

Análisis

1.- Introducción

Las actividades de análisis y diseño ayudan a **transformar los requerimientos iniciales en un diseño implementable en software**. Durante el análisis, partir de los casos de uso, se construye el **modelo de análisis**. El modelo de análisis contiene **clases y las colaboraciones entre ellas** que exhiben el **comportamiento dinámico del sistema**. El nivel de abstracción del modelo de análisis es todavía demasiado elevado para permitir su implementación directa.

Contenidos

- 1.- [Introducción](#)
- 2.- [Análisis de casos de uso](#)
- 3.- [Análisis de clases](#)

Las clases típicamente representan objetos, p.e. carro de la compra, pedido, producto,... en el **dominio de negocio**. El nivel de abstracción es tal que las clases que se puedan identificar durante el análisis podrían igualmente aplicarse a **otras arquitecturas** de aplicación diferentes a la de una aplicación Web. Los procesos y objetos más importantes del dominio del problema se identifican y categorizan durante el análisis.

El análisis se focaliza en los **requerimientos funcionales** del sistema, ignorando por el momento las restricciones de la arquitectura del sistema. El énfasis se pone en asegurar que los requerimientos funcionales, tal y como se expresan en los casos de uso y otros documentos, son contemplados. Idealmente, cada requerimiento y caso de uso se vincula con las clases y paquetes que lo realizan. Este vínculo es importante para asegurar el seguimiento entre requerimientos, casos de uso y las clases que los realizan.

Para facilitar la construcción del modelo de análisis, este se puede completar en dos pasos:

- Primero se analizan los casos de uso con más detalle. Esta fase se denomina "**Análisis de casos de uso**" y tiene como objetivo construir **diagramas de interacción simplificados** que detallan el curso de acontecimientos de cada caso de uso. Por lo tanto esta fase se centra en los aspectos más dinámicos del modelo de análisis.
- La segunda fase, denominada "**Análisis de clases**", se fundamenta en la primera y tiene como objetivo los **elementos estáticos del modelo**, los objetos del dominio. A partir de los objetos más evidentes, identificados en el "Análisis de casos de uso", se detalla el modelo de objetos del dominio. Este se enriquece con más elementos estáticos identificados en los casos de uso y otros documentos. Esto El modelo de objetos se formaliza utilizando **diagramas de clases**.

2.- Análisis de casos de uso




El análisis de casos de uso es una actividad que se realiza cuando los casos de uso están completos o próximos a completarse. En términos del proceso de desarrollo ICONIX esta actividad se denomina análisis de robustez. Los **objetivos** son:

- Identificar las clases que llevarán a cabo el flujo de eventos descrito en los casos de uso.
- Identificar las asociaciones entre las clases.

Los resultados principales de esta fase son **diagramas de interacción**, que contienen clases y relaciones entre ellas. Estos diagramas describen como un caso de uso dado es llevado a cabo en términos de clases.

El análisis de caso de uso se inicia con la elaboración de diagramas de interacción para los **casos de uso base**. Estos diagramas de interacción identifican una serie de **clases básicas**. Estas clases son las encargadas de desarrollar de forma conjunta los comportamientos detallados en los casos de uso. Durante el análisis de casos de uso no se entra en demasiado detalle sobre estas clases. El objetivo es mostrar como se relacionan entre ellas. Luego, en el análisis de clases, se retoman estas clases y se entra en más detalle.

Para facilitar la creación de los diagramas de interacción del análisis, las clases identificadas se enmarcan en tres **tipos**: interfaz, entidad y control.

1.  **Interfaz**: representan los elementos de la interfaz entre el actor y el sistema.
Ejemplos de instancias de estas clases, en el contexto de la Web, pueden ser páginas Web completas.
2.  **Entidad**: son aquellas cosas descritas en el caso de uso pero que perduran más allá del desarrollo de los eventos del caso de uso.
Por ejemplo: pedido, cliente, producto,...
3.  **Control**: representan procesos, actividades del sistema que se pueden nombrar y gestionan el funcionamiento de entidades e interfaces.
Por ejemplo: procesar factura, calcular impuestos,...

