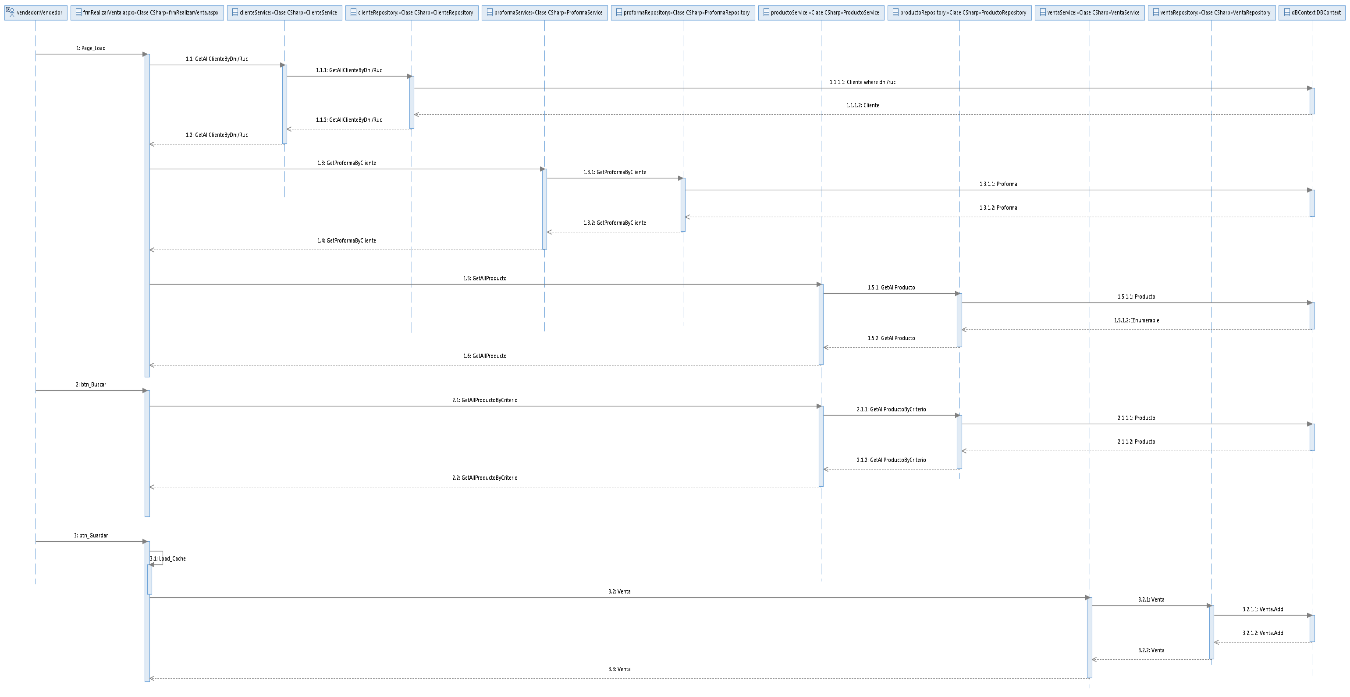
**Opinión de los participantes**

**En este caso de uso se describe los pasos de realizar una venta**

**Escenario básico**



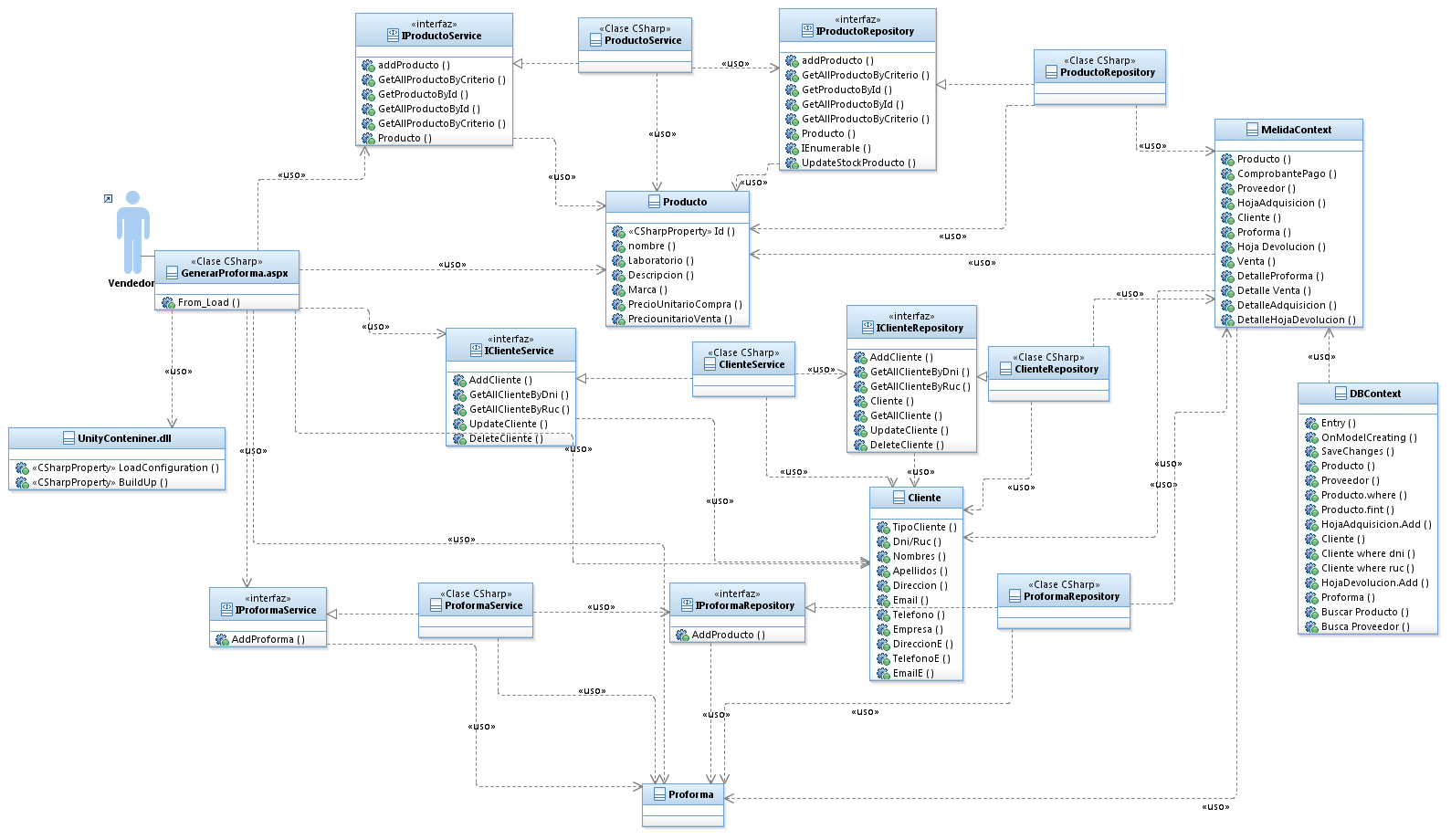
