3 Análisis orientado a objetos

En la fase de Elaboración, siguiendo a UP, las iteraciones que se van a llevar a cabo deberán de servir para:

- Descubrir la mayoría de los requisitos.
- Estabilizar la arquitectura, reduciendo o eliminando los riesgos más importantes.
- Implementar el núcleo de la aplicación.

La Elaboración es una serie de iteraciones durante la cual el equipo lleva a cabo un estudio serio e implementa el núcleo central de la arquitectura. Esta fase puede estar constituida entre dos a cuatro iteraciones. En ellas no sólo se realiza una preparación de los módulos para que se implementen en la fase de construcción, sino que en esta fase es donde se implementan el núcleo del sistema. Durante esta etapa, los desarrolladores no están creando prototipos desechables, al contrario, el diseño y el código son porciones del sistema final con calidad de producción. Por tanto, en la Elaboración se construirá el núcleo central de la arquitectura, se resolverá la mayoría de los requisitos y se podrá estimarse la planificación y los recursos globales del proyecto.

Algunas buenas prácticas en la Elaboración serían:

- Llevar a cabo iteraciones de duración fija y dirigida por el alto riesgo.

- Comenzar a programar pronto.
- Diseñar, implementar y probar, de manera adaptable, las partes básicas y arriesgadas de la arquitectura.
- Probar desde el principio, a menudo y de manera realista.
- Adaptarse en base a la realimentación procedente de las pruebas, usuarios y desarrolladores.
- Escribir con detalle la mayoría de los casos de uso

UP pone los siguientes artefactos para conseguir los objetivos expuestos en la fase de elaboración:

Tabla 3-1. Artefactos en la fase de elaboración

Artefacto	Comentario
Modelo del Dominio	**Es una visualización de los conceptos del dominio; es similar al modelo de información estático de las entidades del dominio.**
Modelo de Diseño	**Es el conjunto de diagramas que describen el diseño lógico. Comprenden los diagramas de clases de diseño, diagramas de interacción, diagramas de paquete, etc.**
Documento de la Arquitectura Software	**Una ayuda de aprendizaje que resume las cuestiones claves de la arquitectura y de cómo se resuelven en el diseño. Es un resumen de las ideas destacadas del diseño y su motivación en el sistema.**
Modelo de Datos	**Incluye esquema de bases de datos y las estrategias de transformación entre representaciones de objetos y no objetuales.**
Modelo de Pruebas	**Una descripción de lo que se probará y de cómo se hará.**
Modelo de Implementación	**Se corresponde con la implementación real (el código, los ejecutables, los manuales, las bases de datos, etc.)**
Guiones de Casos de Uso, prototipos GUI	**Descripción de la interfaz de usuarios, caminos de navegabilidad, modelos de facilidad de uso, etc.**

3.1 El Modelo del Dominio

El Modelo del Dominio es el artefacto más importante del Análisis Orientado a Objetos, AOO. Un modelo del dominio muestra las clases conceptuales significativas en un dominio del problema. Las clases conceptuales son categorías reales del universo del problema, i.e. son las esencias del problema. Un modelo del dominio es fuente de inspiración para el diseño de los objetos. En UP, el Modelo del Dominio pertenece a la disciplina de Modelado del Negocio.

Las clases conceptuales del mundo real no son clases software. Hay tres tipos de clases: conceptuales, de diseño y de implementación. Las primeras corresponden con las esencias del universo del problema, la segunda categoría surge de la solución lógica dada en la etapa de diseño y la tercera se da en la implementación.

Hay diferencia entre las clases conceptuales, las clases de diseño y las de implementación, por tanto, en el modelo del dominio hay que evitar los métodos y los artefactos del software. Por ejemplo, las relacionadas con el GUI (*Graphical User Interface*), las bases de datos,...

El modelo del dominio es una representación visual de las clases conceptuales. El modelo del dominio podría considerarse como un diccionario visual de las abstracciones relevantes, del vocabulario del dominio y un resumen de la información del dominio.

En UML, el Modelo del Dominio se representa con un conjunto de diagramas de clases, mostrándose las clases conceptuales, sus asociaciones y sus atributos. En UML no existe el término de "Modelo de Dominio". UML es sólo unas normas para la representación gráfica de los conceptos de Ingeniería de la Programación; es UP el que crea el "Modelo del Dominio", los "Diagramas de Secuencia del Sistema" (DSS) y los "Contratos de Operación".

Las clases conceptuales podrían considerarse en términos de: a) su símbolo, b) su intensión y c) por su extensión:

- Símbolo: palabra o imágenes que representan una clase conceptual.
- Intensión: la definición de una clase conceptual.
- Extensión: el conjunto de ejemplos a los que se aplica la clase conceptual.

