



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



Departamento de Sistemas Informáticos y Computación  
Universitat Politècnica de València

## Ingeniería de Sistemas de Información Avanzada: del modelo de requisitos al producto software

Trabajo Fin de Máster  
Máster Universitario en Ingeniería y Tecnología de Sistemas Software

**Autor:** Catalán Tendero, Pablo  
**Tutor:** Pastor López, Oscar  
2016/2017



# Resumen

---

El diseño y desarrollo de un Sistema de Información requiere la transformación de los requisitos de los clientes del sistema en un producto software que represente dichos requisitos correctamente. Este proyecto analizará qué modelos y qué transformaciones entre modelos hay que usar para poder realizar dicha transformación de requisitos a código de forma correcta, asegurando la trazabilidad entre la aplicación y los objetivos organizacionales a los que dicha aplicación da respuesta, demostrando la viabilidad de disponer de procesos de producción de software que tengan un grado máximo de automatización.

**Palabras clave:** Sistemas de Información, Desarrollo Dirigido por Modelos, Ingeniería de Requisitos, Modelado Conceptual

# Abstract

---

The design and development of an information system requires the transformation of the requirements of the system's customers into a software product that correctly represents those requirements. This project analyzes which models and which transformations between the models are used to carry out the transformation of the requirements a correct form code, ensuring the traceability between the application and the organizational objectives to which the application of the response, demonstrating the feasibility to provide software production processes that have a maximum degree of automation.

**Keywords:** Information Systems, Model Driven Development, Requirements Engineering, Conceptual Modeling



# Tabla de contenidos

---

## Índice

Índice.....	5
Tabla de figuras .....	8
1. Introducción .....	11
1.1. Motivación de este trabajo .....	11
1.2. Alcance del documento .....	12
2. Fundamentos teóricos y tecnológicos.....	13
2.1. Model Driven Architecture .....	13
2.1.1. Modelo Independiente de Computación.....	13
2.1.2. Modelo Independiente de Plataforma .....	13
2.1.3. Transformación de CIM a PIM .....	14
2.1.4. Modelo Dependiente de Plataforma .....	14
2.1.5. Transformación de PIM a PSM .....	14
2.2. OO-Method .....	15
2.2.1. Modelo de Objeto .....	15
2.2.2. Modelo Dinámico .....	15
2.2.3. Modelo Funcional .....	16
2.2.4. Modelo de Presentación.....	16
2.3. Integranova Modeler.....	17
2.3.1. ¿Cómo funciona?.....	17
2.3.2. PIM - Integranova Modeler .....	17
2.3.3. PSM – Compilador de Modelos .....	18
3. Método para generar modelos ejecutables a través del modelo de requisitos .....	19
3.1. Modelo Independiente de Computación.....	19
3.2. Modelo Independiente de Plataforma .....	20
3.3. Modelo Dependiente de Plataforma .....	20
4. Caso de Estudio 1 .....	21
4.1. Motivación de este caso .....	21
4.2. Descripción contextual del caso .....	22
4.2.1. Descripción general .....	22
4.2.2. Funciones del sistema.....	22
4.2.3. Características de los usuarios.....	23
4.2.4. Requisitos Específicos.....	23

4.2.5.	Formularios de negocio .....	25
4.3.	Análisis comunicativo de negocio.....	28
4.3.1.	Introducción .....	28
4.3.2.	Diagramas de comunicación.....	28
4.3.3.	Descripción de eventos.....	33
4.4.	Modelo conceptual aplicado a la metodología-OO .....	39
4.4.1.	Derivación del modelado de objetos .....	39
4.4.2.	Diagrama de clases .....	43
4.5.	Construcción del modelo conceptual en Integranova .....	44
4.6.	¿Cómo abordar la posible automatización completa del proceso de generación? ....	52
4.6.1.	Relaciones entre clases.....	52
4.6.2.	Diagrama resultante .....	54
4.6.3.	Transacciones .....	55
5.	Caso de Estudio 2 .....	56
5.1.	Motivación de este caso .....	56
5.2.	Descripción contextual del caso .....	57
5.2.1.	Funciones del sistema.....	57
5.2.2.	Características de los usuarios.....	58
5.2.3.	Requisitos Específicos.....	58
5.2.4.	Formularios de negocio .....	60
5.3.	Análisis comunicativo de negocio.....	63
5.3.1.	Introducción .....	63
5.3.2.	Diagramas de comunicación.....	64
5.3.3.	Descripción de eventos.....	68
5.4.	Modelo conceptual aplicado a la metodología-OO .....	75
5.4.1.	Derivación del modelado de objetos .....	75
5.4.2.	Diagrama de clases .....	81
5.5.	Construcción del modelo conceptual en Integranova .....	82
5.6.	¿Cómo abordar la posible automatización completa del proceso de generación? ....	94
5.6.1.	Relaciones entre clases.....	94
5.6.2.	Diagrama resultante .....	97
5.6.3.	Atributos derivados .....	98
6.	Análisis de automatización del proceso .....	100
6.1.	Proceso de generación semiautomática.....	100
6.1.1.	Descripción contextual del caso .....	100
6.1.2.	Análisis comunicativo de negocio .....	100

6.1.3.	Modelo conceptual aplicado a la metodología-OO .....	101
6.1.4.	Modelado en Integranova .....	101
6.2.	Proceso de generación automática .....	102
6.2.1.	Modelado en Integranova .....	102
6.2.2.	Modelo XML .....	102
6.2.3.	Cliente Integranova Star .....	102
6.2.4.	Proceso de la generación de código .....	102
7.	Conclusiones .....	103
7.1.	Ventajas.....	103
7.1.1.	Precisión de la solución .....	103
7.1.2.	Reducción de los costes de trabajo .....	103
7.1.3.	Cambios en el sistema .....	103
7.2.	Desventajas .....	104
7.2.1.	Curva de aprendizaje .....	104
7.2.2.	Modelo de negocio.....	104
7.3.	Trabajo futuro .....	105
	Paso del modelo conceptual a Integranova.....	105
7.3.1.	.....	105
7.4.	Crítica personal.....	106
7.4.1.	Aspectos positivos .....	106
7.4.2.	Aspectos negativos .....	106
8.	Referencias.....	107

## Tabla de figuras

Ilustración 1 OO-Method .....	15
Ilustración 2 Crear nave nodriza.....	28
Ilustración 3 Crear aeronave .....	29
Ilustración 4 Asignar pasajero .....	30
Ilustración 5 Bajar pasajero.....	31
Ilustración 6 Crear pasajero .....	31
Ilustración 7 Revisar aeronave .....	32
Ilustración 8 OO-Crear Nave nodriza.....	39
Ilustración 9 OO-Crear aeronave.....	40
Ilustración 10 OO-Gestionar pasajeros.....	41
Ilustración 11 OO-Revisar aeronave .....	42
Ilustración 12 OO-Diagrama de clases C1.....	43
Ilustración 13 I-Atributos Nave nodriza.....	44
Ilustración 14 I-Servicios Nave Nodriza .....	45
Ilustración 15 I-Atributos Aeronave .....	46
Ilustración 16 I-Servicios Aeronave .....	47
Ilustración 17 I-Atributos Pasajero .....	48
Ilustración 18 I-Servicios Pasajero .....	49
Ilustración 19 I-Atributos Revisión .....	50
Ilustración 20 I-Servicios Revisión .....	51
Ilustración 21 I-Relación Nave Nodriza Destino .....	52
Ilustración 22 I-Relación Nave Nodriza Origen .....	52
Ilustración 23 I-Relación Aeronave Pasajero .....	53
Ilustración 24 I-Relación Revisión Pasajero .....	53
Ilustración 25 I-Relación Revisión Aeronave .....	53
Ilustración 26 I-Diagrama de clases C1 .....	54
Ilustración 27 I-Transacción Revisión .....	55
Ilustración 28 Crear nuevo videoclub.....	64
Ilustración 29 Crear nueva película .....	64
Ilustración 30 Registrar socio .....	65
Ilustración 31 Alquilar película .....	66
Ilustración 32 Realizar estudio estadístico .....	67
Ilustración 33 Crear Administrador .....	67
Ilustración 34 OO-Crear nuevo videoclub .....	75
Ilustración 35 OO-Crear nueva película.....	76
Ilustración 36 OO-Registrar socio.....	77
Ilustración 37 OO-Alquilar película.....	78
Ilustración 38 OO-Realizar estudio estadístico.....	79
Ilustración 39 OO-Crear Administrador.....	80
Ilustración 40 OO-Diagrama de clases C2.....	81
Ilustración 41 I-Atributos Videoclub.....	82
Ilustración 42 I-Servicios Videoclub.....	83
Ilustración 43 I-Atributos Película .....	84
Ilustración 44 I-Servicios Película .....	85
Ilustración 45 I-Atributos Socio .....	86
Ilustración 46 I-Servicios Socio .....	87
Ilustración 47 I-Atributos Alquilar Película .....	88
Ilustración 48 I-Servicios Alquilar Película .....	89

Ilustración 49 I-Atributos Realizar estudio estadístico .....	90
Ilustración 50 I-Servicio Realizar estudio estadístico.....	91
Ilustración 51 I-Atributos Administrador.....	92
Ilustración 52 I-Servicios Administrador.....	93
Ilustración 53 I-Relación Película Videoclub .....	94
Ilustración 54 I-Relación Socio Videoclub.....	95
Ilustración 55 I-Relación Socio Alquiler .....	95
Ilustración 56 I-Relación Película Alquiler .....	95
Ilustración 57 I-Relación Estadística Socio.....	96
Ilustración 58 I-Relación Estadística Administrador .....	96
Ilustración 59 I-Diagrama de clases C2 .....	97
Ilustración 60 I-Derivación Total a Pagar .....	98
Ilustración 61 I-Derivación Total gastado .....	99



# 1. Introducción

---

## 1.1. Motivación de este trabajo

Este documento, presenta una demostración de un método OO conceptual. Esta se aplica mediante una derivación manual a través de varios ejemplos. Se toma como entrada un modelo de requisitos, que mediante el análisis de la comunicación del mismo se procede a resolver el problema aplicando dicho método-OO conceptual. Gracias a ello, se obtendrá un modelo conceptual en la herramienta Integranova Modeler. Dichas demostraciones, corresponden a dos ejemplos aplicados en la asignatura de Ingeniería de Sistemas de la Información.

## 1.2. Alcance del documento

Comenzando por una breve introducción al tema, se presentará el modelo a aplicar, en los ejemplos anteriormente comentados. Tras ello, se procederá a proponer una metodología de proceso aplicado en este campo. Para finalizar, se hará un inciso en una conclusión sobre el método aplicado y sus características.

