Modelado con UML del caso de estudio

"La Mega Subasta Pública"

(Clase del 3 de mayo de 2011)

Introducción

Hemos aplicado un proceso software basado en UML a la especificación de requisitos de un servicio de subasta pública: La Mega Subasta Pública. El proceso está caracterizado por ser iterativo e incremental y basado en casos de uso. El proceso no se ha seguido enteramente, sino que nos hemos centrado en algunos aspectos de las fases de Modelado de Requisitos, Modelo de Análisis y Modelo del Diseño.

- Del Modelado de Requisitos se extrae el modelo de casos de uso y el modelo conceptual.
- En el Modelo de Análisis se obtiene un diseño preliminar del sistema a partir de los casos de uso. En esta fase obtenemos los diagramas de secuencia del sistema que nos ayudan a descubrir las operaciones básicas del mismo. Los contratos permiten especificar de forma detallada el efecto de estas operaciones y las colaboraciones muestran el comportamiento de las mismas, como interacción de objetos. En esta fase se obtiene también el Diagrama de Clases del Análisis.
- En el diseño se obtiene el Diagrama de Clases del Diseño, se define la arquitectura del sistema y se contempla el uso de patrones, entre otras cosas. En esta fase nos limitamos a comentar la ventaja que se podría obtener de usar ciertos patrones de diseño y mostramos un primer listado de la implementación del sistema (código fuente en Java) obtenido directamente de las colaboraciones.

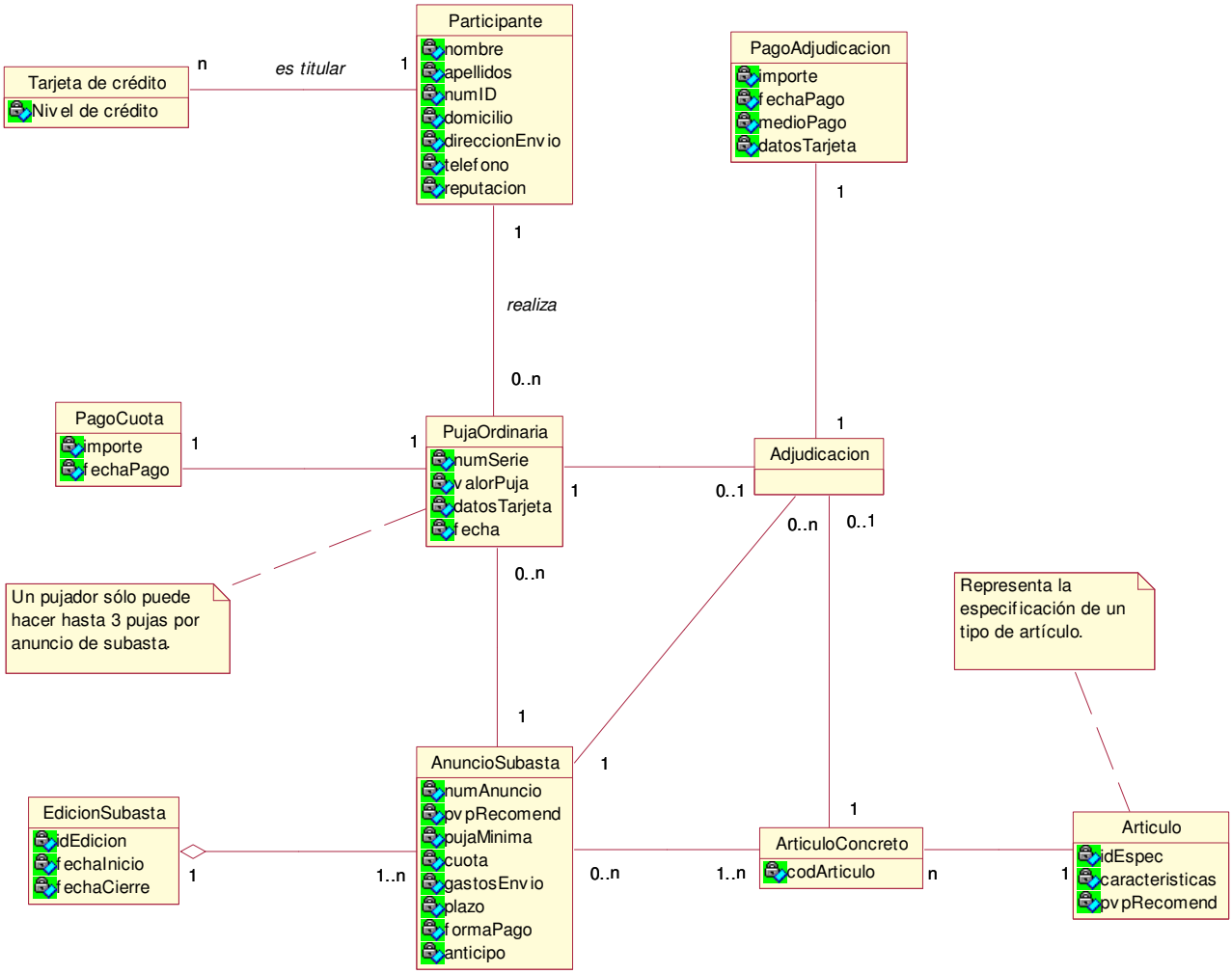
La estructura del documento es la siguiente:

Primero se muestra el diagrama Conceptual y el de Casos de Uso. A continuación, de cada CdU se deriva un apartado que contiene:

- Especificación textual del CdU.
- Diagrama de secuencia del sistema del CdU.
- Contratos junto con los diagramas de colaboración de los mismos. En todos ellos se ha intentado aplicar patrones GRASP.

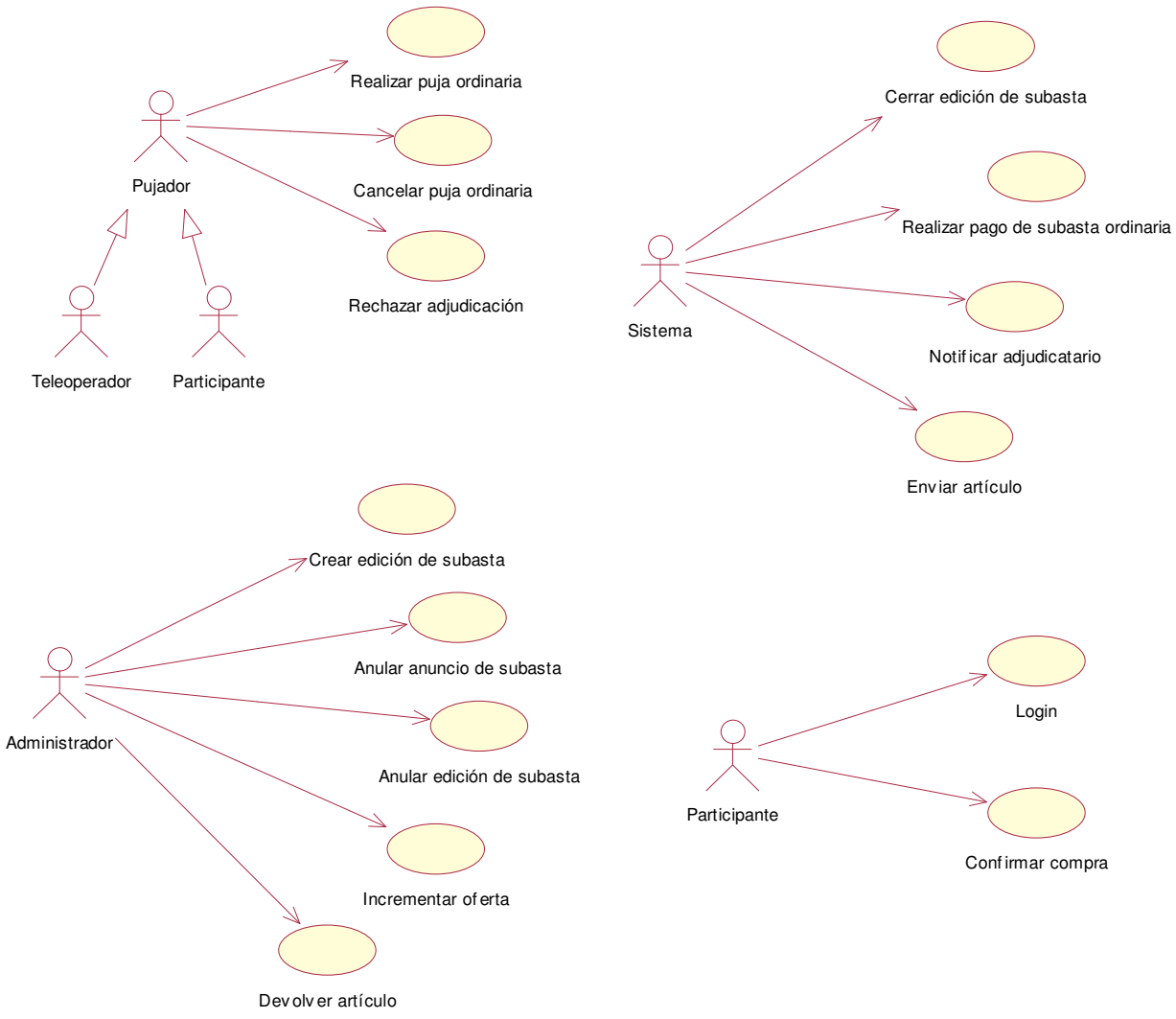
Por último se muestra el diagrama de Clases, comentario acerca del uso de patrones de diseño y el listado de código fuente.