La diferencia esencial entre el análisis orientado a objetos y el análisis estructurado es:

"la división por clases conceptuales (objetos) en lugar de la división por funciones"

Los pasos para crear un modelo del dominio son:

1. Lista de las clases conceptuales candidatas.
2. Representación en un modelo del dominio.
3. Añadir las asociaciones necesarias para registrar las relaciones que hay que mantener en mente.
4. Añadir los atributos necesarios para satisfacer los requisitos de información.

El modelo del dominio se construirá:

- Utilizando los nombres existentes en el dominio.
- Excluyendo las características irrelevantes.
- No añadiendo cosas que no están.

Si se considera que un concepto es más que un número o un texto, entonces será una clase conceptual y no un atributo. En caso de duda, siempre será una clase conceptual.

3.1.1 Identificación de las clases conceptuales

La tarea central del AOO es identificar las clases conceptuales relacionadas con el escenario que se está diseñando. Se presentan dos técnicas:

1. Utilización de una lista de categorías de clases conceptuales.
2. Identificación de frases nominales.

Se comenzará la creación de un modelo del dominio haciendo una lista de clases conceptuales candidatas. Como regla general es mejor definir por exceso el número de clases que por defecto. La tabla 3.2 presenta una lista de categoría de clases conceptuales. Para una mejor comprensión, se ha puesto las categorías de tres problemas distintos: a) Clasificación de los espermatozoides, b) Aplicación para un puesto de venta en un comercio y c) Gestión de los billetes de avión:

Categoría de clase conceptual	Ejemplos
objetos tangibles o físicos	Espermatozoide, Imagen, Registro Avión
especificaciones, diseños o descripciones de las cosas	CaracterísticaDelEspermatozoide EspecificaciónDelProducto DescripcionDelVuelo
Lugares	Laboratorio Tienda Aeropuerto
Transacciones	Clasifica, Procesa Venta, Pago Reserva
roles de la gente	Biomédico Cajero, Cliente Piloto, Pasajero
contenedores de otras cosas	Directorios de imágenes, MatrizDeCaracteristicas Tienda Avión
cosas de un contenedor	Ficheros Artículo Pasajero
otros sistemas informáticos o electromecánicos externos al sistema	ServicioImpresion SistemaAutorizaciónPagoCredito ControlDeTraficoAereo
conceptos abstractos	Ansia Acrofobia
Organizaciones	LaboratorioDeAnálisis DepartamentoDeVentas CompañiaAerea
Hechos	ClasificarEspermatozoides Venta, Pago, Reunión Vuelo, Colisión, Aterrizaje
procesos (normalmente no se representan como conceptos, pero podría ocurrir)	ProcesadorImagenes, ClasificadorEspermatozoide VentaDeUnProducto ReservaUnAsiento
reglas y políticas	Reglas de clasificación espermatozoides PoliticaDeReintegro PoliticaDeCancelacion
catálogos	CatálogoDeEspermatozoides CatalogoDeProductos CatalogoDePiezas
registros de finanzas, trabajo, contratos cuestiones legales	Recibo,LibroMayor, ContratoEmpleo RegistroMantenimiento
instrumentos y servicios financieros	LineaDeCredito Stock
manuales, documentos, artículos de referencia, libros	LineaDeCambiosDePreciosDiarios ManualReparaciones

Tabla 3-2 Categorías de clases conceptuales

Otra técnica es el análisis lingüístico de los documentos de captura de los requisitos. Se trata de identificar los nombres y las frases nominales en las descripciones textuales y considerarlas como clases conceptuales o atributos candidatos. Sin embargo, la imprecisión del lenguaje natural es un punto débil de este enfoque. Se recomienda que se combine con las técnicas de lista de categorías de clases conceptuales.

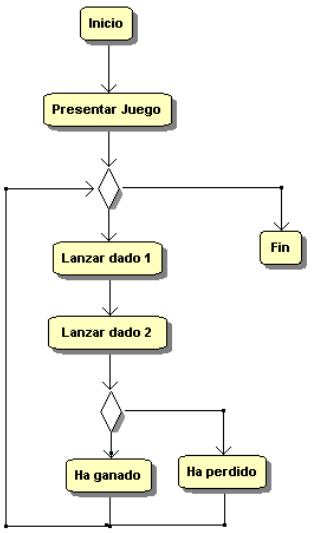
Ejemplo 3.1

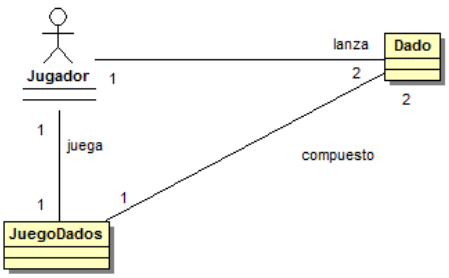
Determinar las clases conceptuales para la aplicación de *RespuestaFrecuencia*.