Esta demostración, además de ilustrar la técnica aplicada, demuestra que la técnica es posible de poner en práctica. Además, dichos resultados de esta demostración proporcionan una mejora hacia las técnicas de derivación.

## 2. Fundamentos teóricos y tecnológicos

---

### 2.1. Model Driven Architecture

Model Driven Architecture (MDA) surgió como una propuesta de estandarización del enfoque Model Driven Development (MDD) para el desarrollo de software. Para ello, su principal regla es la de usar estándares con la mayor frecuencia posible, para poder cumplir sus objetivos. Actualmente, el corazón de la arquitectura está estandarizada por el OMG (Object Management Group), estos realizan el proceso de desarrollo de software mediante el uso de modelos como componentes. Estos modelos son elementos esenciales en la cadena de producción de software, gracias a ello, se ha conseguido establecer un conjunto de lenguajes estándares, para poder tratar eficientemente los modelos. En particular se consideran dos principales modelos: el Modelo Independiente de Plataforma (PIM), y el Modelo Dependiente de Plataforma (PSM).

#### 2.1.1. Modelo Independiente de Computación

CIM, es el modelo independiente de la computación, surge en la fase inicial del proceso de desarrollo comprendiendo la modelación del negocio en su totalidad.

Este modelo forma parte de los elementos a considerar en esta propuesta, el CIM no muestra detalles de la estructura del sistema, gracias a este enfoque se favorece el desarrollo de aplicaciones empresariales completamente funcionales a partir de modelos con UML y otros estándares que garanticen la independencia de las diversas plataformas, abiertas o propietarias, como servicios web, .Net, Corba, J2EE, u otras.

#### 2.1.2. Modelo Independiente de Plataforma

PIM, conforma el mayor nivel de abstracción, que un modelo puede tener. Este describe estructura, funcionalidad y restricciones que lleva el propio sistema, a su vez, omite detalles aplicados a la plataforma.

Este modelo servirá de base para todo el proceso de desarrollo. Al no incluir detalles específicos, relacionados con la tecnología que se va a aplicar, este modelo es útil en varios aspectos. Su facilidad de comprensión es mayor, por lo que los usuarios del sistema les será más sencillo comprender y validar el sistema. También reduce la complejidad para facilitar diferentes implementaciones para los respectivos sistemas con lo que se quiera trabajar, dejando intacta su estructura y funcionalidades básicas.

### 2.1.3. Transformación de CIM a PIM

Existen distintos lenguajes empleados en una transformación de un modelo CIM a un modelo PIM:

- Mediante UML, para construir un PIM para modelado multidimensional en almacenes de datos. Se consideran las fuentes de datos operacionales para completar el PIM inicial según las formas normales multidimensionales BPMN 2.0
- Mediante guías propuestas por SOD-M, mapeo de requisitos a nivel CIM en elementos de modelos de nivel PIM, para obtener dicho modelo.

Dichos lenguajes, son la base de MDA, puesto que dependiendo de la elección que se vaya a utilizar, se podrá elegir una herramienta concreta que pueda tratar el modelo PIM.

### 2.1.4. Modelo Dependiente de Plataforma

PSM, contiene la información del PIM, aplicada a una plataforma para realizar una implementación (.NET, J2EE, etc.). Para obtener este modelo, se realizan una serie de transformaciones a partir del PIM, que se llevan a cabo utilizando la herramienta MDA con la finalidad de llegar a la plataforma destino. Eventualmente, algunos parámetros se deberán proporcionar mediante un analista y/o programador con conocimientos. A partir de un mismo PIM, se pueden crear distintos PSM's cada uno para su respectiva plataforma.

### 2.1.5. Transformación de PIM a PSM

Existen distintas formas de pasar de un modelo PIM a uno PSM:

- Realizando el modelo PSM de forma manual, partiendo del modelo PIM.
- De forma semiautomática, generando una base del PSM que será completada a mano.
- Automática, existen distintas herramientas especializadas que transforman un modelo PIM a un modelo PSM concreto. Dichas herramientas, aplican una serie de algoritmos que tienen implementados.

Las herramientas de transformación automática son un pilar en MDA, proporcionando una ayuda muy valiosa.

## 2.2. OO-Method

Partiendo de la base, en la cual se tiene los requisitos del modelo, este se puede dividir en dos propuestas complementarias. El diagrama de comunicación y la descripción de los eventos, de los cuales se realizan distintas transformaciones para obtener cuatro modelos complementarios entre sí, de los cuales se obtiene un modelo conceptual. Estos son; modelo de objeto, modelo dinámico, modelo funcional y modelo de presentación.

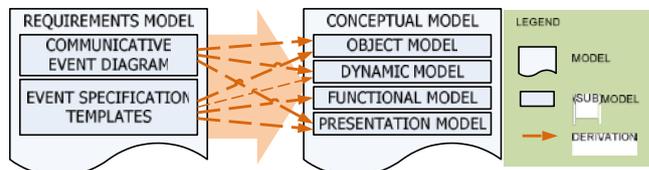


Ilustración 1 OO-Method

### 2.2.1. Modelo de Objeto

El modelo de Objeto se caracteriza por ser un diagrama de UML extendido. Para poder obtener el modelo de Objeto, se han de procesar los distintos eventos de comunicación. Además, por cada evento, se le especifican los requisitos relacionados con el mismo, obtenidos a partir de la propia descripción del evento, para obtener en detalle el susodicho evento. Específicamente, por cada evento comunicativo, se procesa en el orden la estructura del mensaje para derivar en un diagrama de clases, una vez se juntan todos los eventos del diagrama de comunicación.

### 2.2.2. Modelo Dinámico

El modelo dinámico, representa el ciclo de vida de los distintos objetos, especificando los detalles sobre la interacción que realiza el objeto. Todo esto, es representado mediante un diagrama de transición de estado. Por cada modelo de Objeto, se obtiene un diagrama que explica todo el ciclo de vida del objeto. Para obtener el modelo dinámico, se parte del totalmente del diagrama de comunicación, además, se puede complementar con pequeños detalles de la descripción del evento obteniendo un modelo con mayor riqueza.

### **2.2.3. Modelo Funcional**

Mayoritariamente, para este modelo se deriva de la descripción de eventos. Este modelo, como su nombre indica, se encarga de describir las funciones que tendrá cada evento, proporcionando en conjunto todas las funciones que tendrá el modelo es su totalidad.

En esta metodología aplicada, la herramienta Integranova se encargará de realizar dicho modelo.

### **2.2.4. Modelo de Presentación**

El modelo de presentación, se encarga de representar la función de la interfaz que tendrá el modelo. Este modelo consta de una derivación de la descripción de los eventos juntos con el modelo de objetos, para obtener una interfaz rica en detalle.

Para este modelo, como con el anterior modelo, se delegara la funcionalidad de este modelo a la herramienta Integranova.

## 2.3. Integranova Modeler

Integranova M.E.S. (Model Execution System) es un sistema disponible comercialmente que genera aplicaciones de negocios completas a partir de Modelos Conceptuales. La solución MDA, capaz de programar el código de la aplicación final, incluyendo la lógica de negocio sin importar la complejidad de esta.

El objetivo de esta es proporcionar un marco completo de desarrollo basado en modelos, gracias a ello, cerrar la brecha entre los usuarios finales de negocios y la TI mediante el aumento del nivel de abstracción dentro del proceso de desarrollo de software. Esta herramienta se centra en el espacio del problema, los usuarios finales. Gracias a ello, se reduce sustancialmente el tiempo de desarrollo tan valorado por los desarrolladores. La herramienta garantiza una gestión personalizada fiable, rica en funciones, fácil de mantener, escalable y segura.

### 2.3.1. ¿Cómo funciona?

La empresa divide el proceso del software en 2 diferentes fases; creación del modelo(PIM), a través del modelado de Integranova y el compilador de modelos, el cual se divide en; modelo XML, cliente Integranova Star y el proceso de la generación de código los cuales forman parte del PSM. A su vez, existe una división de tareas, las realizadas por el usuario que vaya a proceder a resolver un problema desarrollando una aplicación y la de crear la aplicación final.

### 2.3.2. PIM - Integranova Modeler

#### Creación del modelo

Esta es la única fase en la que se encarga el usuario, cliente de la aplicación, de realizar completamente. A su vez, esta fase es la única que no está completamente automatizada.

Tal y como se indica, el desarrollador se encarga de realizar el modelo PIM del programa, a través de la aplicación para modelar. En ella indica las clases, la relación entre ellas, atributos, servicios que presenta cada clase, transacciones, usuario, precondiciones...

### 2.3.3. PSM – Compilador de Modelos

#### Modelo XML

Una vez realizado el modelo, el usuario podrá validar su modelo. Al realizar esta, si todo está correctamente indicado y conserva una coherencia, la propia herramienta procederá a transformar el modelo a código XML.

#### Cliente Integranova Star

Una vez que el modelo se encuentra en XML, la herramienta proporciona la funcionalidad, a través del cliente Integranova Star, de transformar dicho modelo en el producto final. El cliente mostrará las distintas opciones de generación del programa; aplicado a distintos sistemas operativos, lenguajes de programación...

#### Proceso de la generación de código

Cuando ya se ha realizado todo el proceso del cliente, se envía el modelo XML con las características seleccionadas. Estos datos los recibe el servidor central, el cual procesa la información y devuelve un programa totalmente operativo con las condiciones indicadas en el cliente.

# 3. Método para generar modelos ejecutables a través del modelo de requisitos

---

En este capítulo, se va a detallar el método a aplicar para los siguientes ejemplos, detallado en los siguientes capítulos. Los ejemplos se centrarán en el modelo CIM y la transformación a PIM, puesto que una vez el modelo PIM este completo el resto del proceso se realizará de forma automática.

## 3.1. Modelo Independiente de Computación

Este apartado tratara el modelo CIM dividido en 3 aspectos;

### Descripción contextual del caso

En este apartado, esta concurrido toda la información original del caso de forma descriptiva, la cual se procesará y se adaptara para crear los formularios de negocio, los cuales se utilizarán más adelante en el análisis comunicativo de negocio.

### Análisis comunicativo de negocio

En este apartado, se procederá a analizar los distintos eventos de creación de los objetos que tendrá el programa, los distintos eventos serán descritos con el nivel de detalle suficiente como para poder entender el proceso del mismo, desde los requisitos de entrada como la reacción de los requisitos al finalizar el evento.