**Escenarios adicionales**



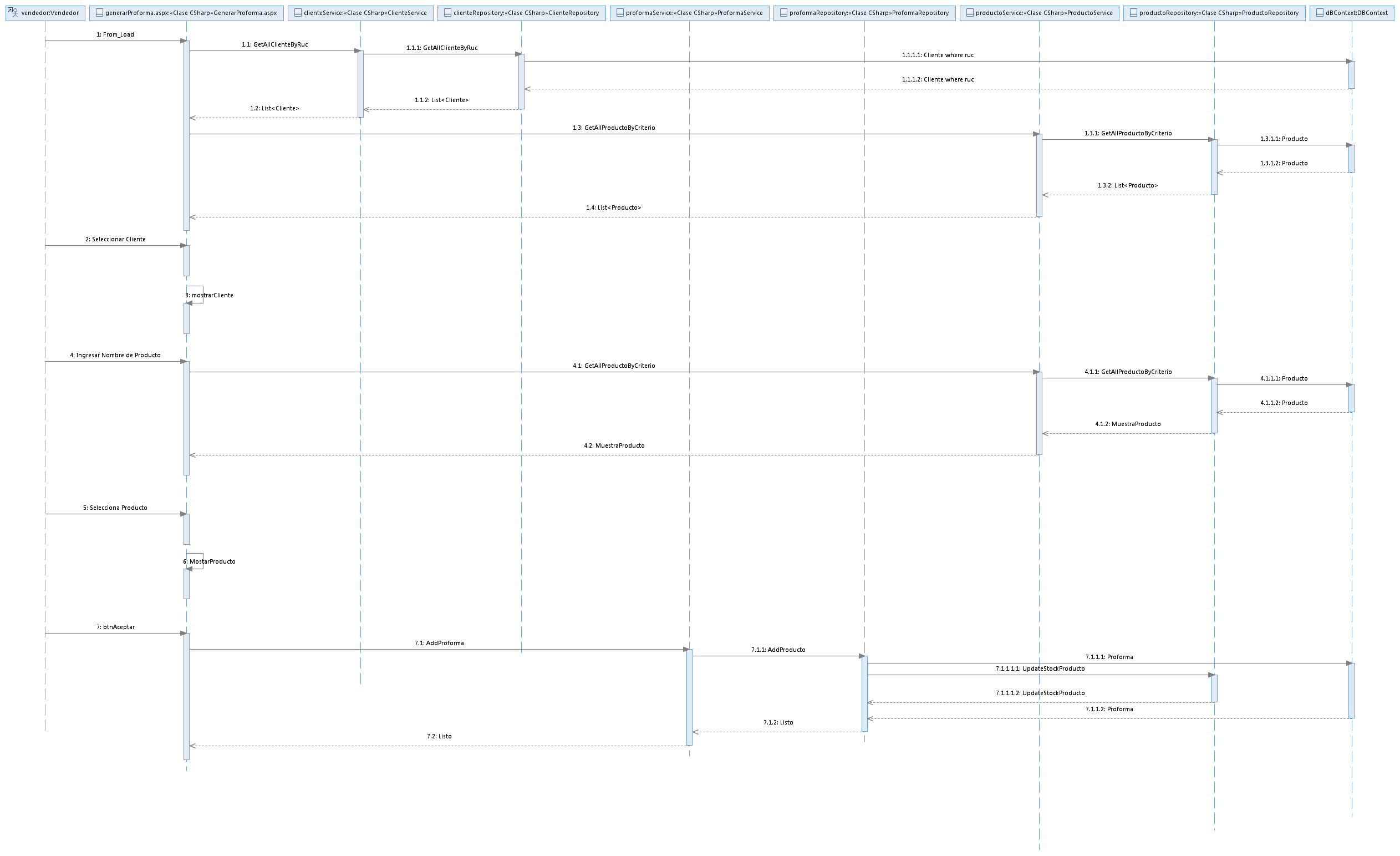
**Generar Proforma**

**Opinión de los participantes**

**Escenario básico**



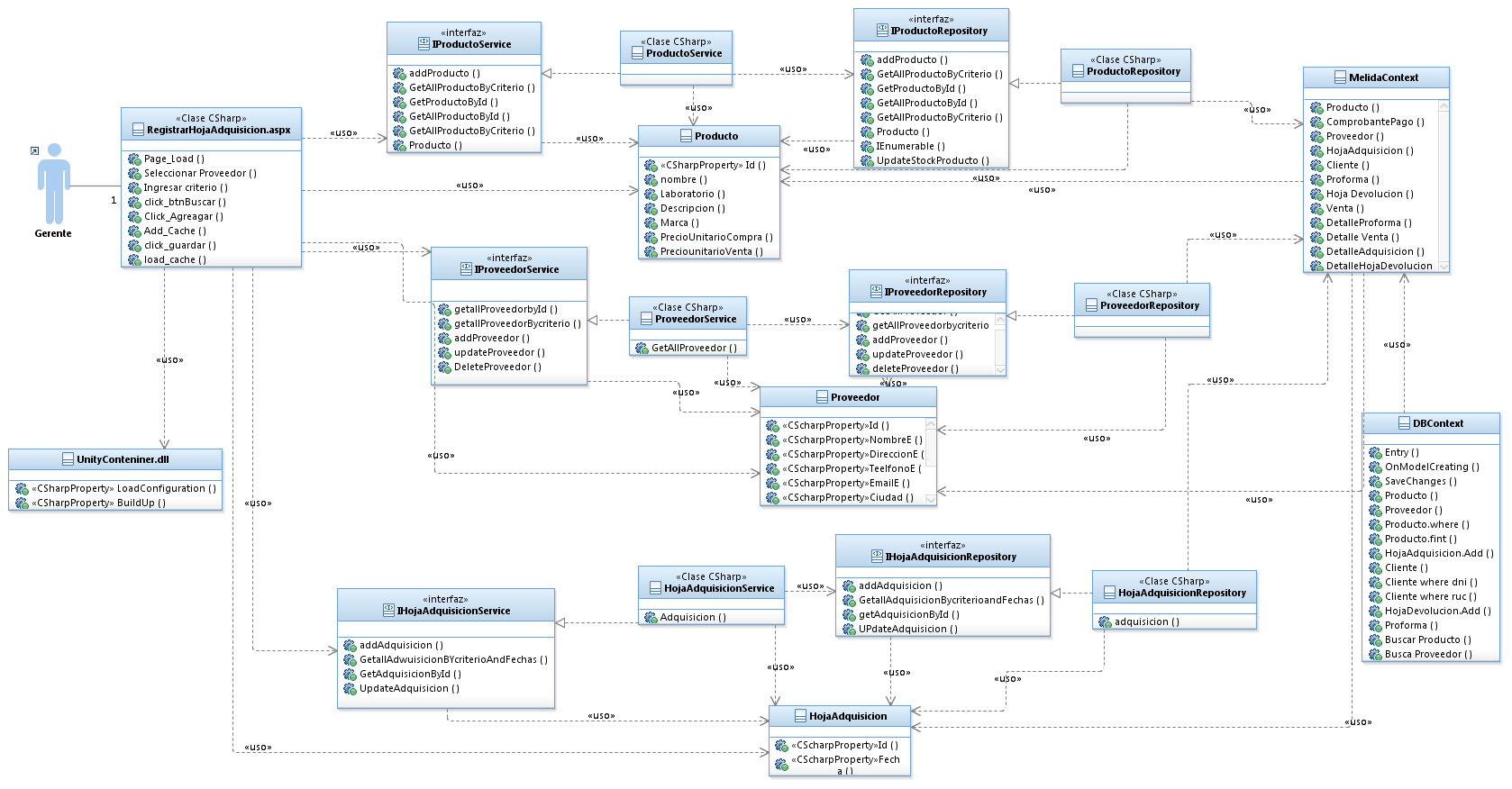