Diagrama Conceptual



Nota del profesor. Observa que se podría haber introducido un concepto "Pago" que generalizara "PagoCuota" y "PagoAdjudicación", así como un concepto "Administrador" que representara el actor que aparece en el modelo de casos de uso y que entre otras cosas crea una edición de subastas o un anuncio de subasta. También se podría introducir un concepto "Banco" o "GestoraDeMediosDePago", que se encargaría de tramitar los pagos con tarjeta. Incluso se podría haber establecido una generalización entre "Tarjeta de crédito" y sus especializaciones: VISA, MasterCard, 4B y Tarjeta Punto Oro, con el objetivo de dotar de más información al modelo.

Diagrama de Casos de Uso



Nota del profesor. Podemos acotar la práctica centrando el trabajo en los siguientes casos de uso: Realizar puja ordinaria, Cancelar puja ordinaria, Cerrar edición de subasta, Realizar pago de subasta ordinaria y Crear edición de subasta.

Aspectos a tener en cuenta al leer los contratos y colaboraciones

De la fase de análisis hemos obtenido un diagrama de clases que varía ligeramente del diagrama conceptual. Este diagrama de clases ha surgido paralelamente a la escritura de las colaboraciones. Por ello, al leer los contratos y colaboraciones pueden aparecer clases que no estén en el diagrama conceptual, en concreto controladores, catálogos y nuevas clases relacionadas con el pago. Por ello, lo mejor es acudir al diagrama de clases para ver las diferencias.

Respecto al pago, hemos visto conveniente considerar que la clase Pago contiene un número variable de transferencias. Pago y Transferencia son clases abstractas. Las subclases de Pago definen un nuevo tipo de pago (por ejemplo PagoCuota) que incluye a su vez algún número de transferencias de cualquier tipo que sea conveniente implementar en el sistema (por ejemplo una transferencia que transfiera dinero de una tarjeta de cliente a una cuenta de la Mega Subasta). Esto facilita introducir nuevas formas de pago y de transferencias en el sistema. El pago lo realiza una clase llamada GestorPago que recibe un objeto Pago y lleva a cabo todas las transferencias especificadas en el mismo. Esto facilita diseñar la colaboración ya que GestorPagos tiene un único método que recibe un objeto Pago.

Use Case UC1: Crear edición de subasta

Stakeholders:

- Administrador: Desea que la lectura de datos sea correcta.
- Proveedor: Desea que el anuncio refleje fielmente la información proporcionada por él.
- Participante: Desea que la descripción del artículo se ajuste a la realidad, así como la fotografía. Que los datos mostrados en el anuncio sean correctos.

Actor: Administrador
Precondiciones: El Administrador está identificado y autenticado en el sistema.
Postcondiciones: Se creó una nueva edición de subasta con un conjunto de anuncios de subasta.
Ver contratos para más detalles.