### Modelo conceptual aplicado a la metodología-OO

Para finalizar, este apartado detallará las distintas comunicaciones entre los eventos, así como las reacciones de los mismos.

## 3.2. Modelo Independiente de Plataforma

Este apartado tratara el modelo PIM, utilizando la herramienta Integranova Modeler.

### Construcción del modelo conceptual en Integranova

En este apartado, se procederá a modelar los distintos eventos detallados en el apartado anterior.

### ¿Cómo abordar la posible automatización completa del proceso de generación?

Para finalizar, se abordará el modelo añadiendo las distintas características que se necesitan para terminar el modelo, las cuales no se han podido obtener del modelo CIM anterior, y necesitan ser implementadas.

## 3.3. Modelo Dependiente de Plataforma

A partir del modelo PIM, el resto del proceso es generado de forma automática, por lo tanto, no será detallado en los siguientes casos de estudio.

## 4. Caso de Estudio 1

---

### 4.1. Motivación de este caso

Este documento presenta una demostración, que ejemplifica la derivación manual de un método OO Conceptual, tomando como entrada un modelo de requisitos de análisis de la comunicación.

La demostración de laboratorio corresponde a los procesos de negocio de un sistema de gestión de naves marcianas. Esta compañía, se encarga de realizar transportes intergalácticos.

Esta demostración, ilustra la técnica de derivación, demuestra que la puesta en práctica de dicha técnica.

## 4.2. Descripción contextual del caso

### 4.2.1. Descripción general

En el espacio existen multitud de aeronaves de transporte para trasladar marcianos entre distintas naves nodriza. Estas aeronaves son una especie de autobuses de marcianos entre distintas naves nodrizas (donde habitan los marcianos). El sistema debe gestionar los movimientos que los marcianos realizan en las aeronaves para alcanzar las naves nodrizas. La funcionalidad del sistema se puede resumir en estos cuatro puntos:

- Crear nave nodriza
- Crear aeronave
- Gestionar pasajeros
- Revisar aeronave

### 4.2.2. Funciones del sistema

#### Crear nave nodriza

Esta funcionalidad almacena los datos de las naves nodriza existentes. De cada nave nodriza sólo se desea almacenar el nombre y un identificador.

#### Crear aeronave

Esta funcionalidad pretende almacenar todas las aeronaves existentes. La aeronave irá de una nave nodriza origen a una nave nodriza destino (origen y destino puede ser la misma nave nodriza en algunas ocasiones). En el sistema se almacena una ficha de aeronave como la mostrada en el Anexo "Aeronave".

#### Gestionar pasajeros

A una aeronave suben y bajan distintos pasajeros. El sistema debe proporcionar la funcionalidad para registrar los pasajeros que suben a la aeronave y los pasajeros que bajan. Cada vez que un pasajero se sube a una aeronave, se debe introducir en el sistema la información que aparece en el Anexo "Asignar pasajero". Cada vez que un pasajero se baja de la aeronave al llegar al destino, en el sistema se debe introducir la información mostrada en el Anexo "Bajar pasajero". No pueden subir más pasajeros que el máximo permitido.

#### Revisar aeronave

Una vez al día, un revisor entra en todas las aeronaves y controla que los marcianos han pagado el billete. Esto sirve además para llevar un control de los marcianos que había subidos en ese momento en la aeronave. En cada aeronave sólo puede entrar un revisor por día. Por cada revisión, el sistema debe almacenar la información mostrada en el Anexo "Revisión".

### 4.2.3. Características de los usuarios

El sistema será utilizado por los trabajadores de dirección general de tráfico espacial (DGTE), que son los encargados de dirigir todo el tráfico intergaláctico.

### 4.2.4. Requisitos Específicos

#### Crear nave nodriza

##### Entrada

De cada nave nodriza sólo se almacena el nombre y un identificador.

##### Proceso

Las naves nodriza son como ciudades flotantes donde viven los marcianos. Se pueden crear varias naves nodrizas conforme va avanzando el tiempo.

##### Salida

Los datos de la nave nodriza quedan registrados en el sistema. Los trabajadores de la DGTE deben poder consultar la lista de naves nodrizas existentes.

#### Crear aeronave

##### Entrada

Los datos a introducir son un Id para la aeronave, un nombre, el máximo de marcianos que puede transportar, la nave nodriza de la que parte (origen) y la nave nodriza hacia la que va (destino). Toda la información a introducir está representada en el Anexo "Aeronave".

##### Proceso

Las aeronaves se crean para hacer la ruta entre dos naves nodriza transportando marcianos. En algunos casos, las aeronaves se crean para que los marcianos turistas puedan hacer recorridos por el espacio, y en estos casos, la nave nodriza origen y destino es la misma.

##### Salida

Se almacenan en el sistema los datos de la nueva aeronave. Los empleados de la DGTE pueden consultar la lista de aeronaves existentes.

#### Gestionar pasajeros

##### Entrada

La gestión de los pasajeros incluye dos funcionalidades: la de asignar un pasajero a la aeronave y la de bajar un pasajero de la aeronave. Para asignar un pasajero a la aeronave se necesita el Id del pasajero y el Id de la aeronave en la cual se ha subido. Toda esta información se puede ver en el Anexo "Asignar pasajero". Si el pasajero no está dado de alta en el sistema, se debe crear antes de asignarlo a la aeronave. Para crear el pasajero, sólo es necesario un Id y su nombre. Para registrar que un pasajero ha bajado de la aeronave cuando ésta llega a su destino, basta con indicar en el sistema el Id de la aeronave y el Id del pasajero, tal y como aparece en el Anexo "Bajar pasajero".

##### Proceso

Los pasajeros suben y bajan de las aeronaves cuando éstas llegan a una nave nodriza. Cuando un pasajero sube a la aeronave, se debe registrar esta información en el sistema. Una restricción es que en una aeronave no pueden subir más pasajeros que el máximo de marcianos permitidos para esa aeronave. Cuando un pasajero baja porque ha llegado a la nave

nodriza destino, esta información se debe almacenar también. De esta forma la información de los pasajeros que hay en cada una de las aeronaves está siempre actualizada.

**Salida**

La información de los pasajeros que están actualmente subidos a una aeronave debe estar actualizada en todo momento. Además, la lista debe indicar el número de pasajeros que hay en ese momento en la aeronave. Los empleados de la DGTE pueden consultar la lista de pasajeros que hay en cada una de las aeronaves.

## Revisar aeronave

**Entrada**

El formulario a rellenar es el mostrado en el Anexo “Revisión”. En este formulario se debe rellenar el Id de la revisión, el nombre del revisor, el Id de la aeronave revisada, la fecha en la que se produce la revisión y la lista de pasajeros que había en ese instante en la aeronave.

**Proceso**

Una vez al día, se sube un revisor empleado de la DGTE en cada una de las aeronaves y registra en un histórico los pasajeros que hay en ese momento en la aeronave. El sistema sólo debe permitir que haya una única revisión por día para cada aeronave. Al registrar la revisión en el sistema, el usuario sólo introduce el Id de la revisión, el nombre del revisor, el id de la aeronave revisada y la fecha. La lista de todos los marcianos que van en la aeronave se debe obtener de forma automática en base a la información que ya hay actualmente en el sistema (previamente, los marcianos han sido asignados como pasajeros y aún no han bajado de la aeronave). La información de la ficha Revisión queda almacenada en el sistema como un histórico (es decir, aunque el marciano se apee de la aeronave posteriormente, la información de que viajaba en ella en el momento de la revisión debe permanecer almacenada). Esta información es útil para controlar los movimientos de los marcianos entre las distintas naves nodrizas a lo largo del tiempo.

**Salida**

Un histórico en el que los empleados de la DGTE pueden listar los pasajeros que había por aeronave en el momento de hacer la revisión. Además, de la información de los pasajeros y de la aeronave, se debe mostrar la fecha en la que se produjo la revisión.

## 4.2.5. Formularios de negocio

### Crear nave nodriza

Crear nave nodriza	
Nombre	Orión
Identificador	N0000001

Campo	Descripción
Nombre	Nombre asignado a la nave
Identificador	Código único para diferenciar las naves nodrizas entre si

### Crear aeronave

Crear aeronave	
Nombre	U Orionis
Identificador	A0000001
Máximo de marcianos	200
Nave nodriza origen	Orión
Nave nodriza destino	Orión

Campo	Descripción
Nombre	Nombre asignado a la aeronave
Identificador	Código único para diferenciar las aeronaves entre si
Máximo de marcianos	Capacidad máxima de pasajeros marcianos de la aeronave
Nave nodriza origen	Origen del viaje
Nave nodriza destino	Destino del viaje

## Gestionar pasajeros

### *Asignar pasajero*

Asignar pasajero	
Identificador pasajero	P0000001
Identificar aeronave	A0000001

Campo	Descripción
Identificador pasajero	Código único para diferenciar los pasajeros entre si
Identificar aeronave	Código único para diferenciar las aeronaves entre si

### *Bajar pasajero*

Bajar pasajero	
Identificador pasajero	P0000001
Identificar aeronave	A0000001

Campo	Descripción
Identificador pasajero	Código único para diferenciar los pasajeros entre si
Identificar aeronave	Código único para diferenciar las aeronaves entre si

### *Crear pasajero*

Crear pasajero	
Identificador pasajero	P0000001
Nombre pasajero	Juan Gómez

Campo	Descripción
Identificador pasajero	Código único para diferenciar los pasajeros entre si
Nombre pasajero	Nombre asignado al pasajero

## Revisar aeronave

Revisión		
Identificador revisión	R0000001	
Nombre del revisor	Pedro Martínez	
Identificador aeronave	A0000001	
Fecha de la revisión	15/06/2017	
	ID Marciano 1	P0000001
	ID Marciano 2	P0000002
	...	
	ID Marciano N	P9999999

Campo	Descripción
Identificador revisión	Código único para diferenciar los revisores entre si
Nombre del revisor	Nombre asignado al revisor
Identificador aeronave	Código único para diferenciar las aeronaves entre si
Fecha de la revisión	Fecha de la revisión
Listado de marcianos a bordo de la aeronave	Lista con los identificadores de los pasajeros

## 4.3. Análisis comunicativo de negocio

### 4.3.1. Introducción

El futuro sistema se llamará SIGEM (Sistema de Gestión de Naves Marcianas), y pretende automatizar el control de las cargas y descargas de las naves de transporte, que actualmente se realiza a mano. Estas naves de transporte trasladan marcianos de una nave nodriza origen a una nave nodriza destino.