**Escenarios adicionales**



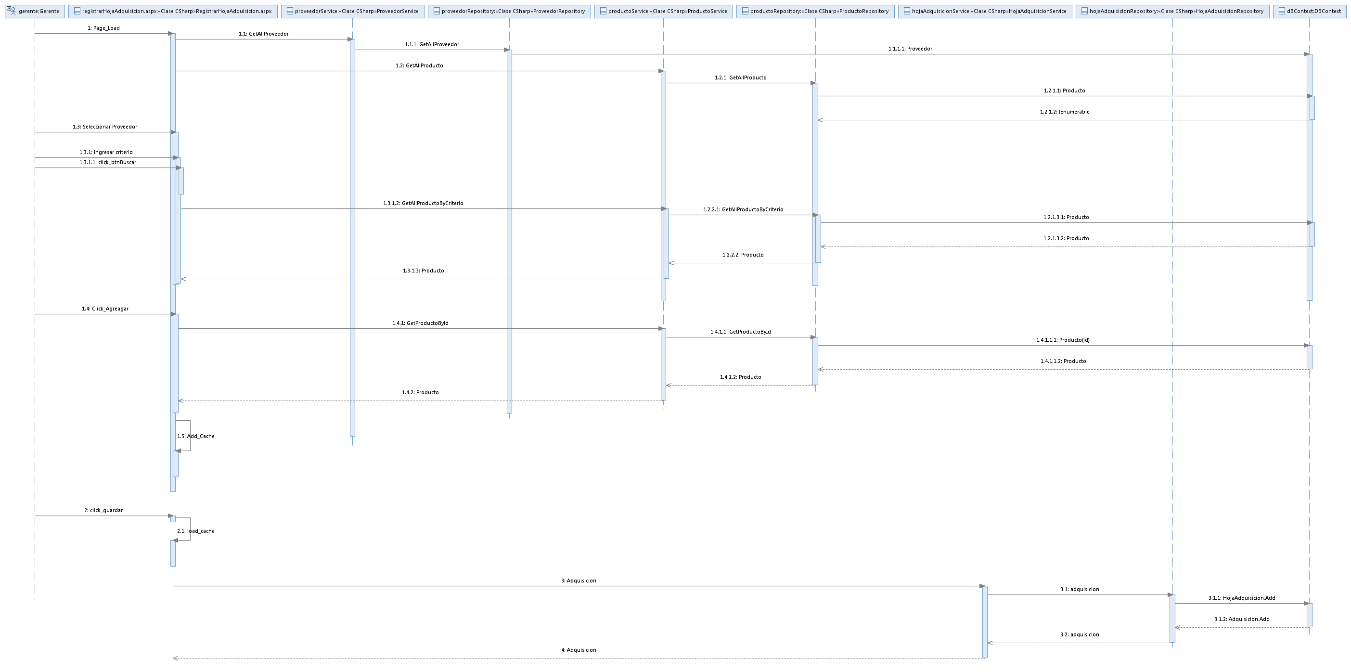
**Realizar Adquisición**

**Opinión de los participantes**

**Escenario básico**