Escenario Principal (o Flujo Básico):

1. El Administrador quiere crear una edición de subasta.
2. El Administrador introduce la fecha de inicio y cierre de la edición.
3. El Sistema registra la nueva edición, le asigna un número de edición y solicita la introducción de nuevos anuncios de subasta en la misma.
   Para cada subasta que el Administrador desea crear se realizan los pasos 4-12:
4. El Administrador crea una nueva subasta ordinaria.
5. El Sistema solicita al Administrador el artículo a subastar y presenta una lista de artículos disponibles.
6. El Administrador elige el artículo a subastar y el número de artículos.
7. El Administrador introduce (en cualquier orden) el valor de la puja mínima, la cuota de participación, los gastos de envío y el plazo de entrega.
8. El Sistema valida los datos.
9. El Sistema presenta al Administrador los datos introducidos con el IVA calculado al valor de puja mínima, a la cuota de participación y a los gastos de envío.
10. El Administrador guarda los cambios.
11. El Sistema registra la nueva subasta y asigna un número a la subasta.
12. El Sistema establece el estado de los artículos subastados a "En subasta".
13. El Administrador guarda los cambios.

Nota del profesor. Observa cómo en la línea 12 se dice que el sistema debe establecer el estado de los artículos subastados a "En subasta", pero en el modelo conceptual no hay ningún atributo que permita reflejar esto. Hay una inconsistencia entre el modelo de casos de uso y el modelo conceptual que sería preciso eliminar.

Extensiones (o Flujos Alternativos):

2-12a. El Administrador cancela la creación de la edición.
1. El Sistema cancela la edición junto con todas las subastas introducidas y no guarda los cambios.

6a. El Sistema detecta que el artículo elegido no está disponible en la cantidad solicitada.

1. El Sistema informa al Administrador de que la cantidad solicitada de artículos a subastar no está disponible y le permite elegir un artículo distinto o un número menor de artículos.

6-12a. El Administrador cancela la creación de subasta.
1. El Sistema cancela la subasta y no guarda los cambios.

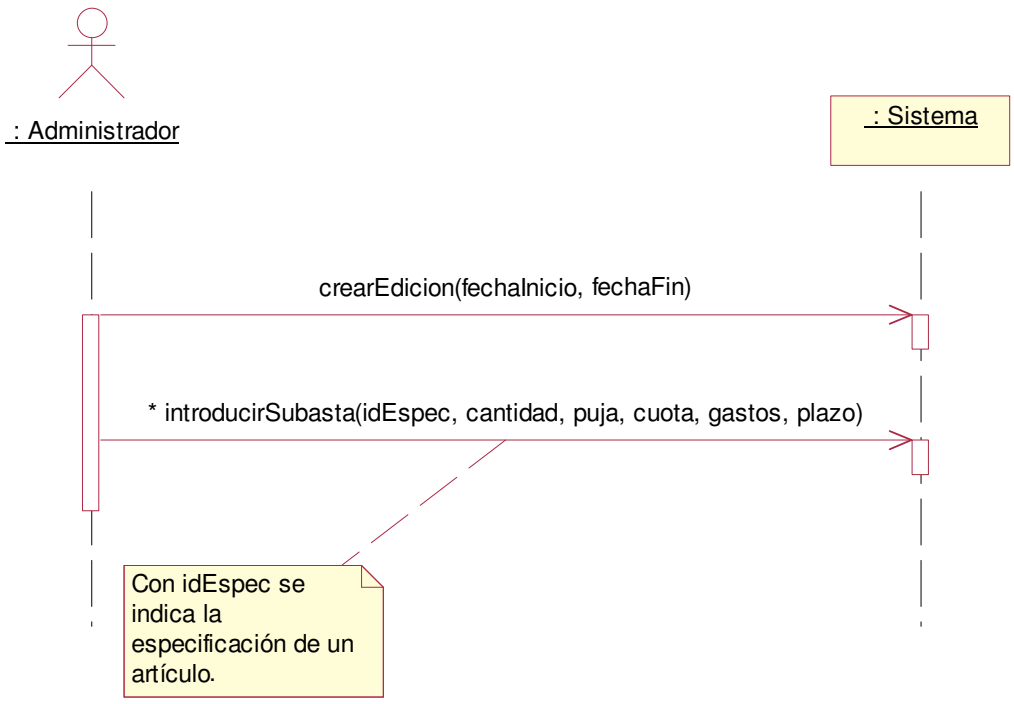
7a. El Administrador desea elegir una forma de pago.
1. El Administrador selecciona una forma de pago dentro de las posibles formas de pago.
2. El Sistema determina el valor del anticipo a partir de la forma de pago y pregunta al Administrador si desea modificar el valor del anticipo.
3. El Administrador no modifica el valor del anticipo.
   3a. El Administrador desea modificar el valor del anticipo.
      1. El Administrador introduce un valor de anticipo.