El trabajo que actualmente se realiza a mano ha sido dividido en procesos de negocio. La siguiente tabla introduce dichos procesos y los empleados que participan en ellos.

Proceso	Subproceso	Empleado encargado
Crear nave nodriza		Empleado
Crear aeronave		Empleado
	Crear pasajero	Empleado
	Asignar pasajero	Empleado
	Bajar pasajero	Empleado
Revisar aeronave		Revisor

### 4.3.2. Diagramas de comunicación

#### Crear nave nodriza

El proceso de creación de la nave nodriza es sencillo, el empleado del sistema introducirá los datos de la nave, tras ello, la nave estará creada en el sistema.

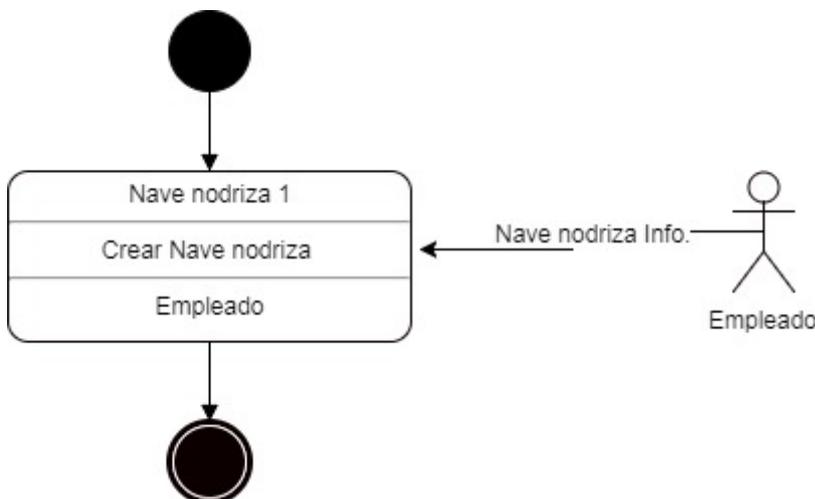


Ilustración 2 Crear nave nodriza

## Crear aeronave

El proceso de creación de la aeronave, es idéntico al de la nave nodriza.

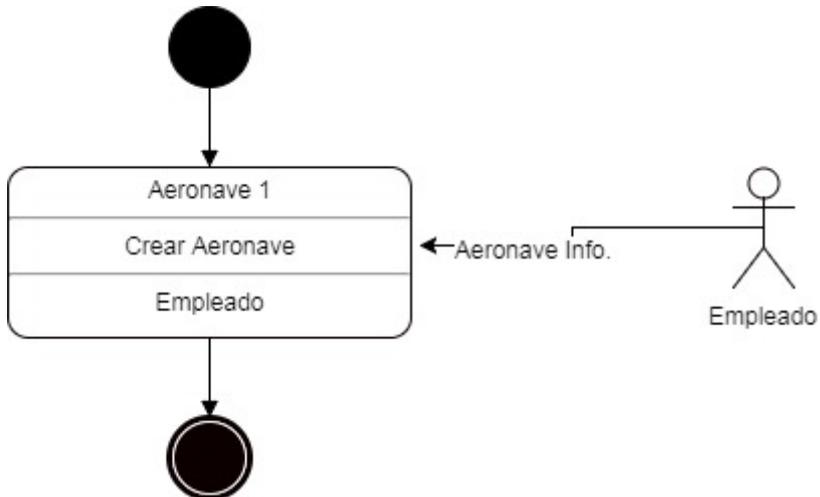


Ilustración 3 Crear aeronave

## Asignar pasajero

El proceso de asignación de pasajero a una aeronave difiere de los anteriores en la necesidad de comprobar si el pasajero está registrado en el sistema para poder proceder a la asignación del mismo en la aeronave. Los dos procesos que se observan en la imagen tienen la misma función que los anteriores, la inserción de la información del pasajero y la asignación del pasajero a la aeronave.

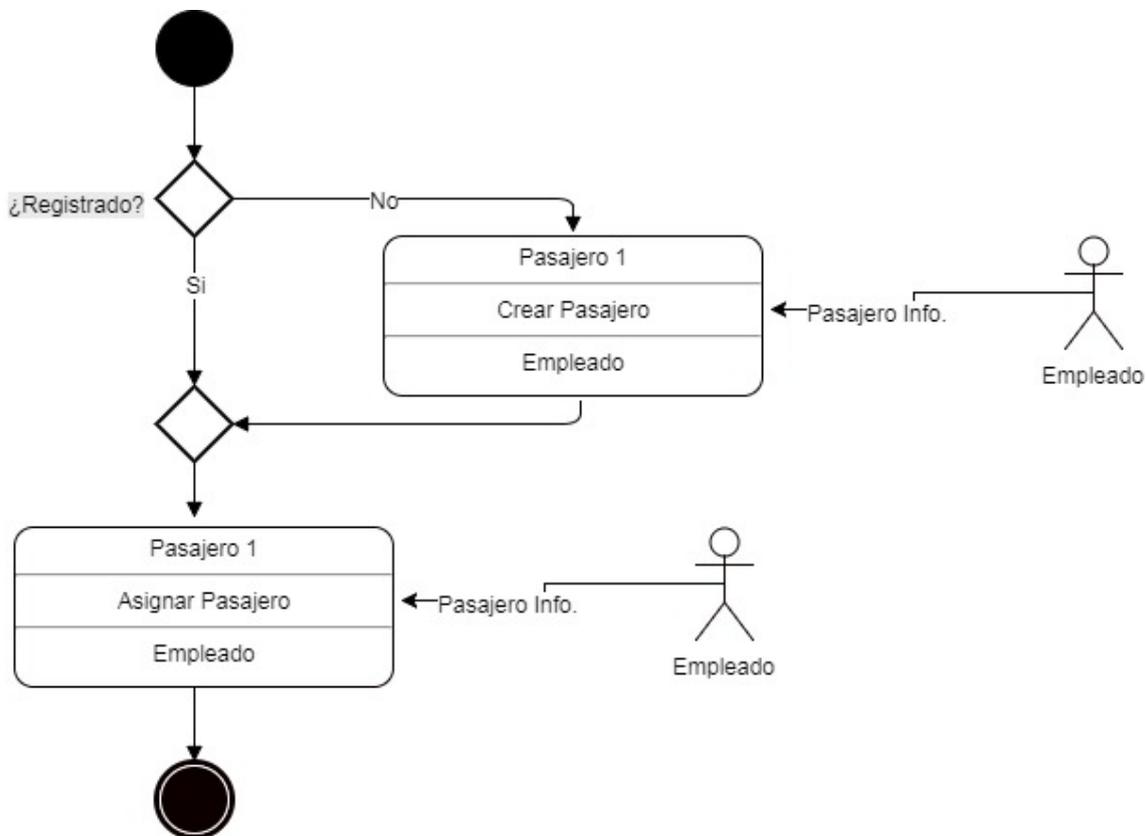


Ilustración 4 Asignar pasajero

## Bajar pasajero

En este proceso se procede a desasignar el pasajero de la aeronave a la cual estaba asignado.

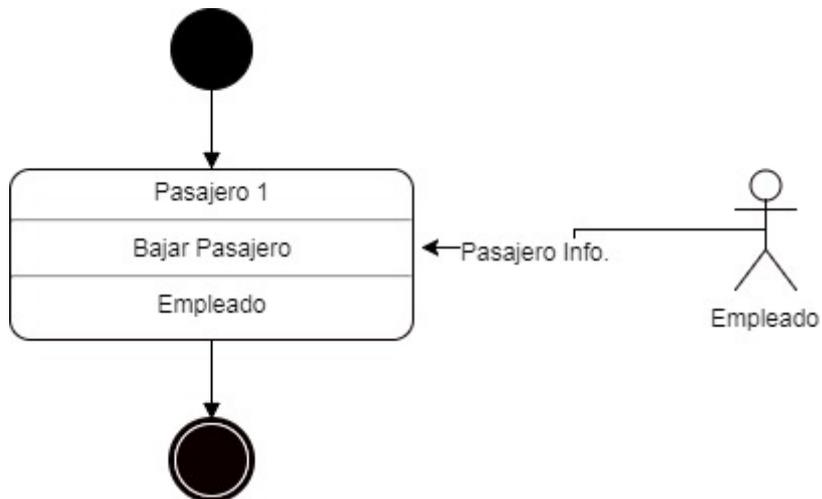


Ilustración 5 Bajar pasajero

## Crear pasajero

El proceso de creación del pasajero, es idéntico a los dos procesos de creación anteriores.

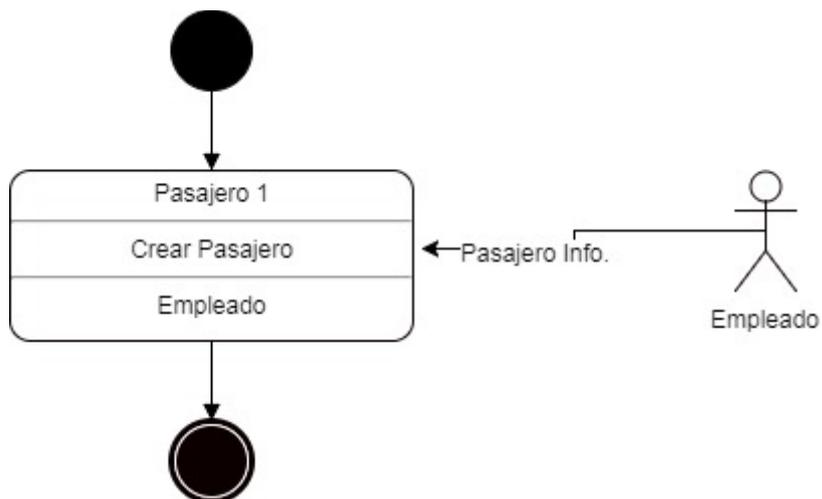


Ilustración 6 Crear pasajero

## Revisar aeronave

Este proceso, tiene como función principal la obtención por parte del sistema de un listado de los pasajeros que se encuentran actualmente en la aeronave, pero existe la precondition de que solamente se puede revisar la nave una vez al día.