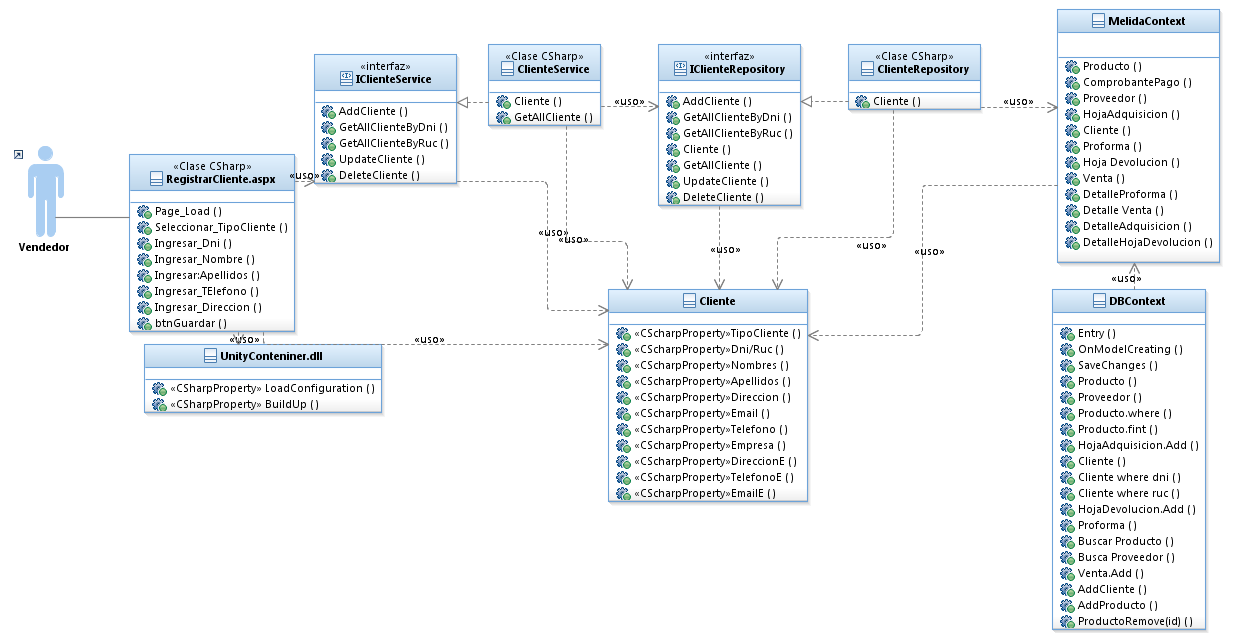
**Escenarios adicionales**



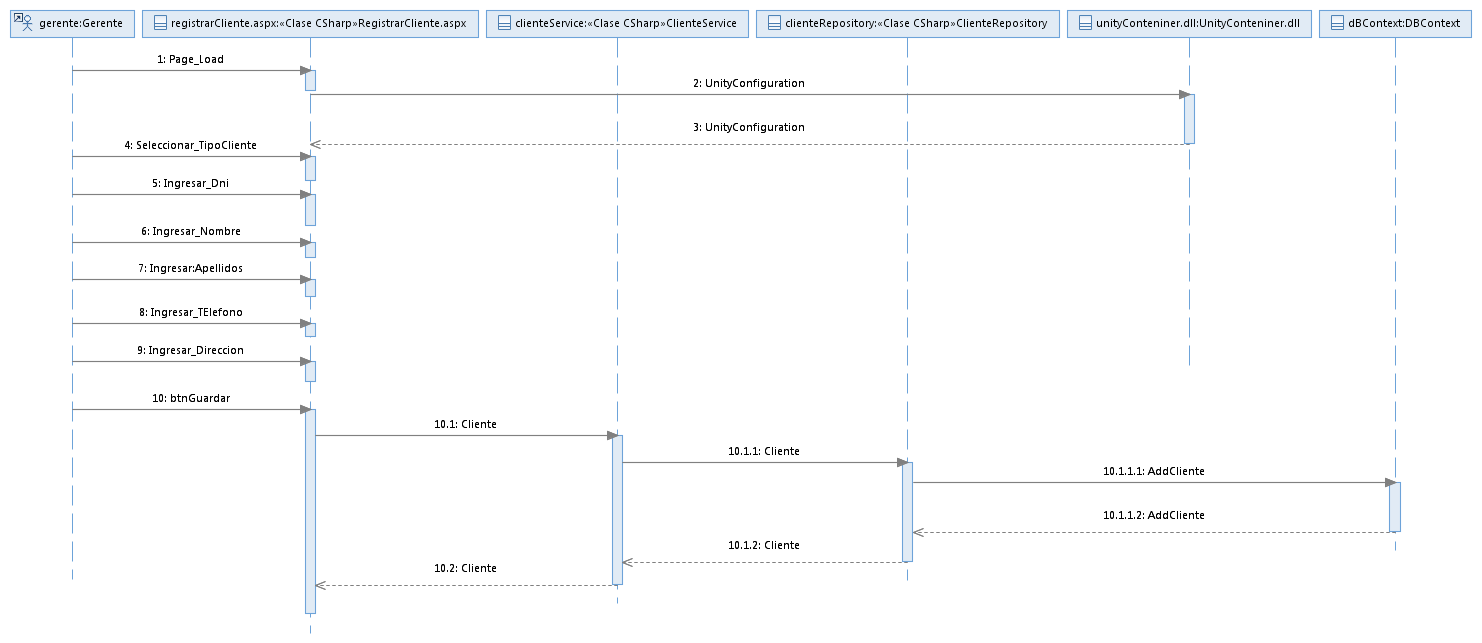
**Registrar Cliente**

**Opinión de los participantes**

**Escenario básico**