Requisitos Especiales:

- Un artículo no está disponible si está siendo subastado (su estado es "En subasta").

Nota del profesor. Ten en cuenta el comentario anterior sobre la consistencia entre el modelo conceptual y el modelo de casos de uso.

Diagrama de secuencia del sistema



Contratos y colaboraciones

Contrato: crearEdicion

Operación: crearEdicion(fechaInicio: Fecha, fechaFin: Fecha)
Referencias: Casos de Uso: Crear edición de subasta
Controlador: ControladorAnuncios
Precondiciones:

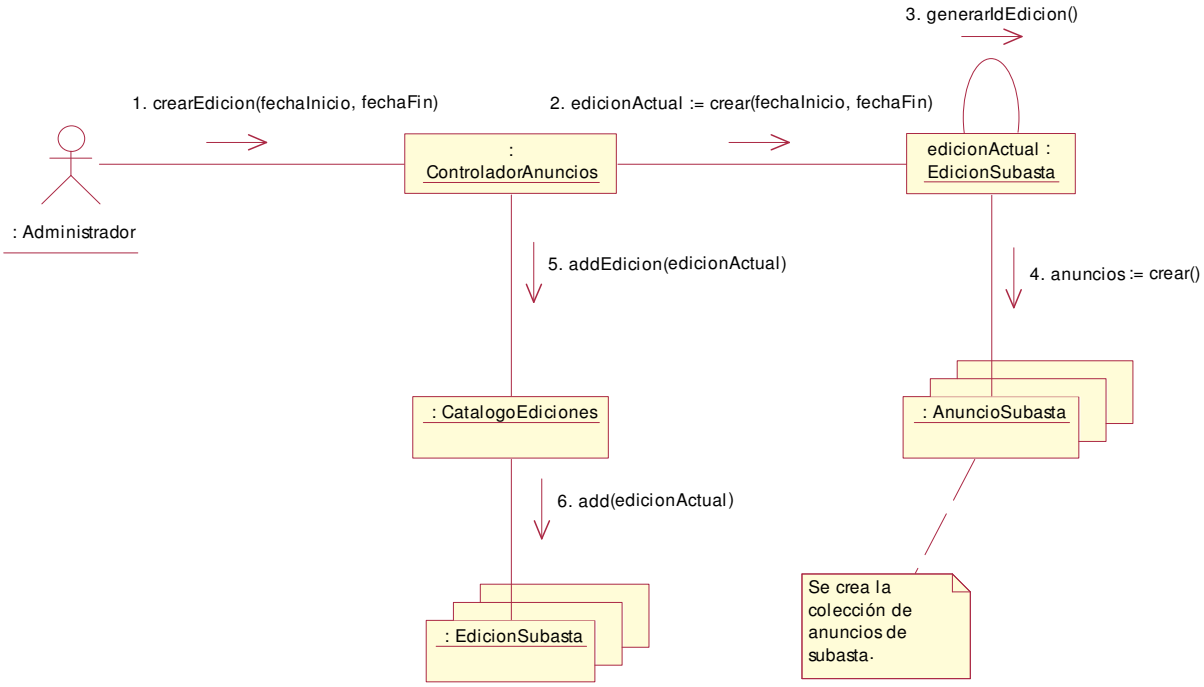
- ~~El Administrador está identificado y autenticado en el sistema.~~
- fechaInicio < fechaFin

Postcondiciones:

Nota del profesor. Observa el uso del pasado.

- Se creó una instancia edicionActual de EdicionSubasta.
- Se inicializó edicionActual.idEdicion
- edicionActual.fechaInicio = fechaInicio
- edicionActual.fechaCierre = fechaFin

Nota del profesor. Cuidado con la creación de colecciones, que se puede olvidar.

- Se creó una colección 'anuncios' para objetos de tipo AnuncioSubasta y se asoció con edicionActual.
- Se asoció edicionActual con el CatalogoEdiciones del ControladorAnuncios (se insertó en el catálogo).

Nota del profesor. Observa que se ha olvidado indicar un valor para edicionActual.estado, que es un atributo que aparece en el modelo de clases del análisis.