Categoría de clases conceptuales	Propuesta
Objetos tangibles o físicos	Filtro lineal, circuito electrónico
Especificaciones o descripciones de cosas	Respuesta en frecuencia, Función de transferencia
Lugares	Laboratorio
Transacciones	
Líneas de la transacción	
Roles de la gente	Ingeniero electrónico
Contenedores de otras cosas	Filtro lineal, Función de transferencia
Cosas en un contenedor	Función de transferencia, polinomios
Otros sistemas informáticos o electromecánicos externos al sistema	Capturador de circuitos electrónicos
Conceptos abstractos	
Organizaciones	
Hechos	
Procesos (normalmente no se representan como conceptos, pero podría ocurrir)	
Reglas y políticas	
Catálogos	
Registros de finanzas,	
Instrumentos	Diagrama de Bode
Manuales, documentos, artículos, libros	Apuntes de Regulación Automática

Ejemplo 3.2

Realizar el análisis de la aplicación de juego “dados” mediante técnicas estructuradas y mediante el artefacto modelo del dominio





3.1.2 Clases conceptuales de especificación

Las clases conceptuales de especificación están fuertemente relacionadas con las cosas que describen. Se añadirán cuando:

a) Se necesita la descripción de un atributo o servicio, independientemente de la existencia actual de algún ejemplo de esos atributos o servicios.

b) La eliminación de las instancias que describen da como resultado una pérdida de información que necesita mantenerse, debido a la asociación incorrecta de la información con la cosa eliminada.

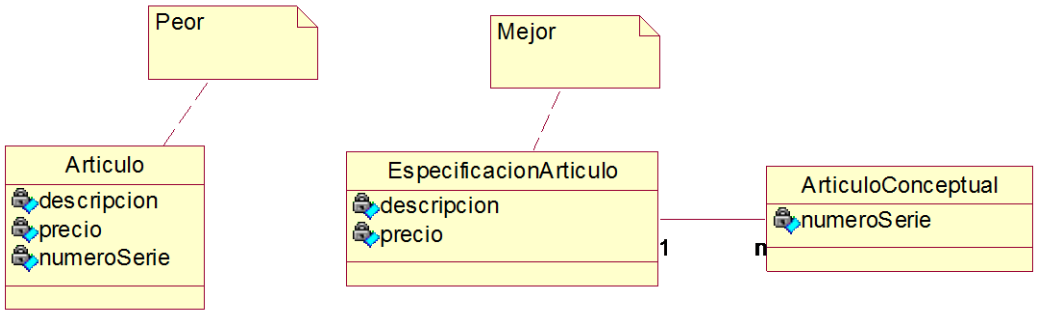
c) Reduce información redundante o duplicada.

Las clases conceptuales de especificación de X describen características de la clase conceptual X.

Ejemplo 3.3

Realizar la descripción de clase conceptual de un artículo de compra de una tienda.

Un artículo en concreto estará constituido por una descripción, un precio y un número de serie. Esta clase conceptual podría ser elaborada de esta manera. Sin embargo, en el caso de que se acabase las existencias del artículo desaparecería toda la información. Suceso éste que no se puede dar. En este caso se utilizará una clase conceptual de especificación.



3.1.3 Reducción en el salto de la representación

El modelo del dominio proporciona un diccionario visual de los conceptos del dominio. En la etapa del diseño, el modelo del dominio inspirará para nombrar algunas cosas de la solución lógica y de la implementación. Este modo de actuar acelerará la

comprensión del código existente. Los nombres de algunas clases del diseño coincidirán con las dadas en las clases conceptuales, luego sabiendo el modelo del dominio sugerirá, de forma natural, la comprensión del código.

Así, por ejemplo, al hablar de las clases conceptuales de "Función de Transferencia" y "Polinomios", al conocer el dominio del problema, cuando éstas pasen a ser clases de diseño y al encontrarlas en el código, sus relaciones con el mundo real deberán de coincidir con la lectura de sus implementaciones.

3.1.4 Modelo del Dominio: Añadir asociaciones

Una asociación es una relación entre clases conceptuales que indica alguna conexión significativa e interesante. Las asociaciones son abstracciones; no se trata de conexiones entre entidades software. Se trata de identificar las asociaciones y añadirlas al modelo del dominio.

Guía para las asociaciones:

- Centrarse en aquellas asociaciones para las que se necesita conservar el conocimiento de la relación durante algún tiempo (asociaciones "necesito-conocer").
- Es más importante identificar las clases conceptuales que identificar asociaciones.
- Demasiadas asociaciones tienden a confundir un modelo del dominio en lugar de aclararlo. Su descubrimiento puede llevar tiempo, con beneficio marginal.
- Evitar mostrar asociaciones redundantes.