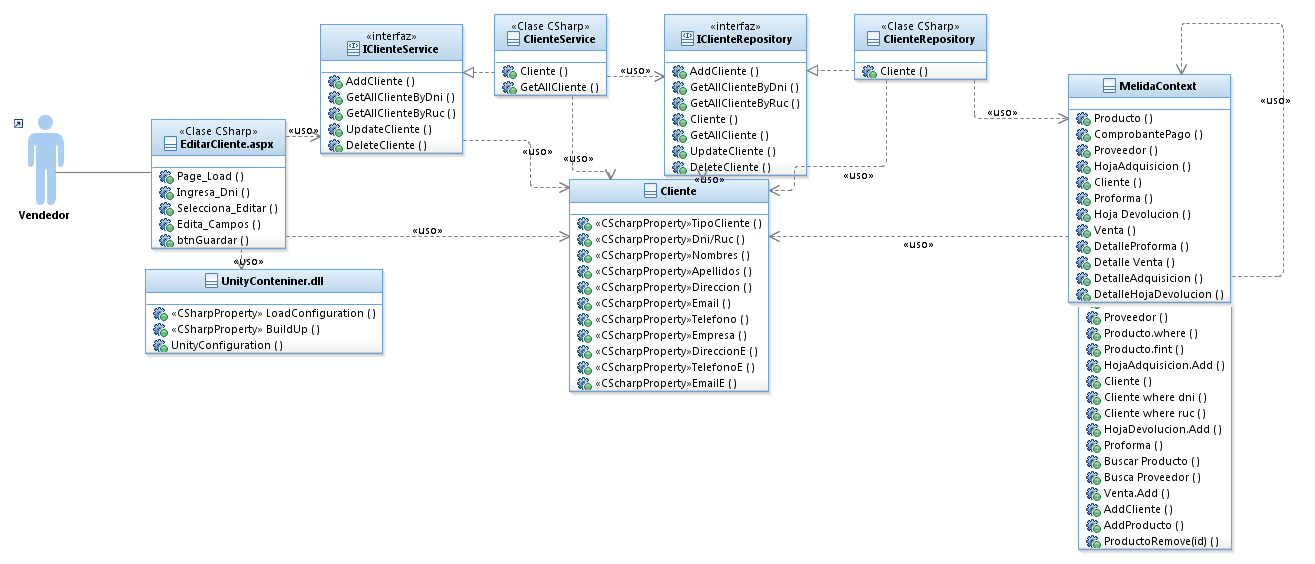
**Escenarios adicionales**



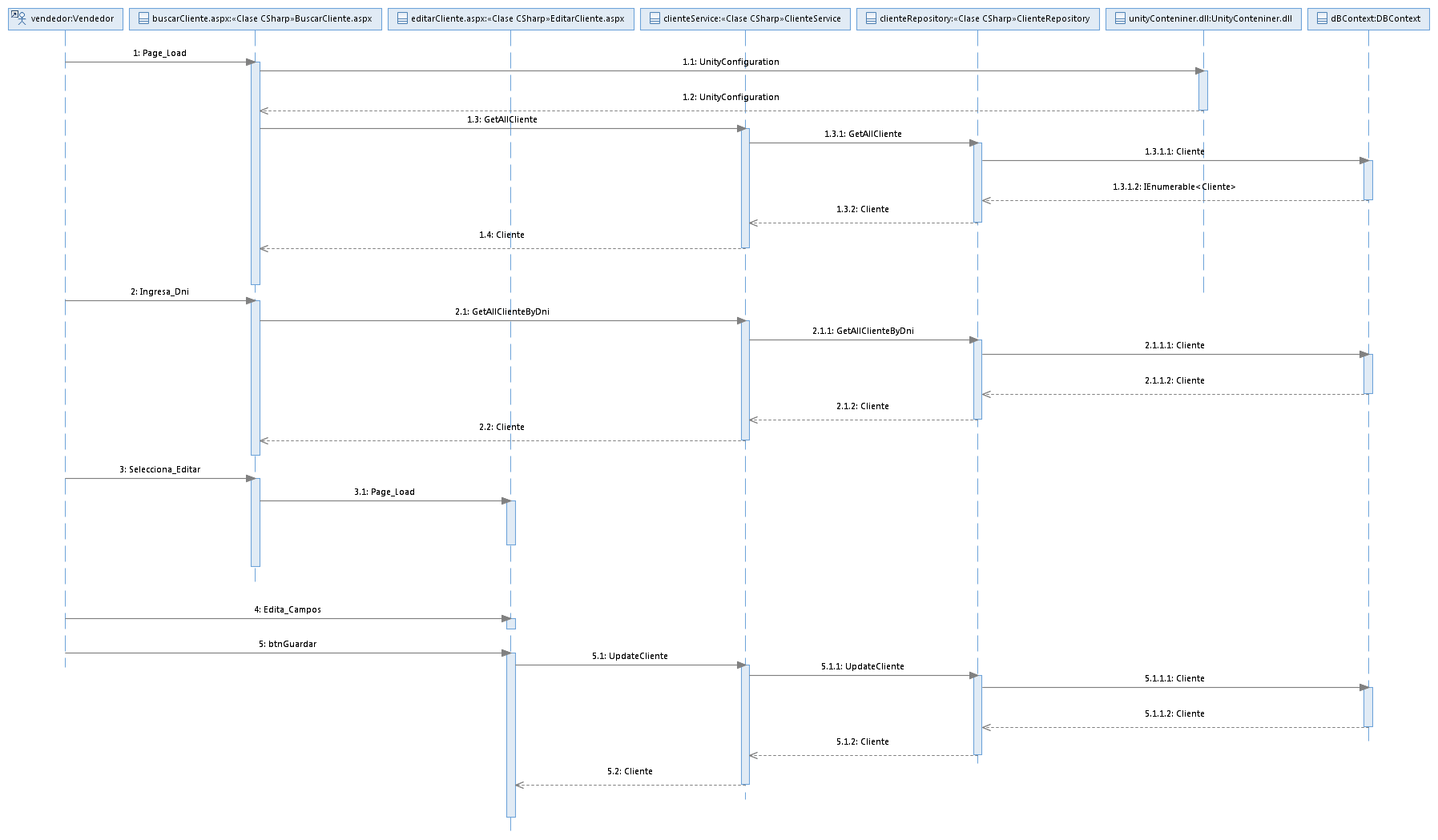
**Editar Cliente**

**Opinión de los participantes**

**Escenario básico**