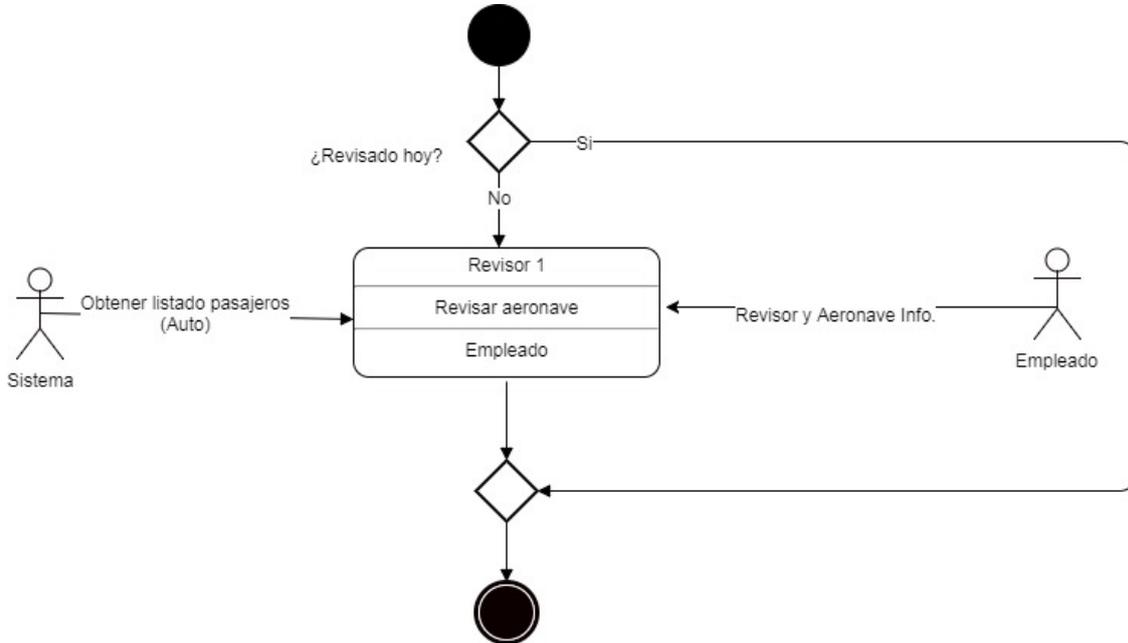


Ilustración 7 Revisar aeronave

### 4.3.3. Descripción de eventos

#### Crear nave nodriza

##### *Información general*

Esta funcionalidad almacena los datos de las naves nodriza existentes. De cada nave nodriza sólo se desea almacenar el nombre y un identificador.

##### *Requisitos generales*

###### **Actores responsables:**

- Empleado

###### **Formulario**

Crear nave nodriza	
Nombre	Orión
Identificador	N0000001

##### *Requisitos comunicativos*

Campo	Operación	Dominio	Ejemplo
NAVE = < Nombre + Identificador >	  	Text  Text	Orión  N0000001

Campo	Descripción
Nombre	Nombre asignado a la nave
Identificador	Código único para diferenciar las naves nodrizas entre si

###### **Restricciones estructurales**

El identificador de la nave nodriza ha de ser único

###### **Restricciones contextuales**

Las naves nodrizas son identificadas por su identificador

##### *Reacción a los requisitos*

###### **Tratamiento**

La nave nodriza queda creada.

## Crear aeronave

### Información general

Esta funcionalidad pretende almacenar todas las aeronaves existentes. La aeronave irá de una nave nodriza origen a una nave nodriza destino (origen y destino puede ser la misma nave nodriza en algunas ocasiones). En el sistema se almacena una ficha de aeronave como la mostrada en el Anexo "Aeronave".

### Requisitos generales

#### Actores responsables:

- Empleado

#### Formulario

Crear aeronave	
Nombre	U Orionis
Identificador	A0000001
Máximo de marcianos	200
Nave nodriza origen	Orión
Nave nodriza destino	Orión

### Requisitos comunicativos

Campo	Operación	Dominio	Ejemplo
AERONAVE =			
< Nombre +		Text	U Orionis
Identificador +		Text	A0000001
Más. Marcianos +		Num	200
N. Origen +		Text	Orión
N. Destino		Text	Orión
>			

Campo	Descripción
Nombre	Nombre asignado a la aeronave
Identificador	Código único para diferenciar las aeronaves entre si
Máximo de marcianos	Capacidad máxima de pasajeros marcianos de la aeronave
Nave nodriza origen	Origen del viaje
Nave nodriza destino	Destino del viaje

#### Restricciones estructurales

El identificador de la aeronave ha de ser único.

El origen y el destino puede coincidir.

#### Restricciones contextuales

Las naves nodrizas son identificadas por su identificador

### Reacción a los requisitos

#### Tratamiento

La aeronave queda creada.

## Asignar pasajero

### Información general

El sistema debe proporcionar la funcionalidad para registrar los pasajeros que suben a la aeronave. Si el pasajero no está dado de alta en el sistema, se debe crear antes de asignarlo a la aeronave. No pueden subir más pasajeros que el máximo permitido.

### Requisitos generales

#### Actores responsables:

- Empleado

#### Formulario

Asignar pasajero	
Identificador pasajero	P0000001
Identificar aeronave	A0000001

### Requisitos comunicativos

Campo	Operación	Dominio	Ejemplo
S_PASAJERO =			
< Identificador P.+		Text	P0000001
Identificador A.		Text	A0000001
>			

Campo	Descripción
Identificador pasajero	Código único para diferenciar los pasajeros entre si
Identificar aeronave	Código único para diferenciar las aeronaves entre si

#### Restricciones contextuales

Si el pasajero no está dado de alta en el sistema, se debe crear antes de asignarlo a la aeronave.

Se debe comprobar que no se va a superar la capacidad máxima de la aeronave antes de proceder a subir al pasajero.

### Reacción a los requisitos

#### Tratamiento

El pasajero queda asignado a la aeronave hasta que se baje.

## Bajar pasajero

### Información general

Para registrar que un pasajero ha bajado de la aeronave cuando ésta llega a su destino, basta con indicar en el sistema el Id de la aeronave y el Id del pasajero.

### Requisitos generales

#### Actores responsables:

- Empleado

#### Formulario

Bajar pasajero	
Identificador pasajero	P0000001
Identificar aeronave	A0000001

### Requisitos comunicativos

Campo	Operación	Dominio	Ejemplo
B_PASAJERO =			
< Identificador P.+	I	Text	P0000001
Identificador A.	I	Text	A0000001
>			

Campo	Descripción
Identificador pasajero	Código único para diferenciar los pasajeros entre si
Identificar aeronave	Código único para diferenciar las aeronaves entre si

### Reacción a los requisitos

#### Tratamiento

El pasajero queda borrado de la aeronave a la que estaba asignado.

## Crear pasajero

### Información general

Para poder asignar a un pasajero a una aeronave es obligado registrarlo primero en el sistema.

### Requisitos generales

#### Actores responsables:

- Empleado

#### Formulario

Crear pasajero	
Identificador pasajero	P0000001
Nombre pasajero	Juan Gómez

### Requisitos comunicativos

Campo	Operación	Dominio	Ejemplo
C_PASAJERO =			
< Identificador +	I	Text	P0000001
Nombre	I	Text	Juan Gómez
>			

Campo	Descripción
Identificador pasajero	Código único para diferenciar los pasajeros entre si
Nombre pasajero	Nombre asignado al pasajero

### Reacción a los requisitos

#### Tratamiento

El pasajero queda insertado en el sistema.

## Revisar aeronave

### Información general

Una vez al día, un revisor entra en todas las aeronaves y controla que los marcianos han pagado el billete. Esto sirve además para llevar un control de los marcianos que había subidos en ese momento en la aeronave. En cada aeronave sólo puede entrar un revisor por día.

### Requisitos generales

#### Actores responsables:

- Revisor

#### Formulario

Revisión		
Identificador revisión	R0000001	
Nombre del revisor	Pedro Martínez	
Identificador aeronave	A0000001	
Fecha de la revisión	15/06/2017	
	ID Marciano 1	P0000001
	ID Marciano 2	P0000002
	...	
	ID Marciano N	P9999999

### Requisitos comunicativos

Campo	Operación	Dominio	Ejemplo
REVISION =			
< Identificador R.+	I	Text	P0000001
Nombre R. +	I	Text	Juan Gómez
Identificador A.+	I	Text	A0000001
Fecha Revisión +	I	Date	15/06/2017
Listado Pasajeros	G	List	
>			

Campo	Descripción
Identificador revisión	Código único para diferenciar los revisores entre si
Nombre del revisor	Nombre asignado al revisor
Identificador aeronave	Código único para diferenciar las aeronaves entre si
Fecha de la revisión	Fecha de la revisión
Listado de marcianos a bordo se la aeronave	Lista con los identificadores de los pasajeros

### Reacción a los requisitos

#### Restricciones contextuales

El listado de pasajeros lo debe devolver el sistema al crear la nueva revisión.

## 4.4. Modelo conceptual aplicado a la metodología-OO

### 4.4.1. Derivación del modelado de objetos

Antes de aplicar las directrices de transformación, el diagrama de eventos comunicativos debe ser pre-procesado. Para ello se adoptan las siguientes acciones:

El diagrama de eventos comunicativos se extiende con eventos comunicativos de otros procesos.

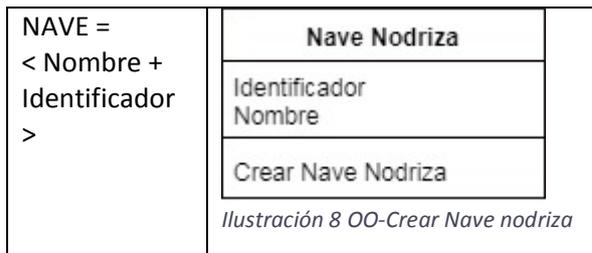
El diagrama de evento comunicativo extendido se elimina las relaciones de precedencia de bucle back (como, así como otros elementos) hasta que se obtiene un conjunto ordenado de eventos comunicativos.

El árbol dirigido raíz resultante es procesado para obtener una lista ordenada de eventos. Los sucesos se procesan en el orden definido, aplicando las directrices de derivación.

#### Crear nave nodriza

La estructura de mensajes del evento comunicativo Nave nodriza, consiste en una Agregación llamada NAVE.