Para la localización de las asociaciones comunes se emplea la técnica de rellenar la siguiente lista propuesta:

Tabla 3.3. Lista de asociaciones comunes

Categoría	Ejemplos
A es una parte física de B	Halo-Espermatozoide Cajón-Registro Ala-Avión
A es una parte lógica de B	Ficheros-DirectorioDeImagenes LineaDeVenta-Venta EtapaVuelo-RutaVuelo
A está contenida físicamente en B	Espermatozoides-muestra Registro-Tienda, Articulo-Estantería Pasajero-Avión
A está contenido lógicamente en B	Espermatozoides-imagen DescripcionDelArticulo-Catalogo Vuelo-PlanificacionVuelo
A es una descripción de B	Caractertisticas-Espermatozoide

	DescripcionDelArticulo-Articulo DescripcionDelVuelo-Vuelo
A es una línea de una transacción o informe de B	Histograma-Espermatozoides LineaDeVenta-Venta TrabajoMantenimiento-RegistroMantenimiento
A se conoce/registra/recoge/informa/captura en B	Venta-Registro Reserva-ListaPasajeros
A es miembro de B	Cajero-Tienda Piloto-CompañiaAerea
A es una subunidad organizativa de B	Departamento-Tienda Mantenimiento-CompañíaAerea
A utiliza o gestiona B	Cajero-Registro Piloto-Avión
A se comunica con B	Procesador-Clasificador Cliente-Cajero AgenteDeReserva-Pasajero
A está relacionado con una transacción de B	Cliente-Pago Pasajero-Billete
A es una transacción relacionada con otra transacción B	Pago-Venta Reserva-Cancelación
A está al lado de B	LineaDeVenta-LineaDeVenta Ciudad-Ciudad
A es prioridad de B	Registro-Tienda Avión-CompañíaAerea
A es un evento relacionado con B	Venta-Cliente Salida-Vuelo

Las asociaciones más prioritarias son:

- A es una parte lógica o física de B.
- A está contenida física o lógicamente en B.
- A se registra en B

En UML se representan como una línea entre clases con un nombre de asociación. La asociación es inherentemente bidireccional. Es posible el recorrido lógico de una a otra. Este recorrido es puramente abstracto. No se trata de una sentencia entre conexiones. Una flecha, de carácter opcional, indica la dirección de lectura. No indica la dirección de visibilidad o navegación.

Los roles son cada extremo de una asociación. Se suele indicar la multiplicidad (*: cero o muchos, 1...*: uno o más, 1 ...40: de uno a 40, 5: exactamente 5, 3,7,8: exactamente 3,7 y 8). La multiplicidad define cuántas instancias de una clase A puede asociarse con una instancia de una clase B.

El nombre de las asociaciones comienza por una mayúscula, puesto que representa un clasificador de enlaces entre instancias. Su formato es NombreTipo-FraseVerbal-NombreTipo (los nombres tipos corresponden a las clases conceptuales).

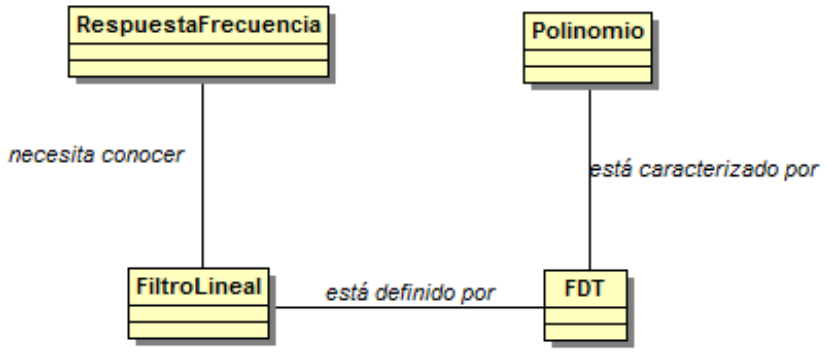
Hay que centrarse en las asociaciones necesito-conocer, pero se puede completar las asociaciones de sólo-comprensión para enriquecer el conocimiento básico del dominio.

Estas asociaciones, posteriormente, se implementarán muchas de ellas como caminos de navegación y visibilidad, pero su presencia en la vista conceptual de un modelo del dominio no requiere de su implementación.

No todas las relaciones descubiertas en la lista de asociaciones comunes deben ser colocadas. Puede haber un exceso y mostrar un ruido visual en el modelo del domino. Hay que seleccionar las relaciones necesito-conocer durante algún tiempo y eliminar asociaciones redundantes o derivadas.