En los diagramas de interacción se muestran los actores del caso de uso, las clases (interfaces, entidades y controladores) y las relaciones entre ellos. Para facilitar las cosas se aplican estas **restricciones**:

- Los actores sólo pueden interactuar con interfaces.
- Las entidades únicamente se relacionan con controladores.
- Los controladores pueden interactuar con interfaces, entidades y otros controladores.

Una forma de empezar el análisis de casos de uso es examinando el texto del caso de uso en busca de nombres y verbos. Los nombres son candidatos a entidades. Los verbos son posibles controladores.

Por ejemplo, para el escenario principal del caso de uso "Crear nuevo cliente": *El usuario solicita darse de alta como cliente. Se muestra la pantalla de alta de cliente, el usuario introduce sus datos. Se comprueba que todo este correcto y entonces se registra el nuevo cliente en el sistema*, el análisis del caso de uso daría como resultado el diagrama de interacción (concretamente de robustez):



Diagrama de interacción del análisis del caso de uso "Crear nuevo cliente"

El diagrama anterior corresponde a un caso de uso muy sencillo. Normalmente, los diagramas de interacción del análisis múltiples interfaces, controladores y entidades. Por ejemplo, si el usuario pudiese darse de alta como un cliente avanzado para lo que se le solicitan más datos, el diagrama podría ser el siguiente:

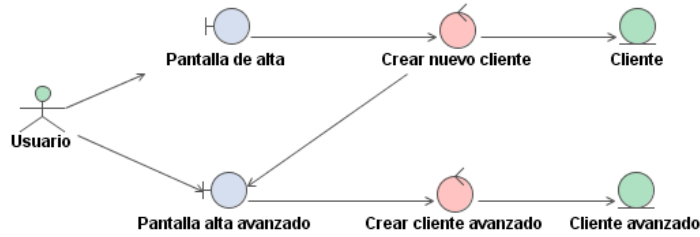


Diagrama de interacción del análisis del caso de uso "Crear nuevo cliente avanzado"

Los diagramas de interacción del análisis pueden tomar la forma de los presentados hasta el momento (llamados diagramas de robustez) o también la forma de los más comunes diagramas de secuencia, que también son diagramas de interacción:

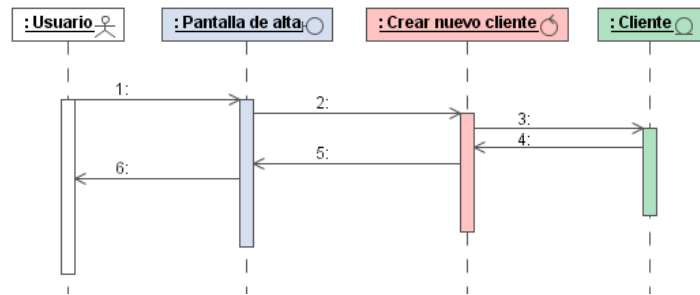


Diagrama de secuencia del análisis del caso de uso "Crear nuevo cliente"

o de igual forma para el caso más complejo:

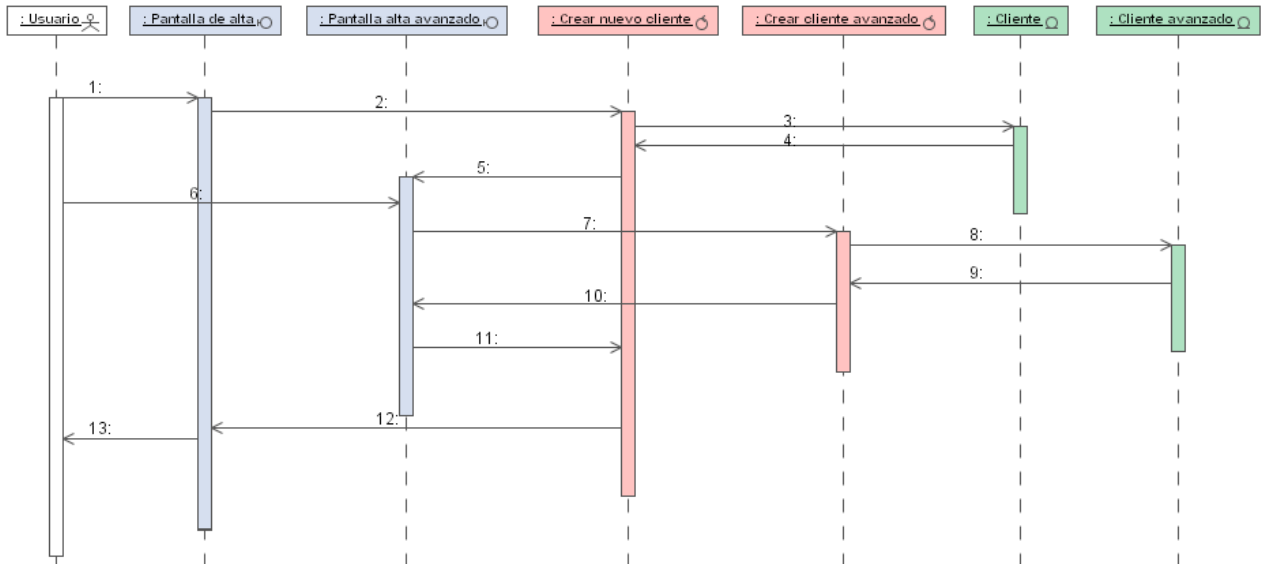


Diagrama de secuencia del análisis del caso de uso "Crear nuevo cliente avanzado"

Una vez completados los diagramas de análisis de los casos de uso, se han identificado y relacionado un conjunto de clases (categorizadas inicialmente como interfaces, controladores y entidades). Ahora es el momento de entrar en más detalle sobre los elementos que finalmente implementarán la aplicación, las clases. Para ello se pasa a la actividad llamada "Análisis de clases".

3.- Análisis de clases

El análisis de clases se inicia **partiendo de las identificadas en el análisis de casos de uso**. Para facilitar su manejo, las clases se organizan en paquetes. Los paquetes son herramientas UML para organizar las herramientas de representación que también aporta UML, ya sean estas últimas casos de uso (como ya se ha visto durante la definición de los casos de uso) o clases.

La estructura de paquetes de casos de uso desarrollada previamente se utiliza ahora como punto de partida para organizar las clases. A medida que se avanza en el análisis de clases esa estructura básica se irá enriqueciendo teniendo en cuenta los siguientes criterios para crear una estructura de paquetes "útil":

- **Comprensible:** cualquiera debe de ser capaz, con poco esfuerzo, de comprender la razón de la existencia de los diferentes paquetes, los elementos que se supone contendrán y sus responsabilidades.
- **Cohesiva:** todas las clases de un paquete forman un grupo de manera natural a algún nivel de abstracción y están por lo tanto relacionadas.
- **Poco acoplada:** generalmente, las clases tendrán más relaciones con las clases en del mismo paquete que con las clases de otros paquetes.
- **Jerarquía poco profunda:** las jerarquías profundas tienden a ser menos comprensibles. Por lo tanto, es mejor mantener en número de niveles de la jerarquía bajo.

Por lo tanto, partiendo de las clases ya identificadas, se examinan los casos de uso y los requerimientos funcionales **centrándose en las cosas que se describen** más que en las acciones. De esta manera se pueden identificar más clases y relaciones, además de nuevas relaciones entre las clases ya identificadas.

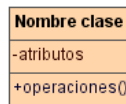
Como pauta para la identificación y enriquecimiento de las clases, se estudian nuevamente los casos de uso y requerimientos. Los **nombres**, como ya se ha comentado, son candidatos a constituir **clases** y los **verbos**, en este caso, **operaciones** de las clases. Por lo tanto, las entidades que ya se identificaron constituyen clases. Los **controladores** pueden constituir **clases** en si mismas o pasar a ser **operaciones** de clases existentes o nuevas. Finalmente, las interfaces se suelen dejar a parte durante el análisis de clases ya que se contemplarán durante el diseño de la interfaz.

Por ejemplo, para el fragmento de caso de uso "Pasar por caja":

El cliente le dice al sistema que está listo para pasar por caja. Se examina el contenido del carro de la compra y se genera una lista de todos los productos listos para la compra. El cliente confirma la compra y comunica al sistema que la procese,

se pueden identificar unos cuantos nombre que suenan importantes y que son buenos candidatos a clases: *cliente, carro de la compra, compra,...* Por otro lado las frases verbales y verbos: *pasar por caja, procesar,...* que indican acciones significativas del caso de uso y parecen buenas candidatas a operaciones de clases.