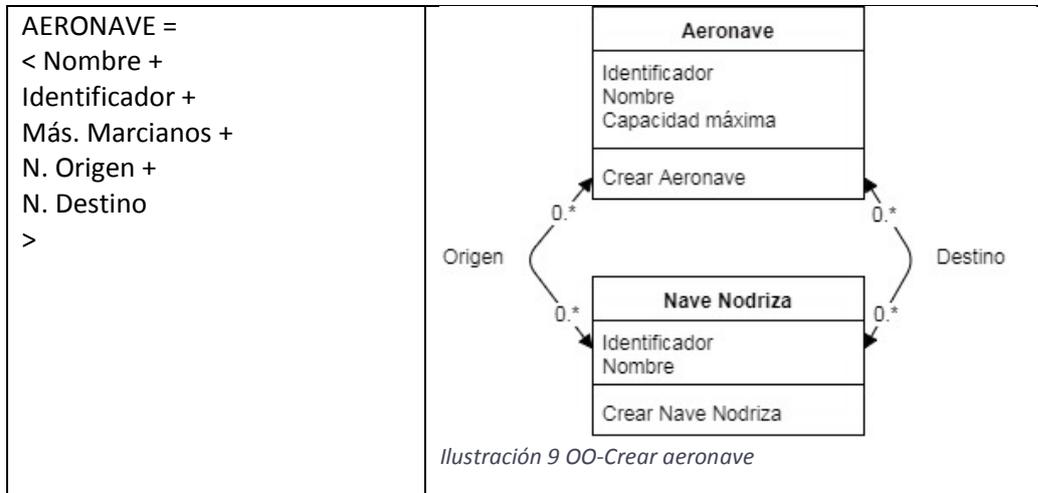
Especifica los datos de la Nave nodriza correspondiente. Además, se agrega un servicio de creación denominado Crear nave nodriza a la clase; Este servicio tiene tantos argumentos entrantes como atributos de clase se han derivado.



## Crear aeronave

La estructura de mensajes del evento comunicativo Aeronave, consiste en una Agregación llamada AERONAVE.

Especifica los datos de la Aeronave correspondiente. Además, se agrega un servicio de creación denominado Crear aeronave a la clase; Este servicio tiene tantos argumentos entrantes como atributos de clase se han derivado.



Teniendo en cuenta que parte de la información, los campos N. Origen y N. Destino, que se insertan a la hora de crear una nueva aeronave. Debido a ello se establece una relación con Nave Nodriza por cada campo.

## Gestionar pasajeros

La estructura de mensajes del evento comunicativo Pasajero, consiste en tres métodos.

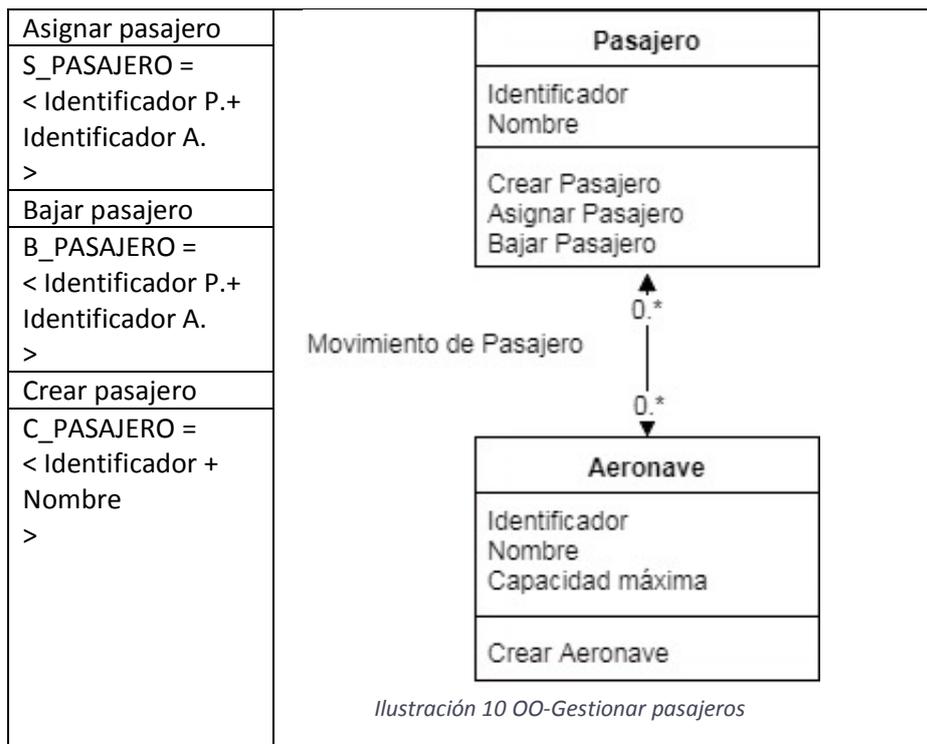
- Agregación llamada C\_PASAJERO.
- Evento llamado S\_PASAJERO.
- Evento llamado B\_PASAJERO.

Los dos servicios de evento comparten argumentos entrantes, mientras que el servicio de agregación tiene más atributos de clase.

C\_PASAJERO especifica los datos del Pasajero correspondiente.

S\_PASAJERO especifica el movimiento del pasajero de introducirse en una aeronave para viajar.

B\_PASAJERO especifica el movimiento del pasajero de bajarse de la aeronave, a la que previamente se había subido.

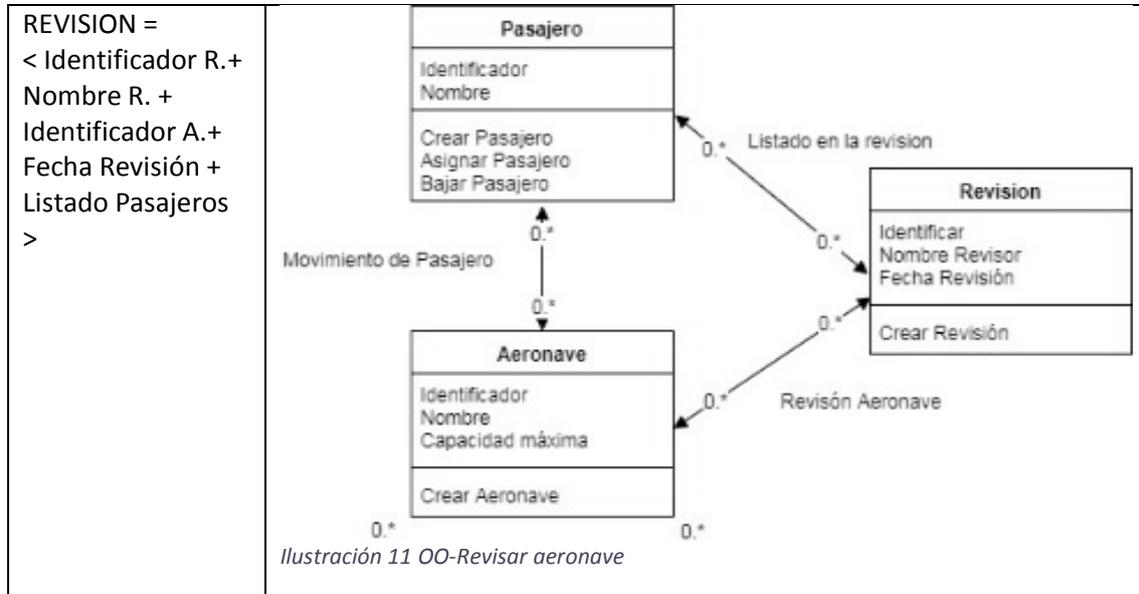


Teniendo en cuenta que en los dos eventos se especifica parte de la información de la aeronave, Identificador A., que se insertan a la hora de crear cualquiera de los dos eventos. Debido a ello se establece una relación con Aeronave por el campo.

## Revisar aeronave

La estructura de mensajes del evento comunicativo Revisión, consiste en un evento agregación llamado REVISION. Este servicio tiene tantos argumentos entrantes como atributos de clase se han derivado.

Además, tiene un método iterativo para obtener el listado de los pasajeros que ocupan la nave en ese momento del día.



Teniendo en cuenta que en el evento se especifica parte de la información de la aeronave y se obtiene un listado de pasajeros, Identificador A. y Listado Pasajeros, que se insertan a la hora de crear el evento. Debido a ello se establece una relación con Aeronave por el campo y una con Pasajero.

### 4.4.2. Diagrama de clases

Teniendo en cuenta todo lo explicado en el punto anterior, aquí se muestra el diagrama de clases resultante.

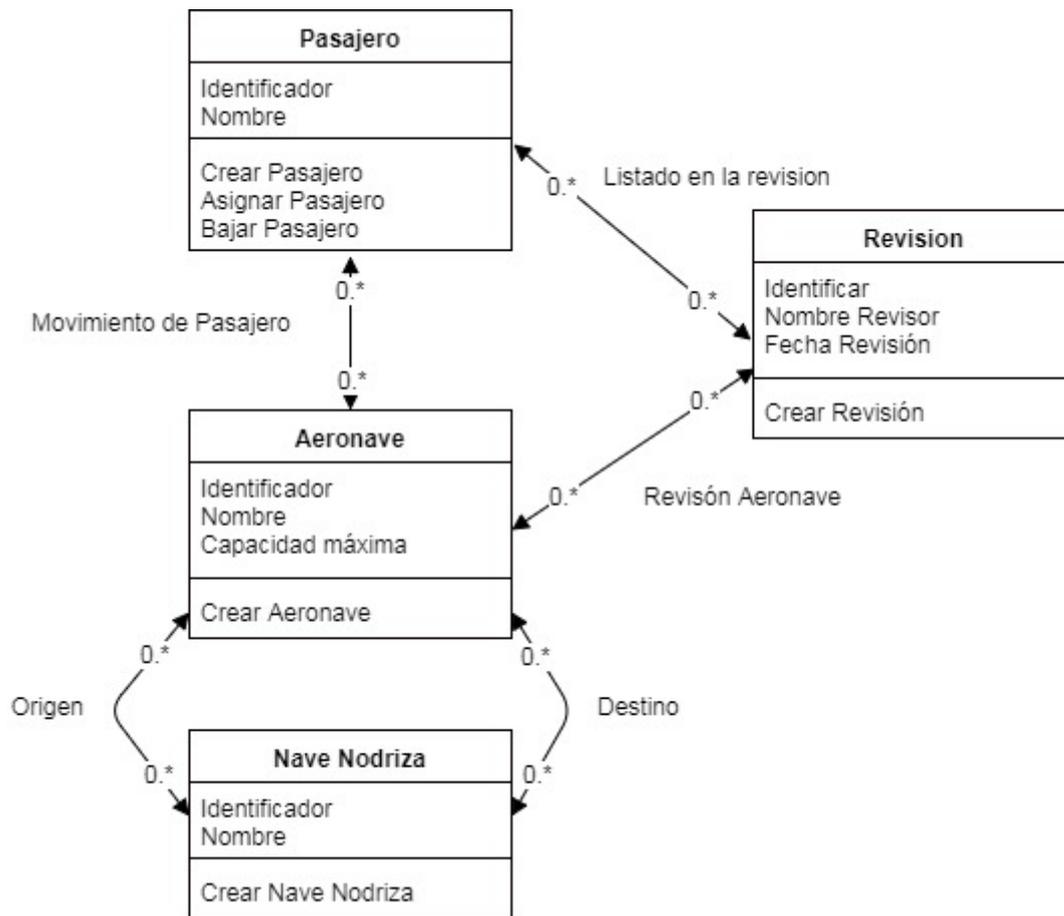


Ilustración 12 OO-Diagrama de clases C1

## 4.5. Construcción del modelo conceptual en Integranova

En la siguiente serie de imágenes se ilustra, la parte automatizable del modelo de clases creado utilizando a la herramienta Integranova.