Ejemplo 3.4

Dibujar el modelo del dominio de la aplicación *RespuestaFrecuencia*, indicando las asociaciones.



3.1.5 Modelo del dominio: añadir los atributos

Un atributo es un valor de un objeto. Los atributos se incluirán cuando se sugiera o implique una necesidad de registrar esta información.

Un error típico es modelar un concepto complejo del dominio como un atributo. Los atributos deben ser datos simples o tipos de datos. Los tipos de datos de los atributos incluyen: booleanos, fechas, números, horas, color, código postal, etc.

Una regla básica es que un atributo no es significativo de crear instancias diferentes del él. Por ejemplo, el concepto *Estudiante* tiene entidad de ser clase conceptual, pero no el *número de matrícula*. Existen muchos estudiantes con diferentes atributos, pero el número de matrícula carece de sentido de que tenga más de una

instancia por estudiante. Por tanto, *Estudiante* si es una clase y el *número de matrícula* es un atributo.

En caso de duda sobre un concepto, éste se definirá como una clase conceptual en vez de un atributo.

Guía para presentar un concepto como una clase y no como un atributo:

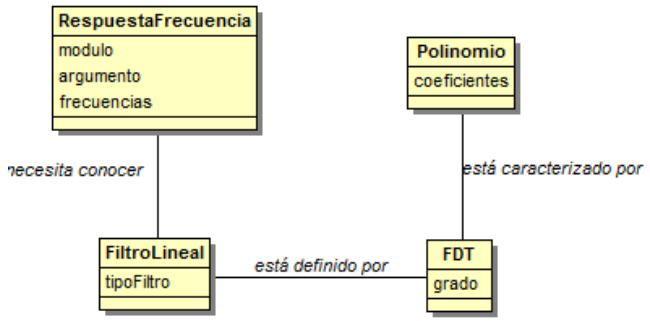
1. Si el concepto está constituido por partes.
2. Hay operaciones asociadas con él.
3. Es una cantidad con unidades.
4. Tiene otros atributos.
5. Es una abstracción de uno o más tipos (superclase – subclase, relacionado con conceptos de herencia).

Los tipos de datos pueden ser representados en el modelo del dominio, pero puede producir ruido visual.

No se debería utilizar atributos para relacionar las clases conceptuales en el modelo del dominio. Hay que relacionarlas con una asociación, no con un atributo. Se trata de posponer y no deslizarse hacia el diseño.

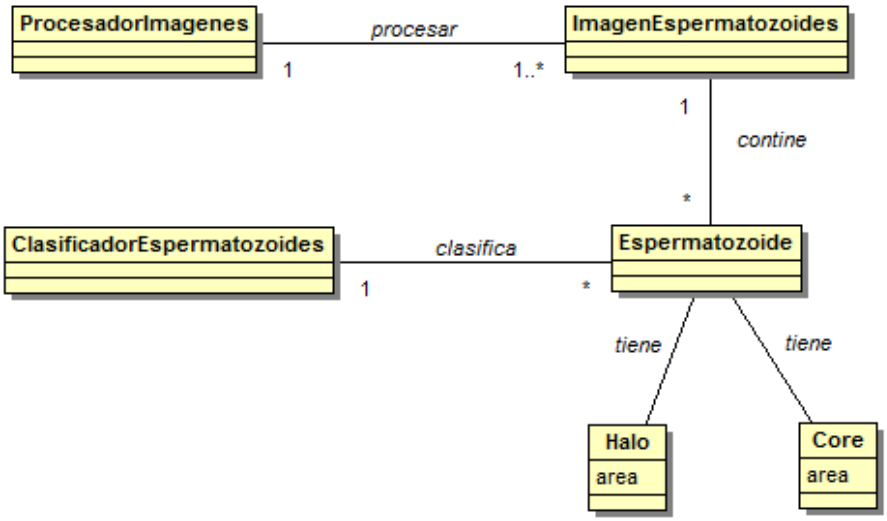
Ejemplo 3.5

Describir el modelo del dominio para el caso de uso de *RespuestaFrecuencia*



Ejemplo 3.6

Describir el modelo del dominio para el caso de uso de *ClasificaciónEspermatozoides*