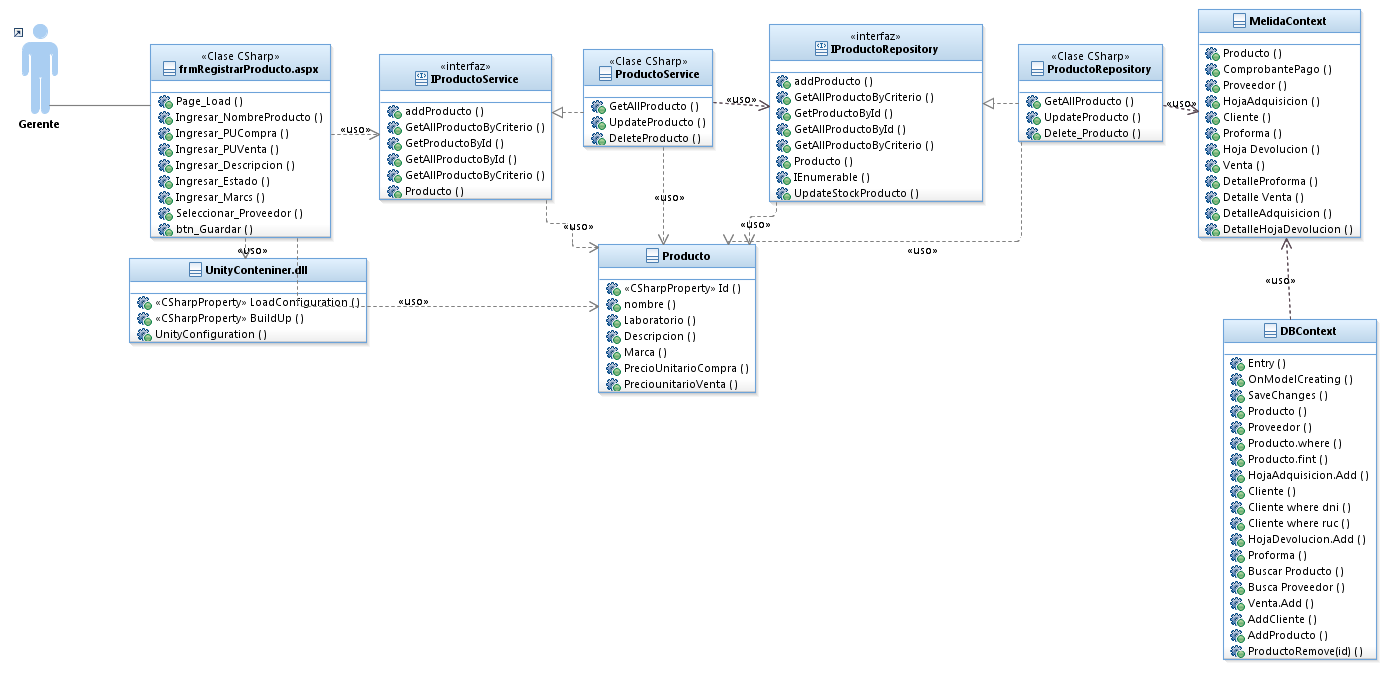
**Escenarios adicionales**



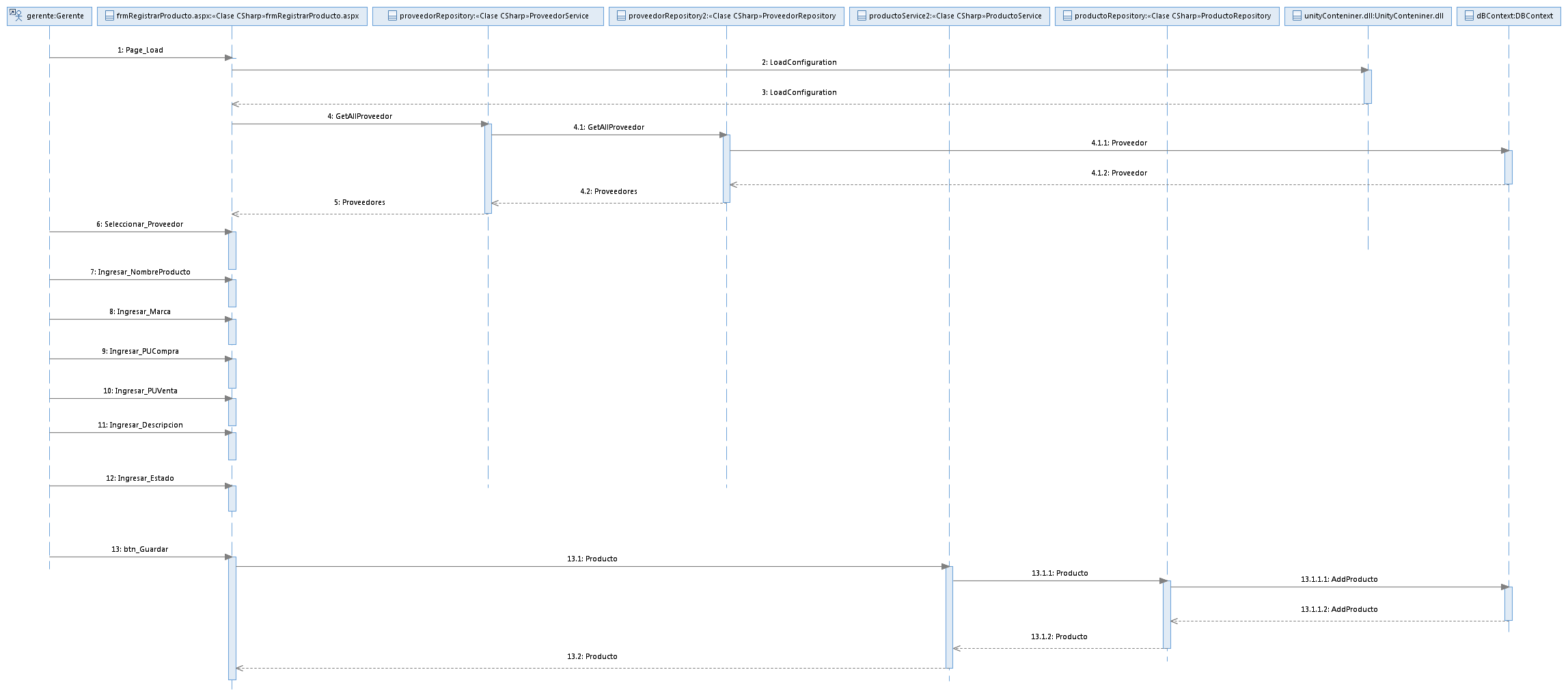
**Registrar Producto**

**Opinión de los participantes**

**Escenario básico**