### Nave nodriza

La siguiente imagen muestra la clase Nave Nodriza, junto a sus atributos.

Name	Attribute type	Data type	Id	S...	Default value	Request upon creation	Nulls
id_NaveNodr...	Constant	Autonumeric	Yes	No	40	Yes	No
nombre_Nav...	Variable	String	No	40		Yes	Yes

Ilustración 13 I-Atributos Nave nodriza

Como se puede observar se han creado dos atributos:

El identificador de la nave, el cual es un atributo constante y del tipo autonumérico.

El nombre de la nave, atributo variable de tipo String.

Dichos atributos han tenido que ser insertados a mano, por lo cual se ha tenido que proceder al uso de un analista para que lo añadiese.

En la siguiente imagen se muestran los servicios de la propia clase, los tres primeros son creados por defecto por Integranova y los dos últimos son creado cuando se introduce Aeronave, mostrado a continuación.

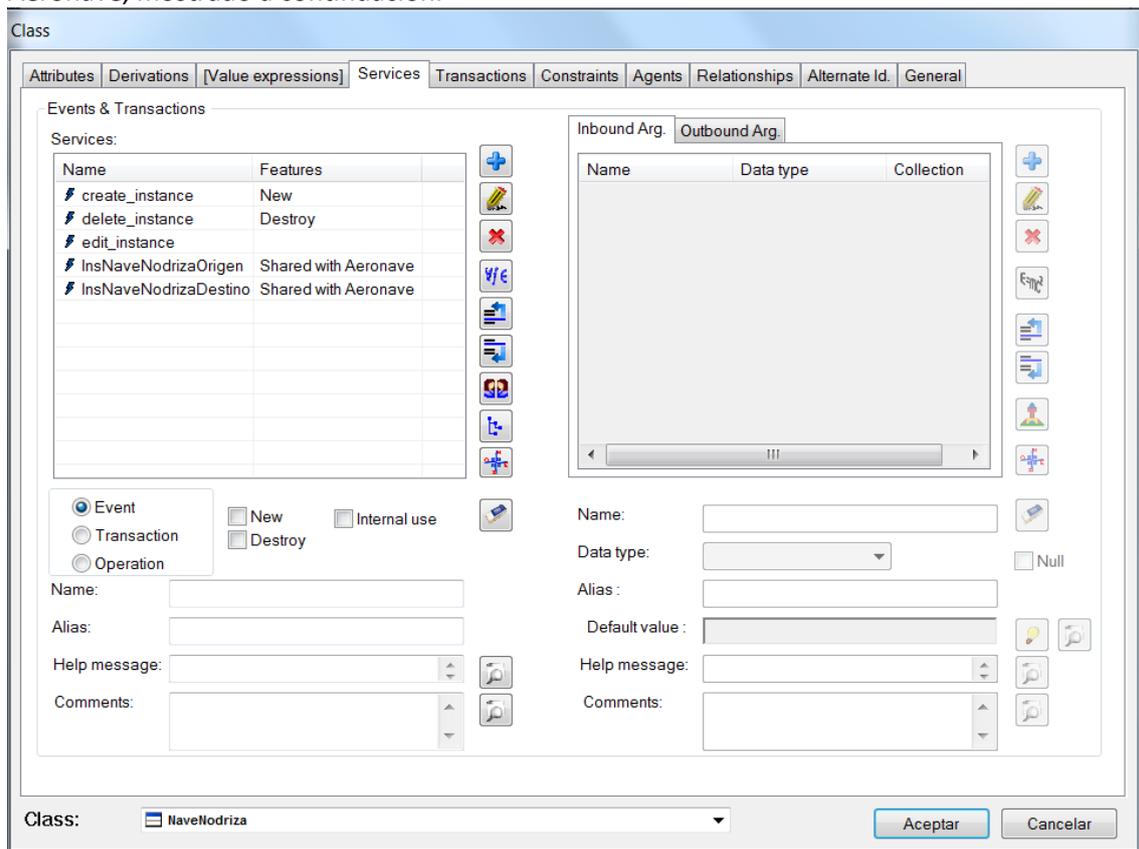


Ilustración 14 I-Servicios Nave Nodriza

Gracias a Integranova los procesos de creación, edición y destrucción, se han creado al crear la clase en el modelador. Debido a ello, no es necesario insertarlos a mano.

A su vez, los dos últimos procesos también han sido creado automáticamente, al añadir la relación de la clase Nave Nodriza con Aeronave, por lo tanto, tampoco será necesaria su posterior inclusión en el modelo.

## Aeronave

La siguiente imagen muestra la clase Aeronave, junto a sus atributos.

Class

Attributes | Derivations | [Value expressions] | Services | Transactions | Constraints | Agents | Relationships | Alternate Id. | General

Attributes

Name	Attribute type	Data type	Id	Size	Default v...	Request...	Nulls
id_Aeronave	Constant	Autonumeric	Y...			Yes	No
nombre_Aero...	Variable	String	No	40		Yes	Yes
maximo_Mar...	Variable	Int	No			Yes	Yes

Name:  Attribute type:

Alias:  Data type:

Help Message

Comments

Class:

Ilustración 15 I-Atributos Aeronave

En la imagen se reflejan los distintos atributos creados a mano por el analista.

En la siguiente imagen se muestran los servicios de la propia clase. Los tres primeros son creados por defecto por Integranova, los dos siguientes son creado cuando se especifica la unión de Aeronave junto con Nave Nodriza, se produce el mismo efecto con los últimos dos servicios, pero esta vez al unir Pasajero junto con Aeronave.

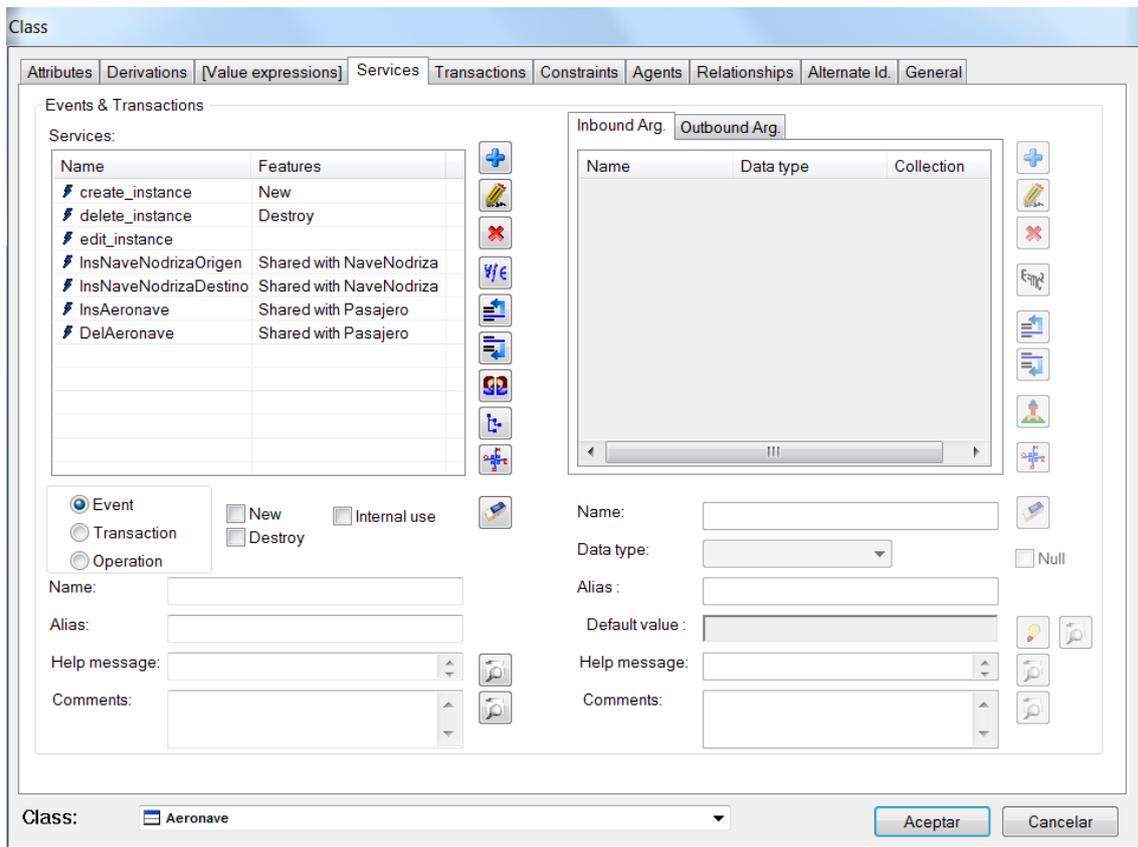


Ilustración 16 I-Servicios Aeronave

Se repite el mismo proceso que se ha producido con Nave Nodriza, ninguno de sus servicios se ha tenido que crear, Integranova lo ha realizado automáticamente.