3.2 Diagrama de secuencias del sistema

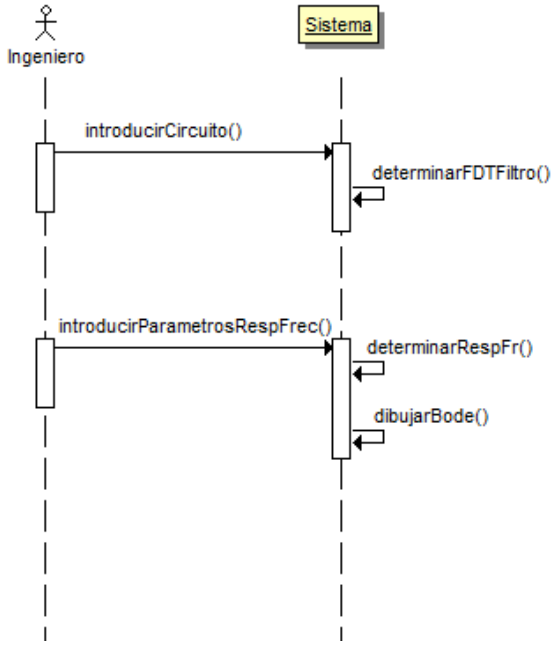
El comportamiento de un sistema es una descripción de lo qué hace el sistema, sin explicar cómo lo hace. Una parte de esta descripción es un diagrama de secuencias del sistema, DSS. Otras partes corresponden con los casos de uso y con los contratos de operación.

Un diagrama de secuencias del sistema o DSS es un dibujo que muestra, para un escenario específico de un caso de uso, los eventos que generan los actores, indicando el orden y la respuesta del sistema a modo de caja negra cerrada. Los DSS forman parte de la disciplina de Modelo de Casos de Uso. La mayoría de los DSS se crean durante la fase de Elaboración, cuando es útil identificar con detalle los eventos del sistema. No es necesario crear DSS para todos los escenarios de los casos de uso, sólo para aquellos seleccionados en la iteración con la que se trabaja.

Los eventos del sistema y sus operaciones asociadas deberían expresarse al nivel de intenciones, en lugar de en términos de interacción con los elementos del interfaz gráfico. También se mejora la claridad, si se comienza el nombre de un evento del sistema con un verbo.

Ejemplo 3.7

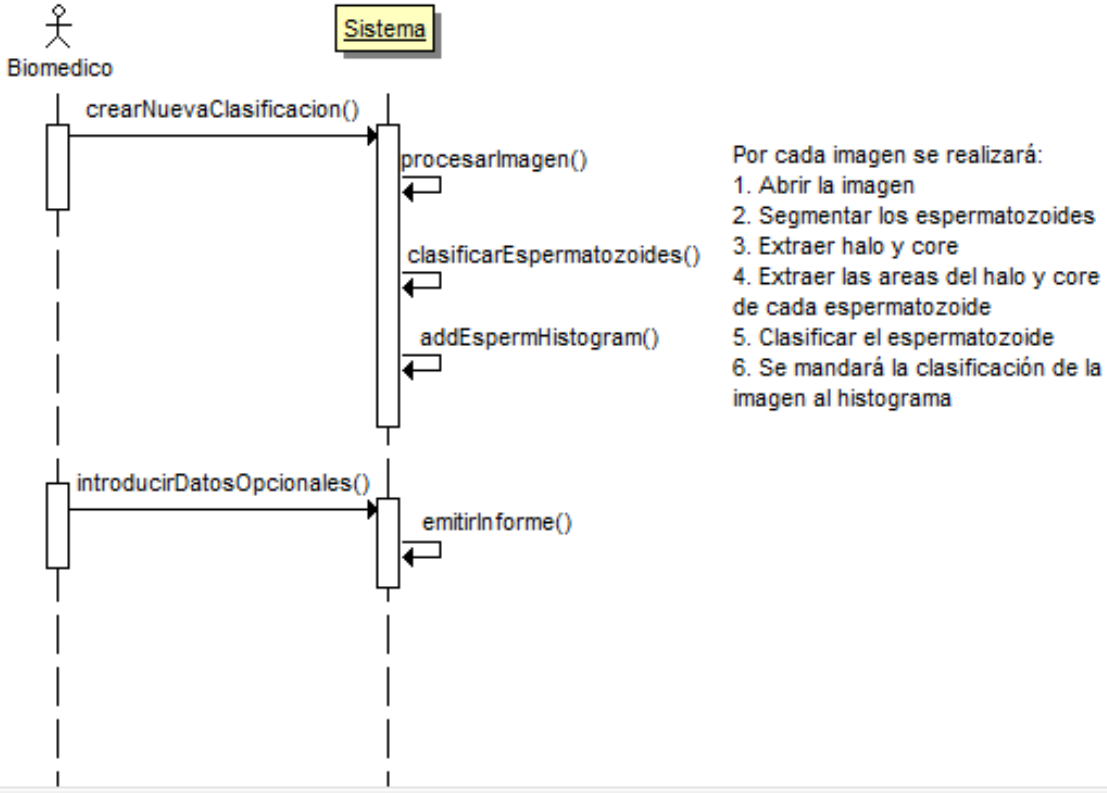
Realizar el DSS para el caso de uso de *RespuestaFrecuencia*.