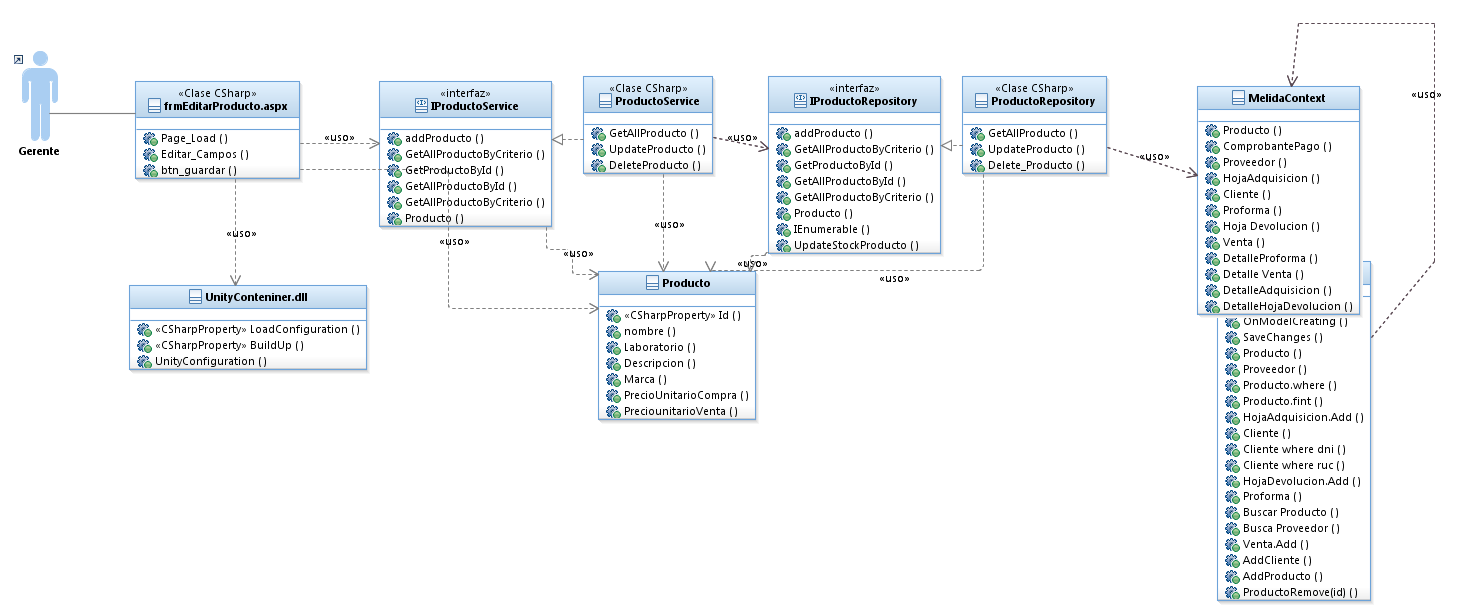
**Escenarios adicionales**



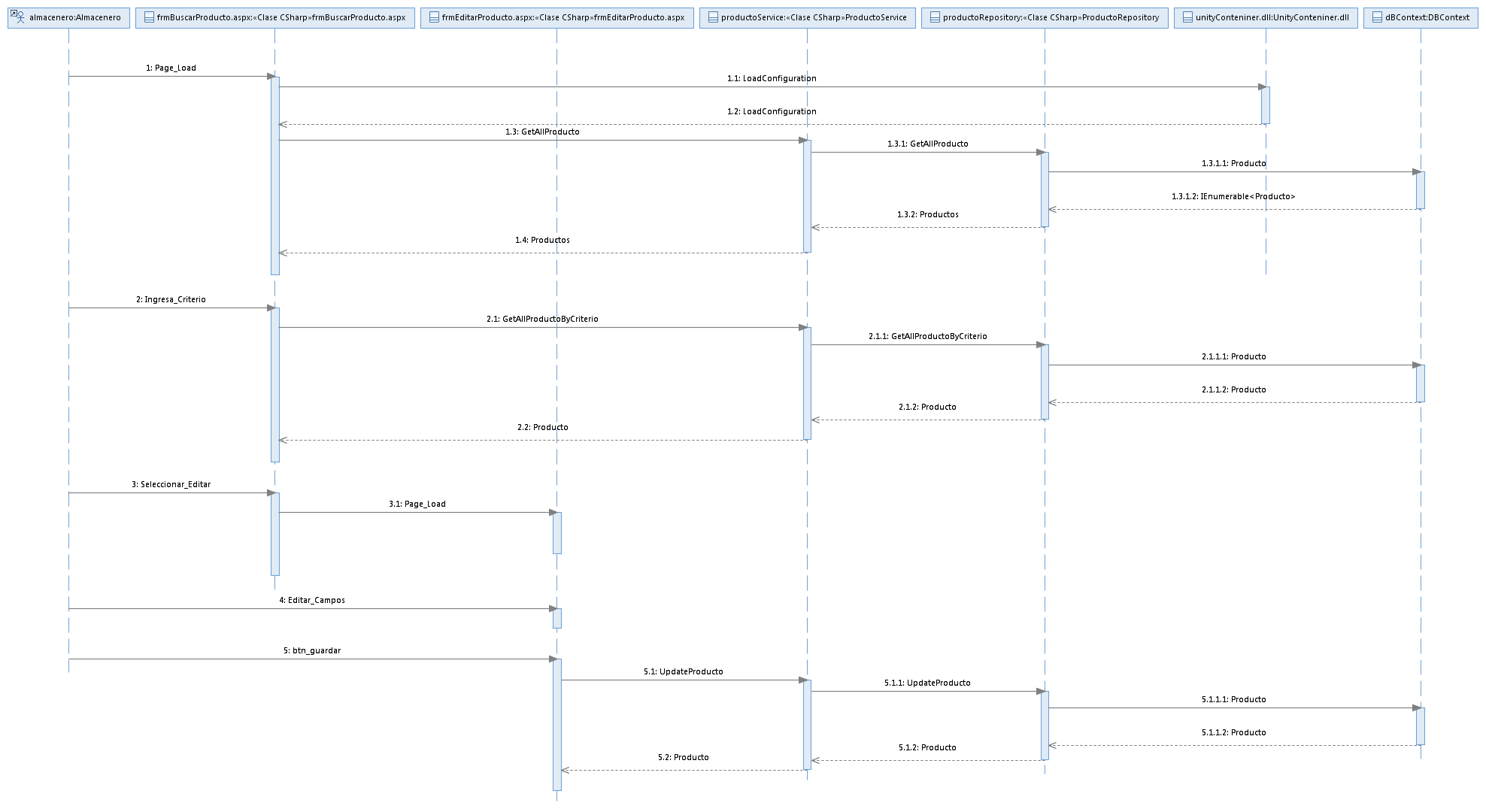
**Editar Producto**

**Opinión de los participantes**

**Escenario