A partir de las clases y operaciones identificadas, más el conocimiento del dominio aportado por los casos de uso, expertos, el equipo de desarrollo, etc. se puede formalizar el análisis de clases en un diagrama UML. Se trata del diagrama de clases. Cada caja del diagrama representa una clase. Está dividida en tres partes: la superior contiene su nombre, la intermedia los atributos y la inferior las operaciones.



Partes de una clase de un diagrama de clases UML

Además, las clases se relacionan entre sí según se especifique en los casos de uso y en el conocimiento adquirido sobre el dominio. Existen múltiples tipos de relaciones entre clases, destacaremos:

- **Generalización:** también conocida como herencia. Permite reutilizar el comportamiento de las clases. La clase padre (destino de la flecha) es más general que la clase hija. Esta última tiene todo el comportamiento de la clase padre (lo reutiliza) y añade nuevo comportamiento más específico.
- **Asociación:** relaciona instancias de clases. La instancia de una de las clases necesita información de una instancia de la otra clase para llevar a cabo su comportamiento. Se representa como un segmento que conecta las dos clases. Se puede dar nombre a los dos extremos del segmento y también cuantificar la cardinalidad de la relación. Con cuantas instancias de la segunda clase se relaciona cada instancia de la primera, un valor del estilo 1, 1..*, 0..* que se coloca en el extremo del segmento más próximo a la segunda clase. Y viceversa, con cuantas instancias de la primera clase se relaciona una de la primera.
- **Agregación:** es un tipo especial de asociación en el cual se distingue un todo y una parte. Una instancia de una clase (todo) se relaciona con una colección de instancias de la otra clase (parte) que la primera agrega. Una agregación se representa con un segmento con un extremo en forma de diamante apuntando hacia la clase que representa el todo.

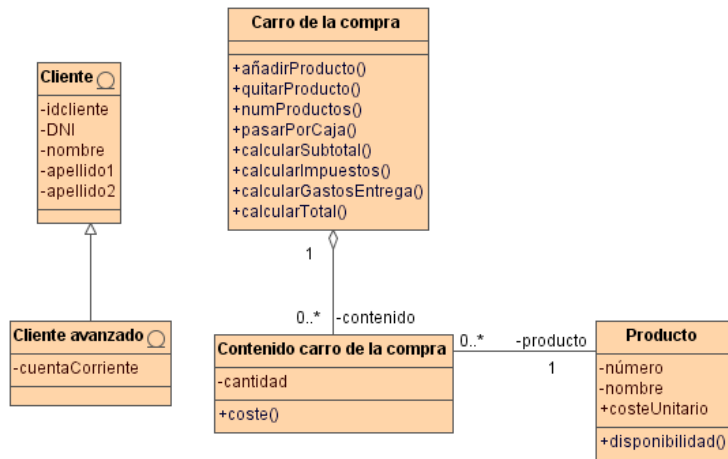


Diagrama de clases de análisis

Es importante destacar, llegados a este punto, que los **diferentes diagramas son vistas de un modelo común** de la aplicación. Por lo tanto no se desarrollan de forma aislada. Cambios en uno pueden enriquecer la visión dada por otros. En el caso de los diagramas utilizados en el análisis, una vez que se ha detallado el diagrama de clases, tanto las nuevas clases como las operaciones (como nombres de las flechas que van hacia los objetos que las contienen) pueden mostrarse en los diagramas de interacción previos. Así se consigue un modelo más rico del sistema.

Por ejemplo, ver el diagrama de interacción correspondiente al caso de uso "Pasar por caja" al que se ha añadido la operación identificada para el carro de la compra "calcularTotal" como el nombre de la flecha (invocación) que va desde el controlador a la entidad correspondientes.

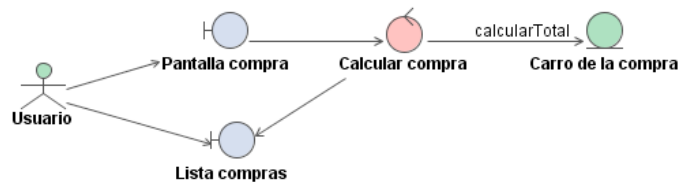


Diagrama de interacción del análisis enriquecido con operaciones del análisis de clases