A los DSS se les puede añadir una leyenda adicional sobre los eventos del sistema.

Ejemplo 3.8

Realizar el DSS del caso de uso de *ClasificarEspermatozoides*.



3.3 Los contratos de operación

Los contratos de operación describen los resultados de las ejecuciones de las operaciones del sistema en función de los cambios de estado de los objetos del dominio. Los contratos son complementarios a los casos de uso.

Se pueden definir contratos para las operaciones del sistema. Éstas se descubren identificando los eventos del sistema.

Las secciones del contrato son:

Operación: Nombre de la operación y parámetros.

Referencias cruzadas: (opcional) Casos de usos en los que pueden tener lugar esta operación.

Precondiciones: Suposiciones relevantes sobre el estado del sistema o de los objetos del modelo del dominio, antes de la ejecución de la operación. Se supondrá que todo es verdad.

Poscondiciones: El estado de los objetos del modelo del dominio después de que se complete la operación.

Las poscondiciones no son acciones que se ejecutarán durante la operación; más bien son cómo los objetos del dominio han evolucionado cuando la operación ha terminado. Las postcondiciones se dividen en tres categorías:

- Creación y eliminación de objetos.
- Modificación de atributos.
- Formación y rotura de asociaciones.

La eliminación de instancias es más rara, nadie se preocupa de forzarla explícitamente. Ésta es una perspectiva conceptual, no de implementación.

Los contratos de operación se expresan dentro del Modelo del Dominio y describen los cambios de estado que requiere una operación del sistema, sin tener que describir cómo se van a llevar a cabo.

Es normal, durante la creación de los contratos, descubrir nuevas clases conceptuales. Hágase una realimentación.

Los casos de uso son los principales artefactos de los requisitos del proyecto. Los contratos de operación se emplearán donde la complejidad sea alta y añade valor la precisión detallada.

Para hacer contratos:

1. Identifique las operaciones del sistema a partir de los Diagramas de Secuencias del Sistema, DSS.

2. Construir un contrato para las operaciones complejas del sistema y quizás sutiles en sus resultados.

3. Para describir las postcondiciones utilice las siguientes categorías:

    a) Creación y eliminación de instancias

    b) Modificación de atributos

    c) Formación y rotura de asociaciones.

Las sentencias de las postcondiciones se emplearán en forma verbal pasiva y en pasado, ya que son resultados del pasado.

Los Contratos de Operación no se justifican en la fase de Inicio, son demasiado detallados, se hacen en la Elaboración y sólo para las operaciones más complejas y sutiles.

Ejemplo 3.6

Realizar los contratos de operación de *RespuestaFrecuencia*.

Contrato CO1: `introducirCircuito()`

Operación:	`introducirCircuito` ();
Referencias Cruzadas:	Caso de uso: Respuesta en frecuencia (inclusión de Capturar circuito lineal)
Precondiciones:	Se le pasa el esquema del circuito lineal
Poscondiciones:	Se creó una instancia de Filtro lineal pFiltro. pFiltro se construyo con la FDT correspondiente .

Contrato CO2: `introducirParamResFr`

Operación:	`introducirParamResFr` ();
Referencias Cruzadas:	Caso de uso: Respuesta en frecuencia
Precondiciones:	Existe pFiltro.
Postcondiciones:	Se creo una instancia de Respuesta en Frecuencia. Se cálculo el módulo y argumento para el rango de frecuencias dadas.

3.4 Problemas

1. Objetivos, disciplinas y artefactos al servicio de la fase de elaboración en UP.

2. Diferencia entre el análisis estructurado y el análisis orientado a objetos.

3. Diferencia entre clases conceptuales, clases de diseño y clases de implementación.

4. Guía para la localización de las clases conceptuales.

5. Asociaciones más importantes a retener en el modelo del dominio.

6. Cómo diferenciar que un concepto es una clase conceptual o un atributo.

7. Que es un DSS.

8. Cuando emplear un contrato de operación y sus partes.

Problema 3.1

Para el simulador de dinámica temporal de un sistema LTI (ver problema 1 del capítulo 2), se solicita:

1. Modelo del dominio

2. Diagrama de Secuencias del Sistema (DSS)

3. Contrato de Operaciones para los eventos más importantes.

Problema 3.2

Determinar para el juego de las tres en raya:

1. Modelo del dominio

2. Diagrama de Secuencias del Sistema (DSS)

3. Contrato de Operaciones para los eventos más importantes.

Problema 3.3

Emplear los artefactos necesarios para el AOO del juego del frontón (ver problema 2.4)