básico**



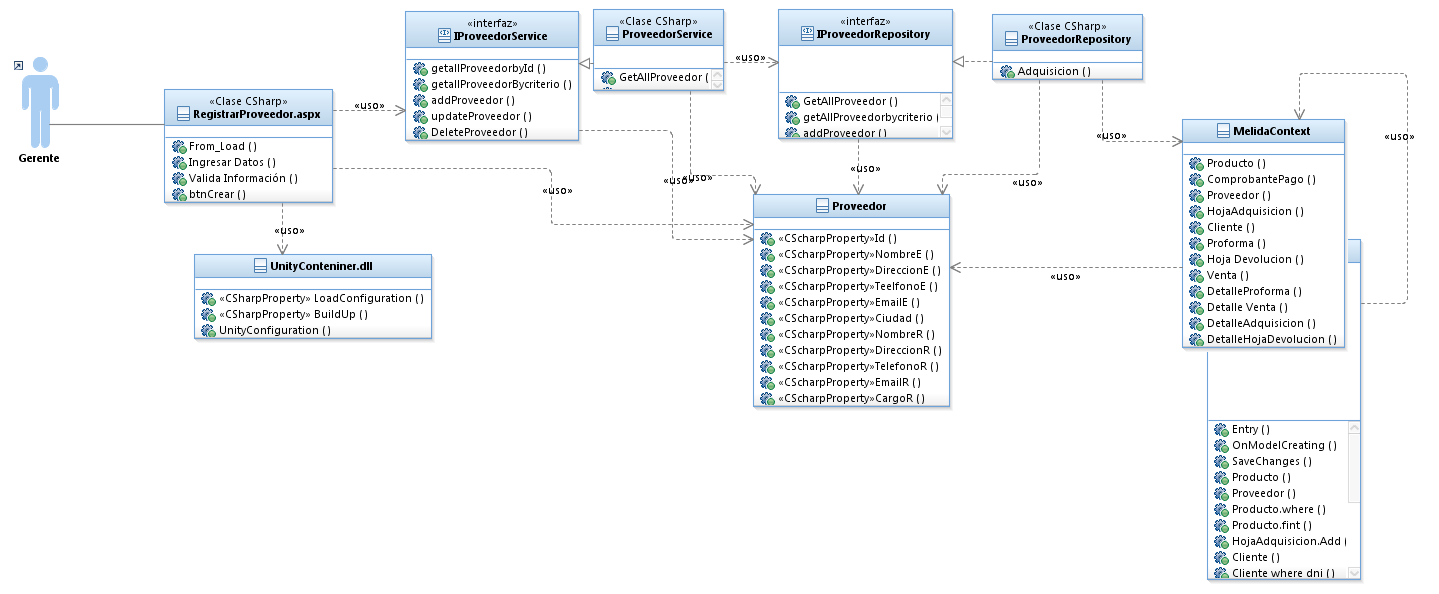
**Escenarios adicionales**



**Registrar Proveedor**

**Opinión de los participantes**

**Escenario básico**



**Escenarios adicionales**