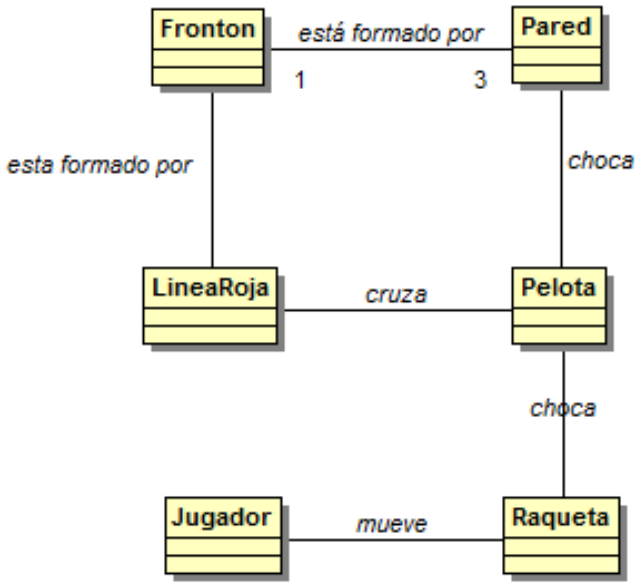
Empleando la información de la lista jerárquica a dos niveles, el glosario, la lista de evento-actor-objetivo y el caso de uso, se rellena la tabla de clases de conceptos:

Categoría de clases conceptuales	Propuesta
Objetos tangibles o físicos	Raqueta, pelota, frontón, pared, línea roja
Especificaciones o descripciones de cosas	
Lugares	Frontón
Transacciones	
Líneas de la transacción	
Roles de la gente	Jugador
Contenedores de otras cosas	Frontón
Cosas en un contenedor	Pared, línea roja
Otros sistemas informáticos o electromecánicos externos al sistema	
Conceptos abstractos	
Organizaciones	
Hechos	
Procesos (normalmente no se representan como conceptos, pero podría ocurrir)	
Reglas y políticas	
Catálogos	
Registros de finanzas,	
Instrumentos	Raqueta
Manuales, documentos, artículos, libros	

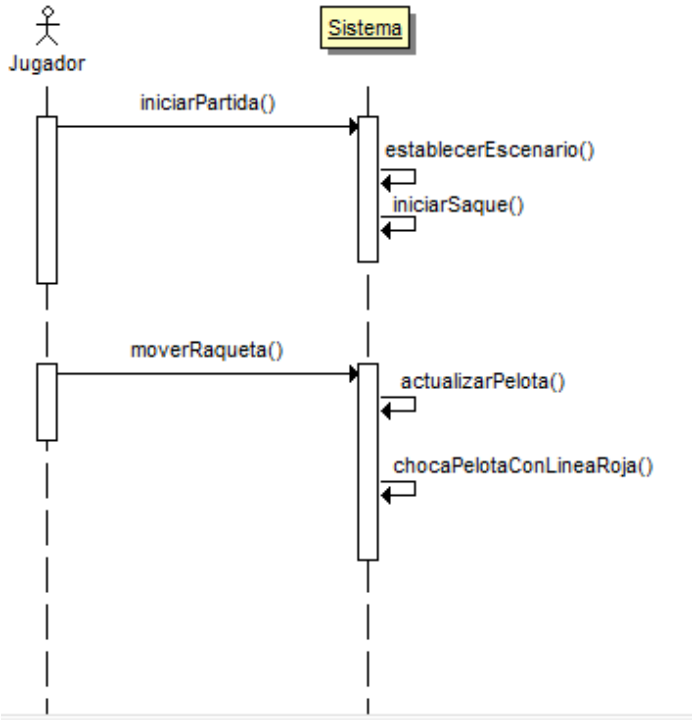
Seguidamente se buscarán asociaciones entre las clases conceptuales:

Categoría	Ejemplos
A es una parte física de B	Pared-frontón, línea roja-frontón
A es una parte lógica de B	
A está contenida físicamente en B	Pared-frontón, línea roja-frontón
A está contenido lógicamente en B	
A es una descripción de B	
A es una línea de una transacción o informe de B	
A se conoce/registra/recoge/informa/captura en B	Raqueta-pelota, pelota-pared, pelota-línea roja
A es miembro de B	
A es una subunidad organizativa de B	
A utiliza o gestiona B	Jugador-raqueta
A se comunica con B	
A está relacionado con una transacción de B	
A es una transacción relacionada con otra transacción B	
A está al lado de B	
A es prioridad de B	
A es un evento relacionado con B	

A continuación se refleja las asociaciones más importantes en el Modelo del Dominio:



Otro artefacto a realizar es el Diagrama de Secuencias del Sistema.



Problema 3.4

Para el juego del '*Pang*' (ver problema 2.5), se solicita:

1. Modelo del dominio

2. Diagrama de Secuencias del Sistema (DSS)

3. Contrato de Operaciones para los eventos más importantes.