## Pasajero

La siguiente imagen muestra la clase Aeronave, junto a sus atributos

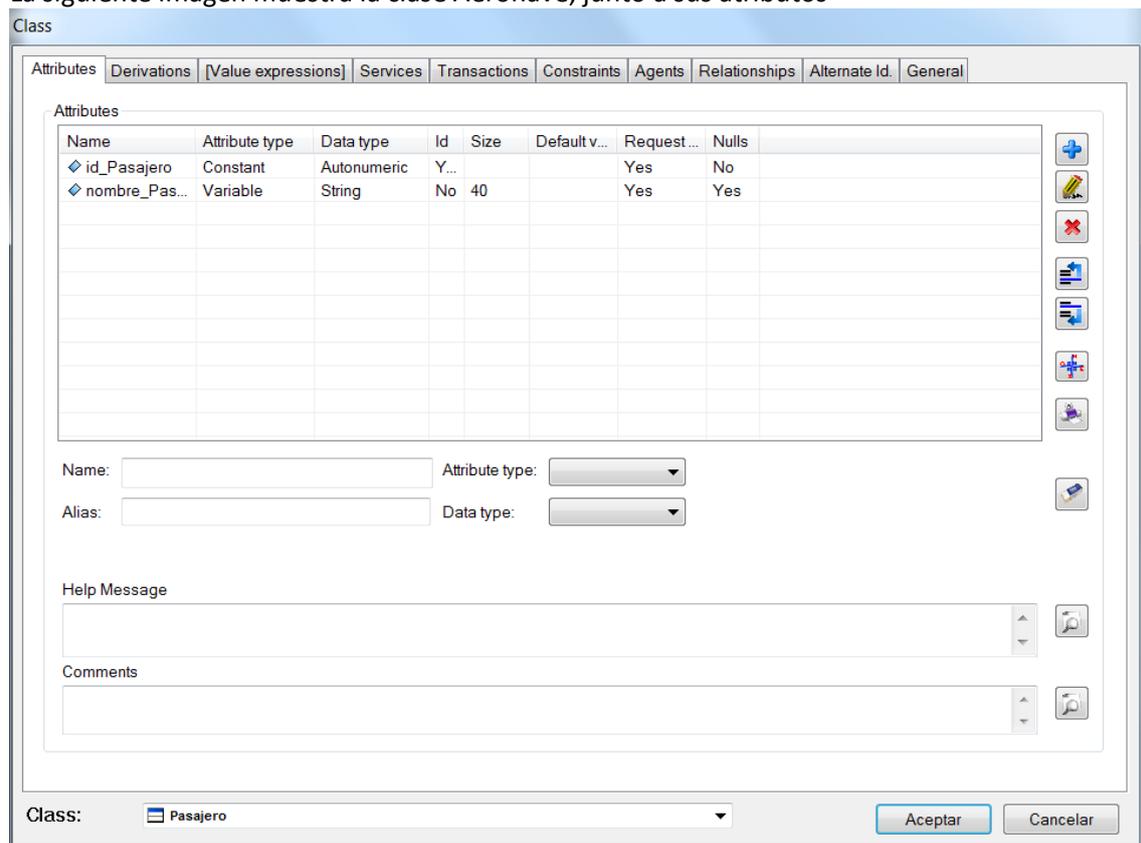


Ilustración 17 | Atributos Pasajero

En la imagen se reflejan los distintos atributos creados a mano por el analista.

En la siguiente imagen se muestran los servicios de la propia clase. Los tres primeros son creados por defecto por Integranova, el cuarto y el ultimo son creados cuando se especifica la unión de Aeronave junto con Pasajero, se produce el mismo efecto con los dos servicios restantes, pero esta vez al unir Pasajero junto con revisión.

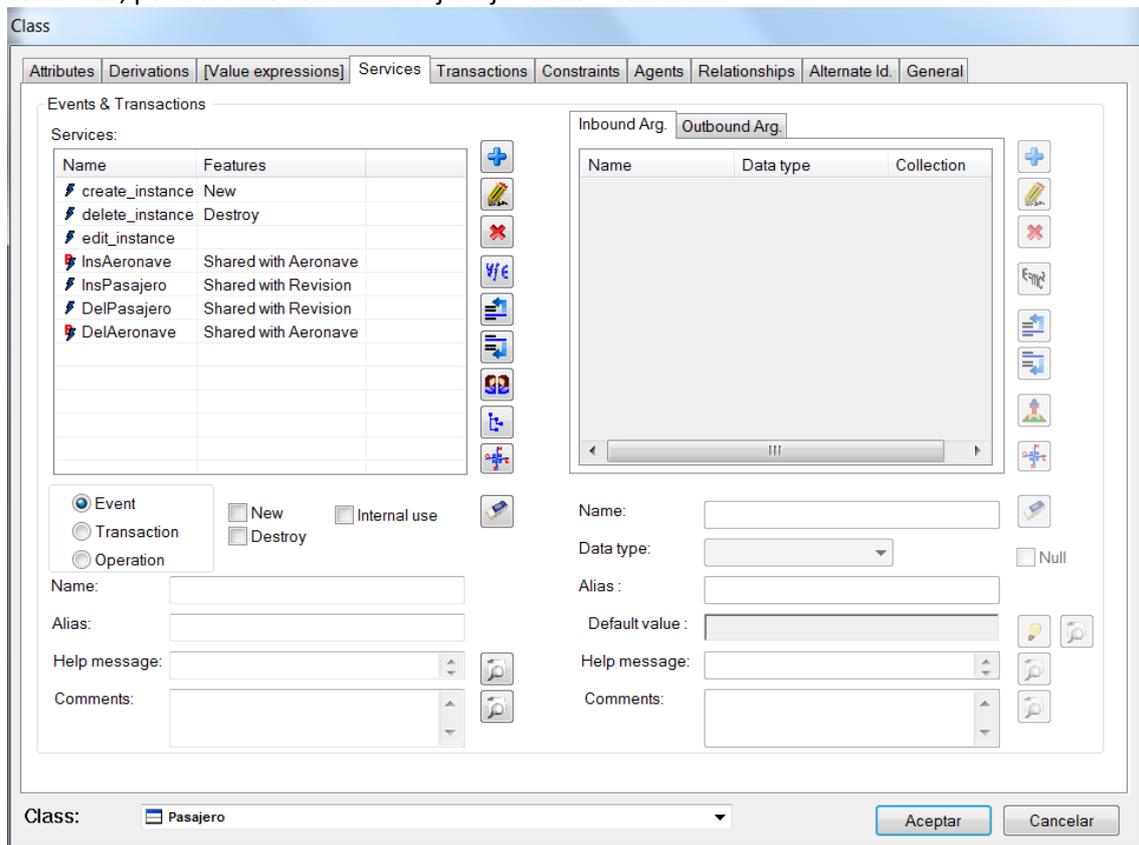


Ilustración 18 I-Servicios Pasajero

Se repite el mismo proceso que se ha producido con las anteriores clases, ninguno de sus servicios se ha tenido que crear, Integranova lo ha realizado automáticamente.

## Revisión

La siguiente imagen muestra la clase Aeronave, junto a sus atributos

Class

Attributes | Derivations | [Value expressions] | Services | Transactions | Constraints | Agents | Relationships | Alternate Id. | General

Attributes

Name	Attribute type	Data type	Id	Size	Default v...	Request...	Nulls
id_Revision	Constant	Autonumeric	Y...			Yes	No
nombre_Revi...	Variable	String	No	40		Yes	Yes
fecha_Revisi...	Variable	Date	No			Yes	Yes

Name:  Attribute type:

Alias:  Data type:

Help Message

Comments

Class:

Ilustración 19 I-Atributos Revisión

En la imagen se reflejan los distintos atributos creados a mano por el analista.

En la siguiente imagen se muestran los servicios de la propia clase. Los tres primeros son creados por defecto por Integranova, el cuarto y el quinto son creados cuando se especifica la unión de Revisión con Pasajero. El último servicio es creado a mano por el analista, este crea una transacción que se encargue de la revisión de la aeronave, la cual será detallada en el próximo punto.

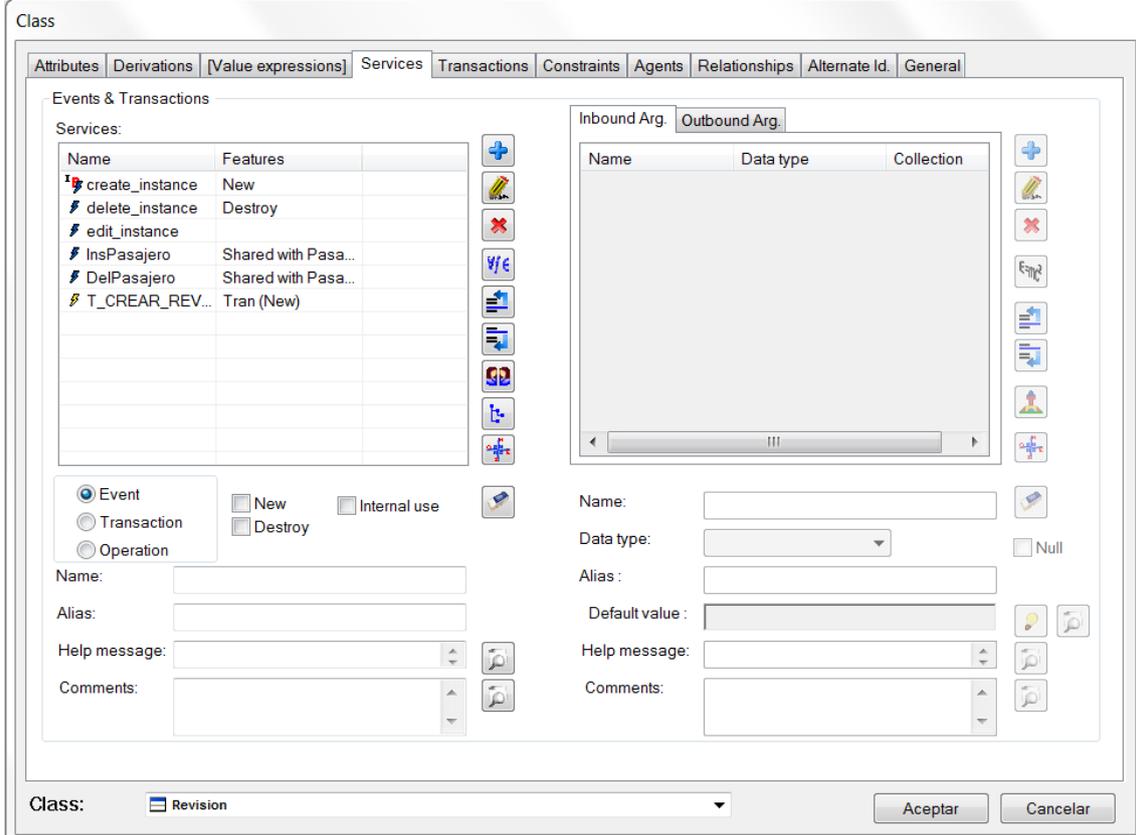


Ilustración 20 I-Servicios Revisión

Salvo por el último servicio, se repite el mismo proceso que se ha producido con las anteriores clases, ninguno de sus servicios se ha tenido que crear, Integranova lo ha realizado automáticamente.

## 4.6. ¿Cómo abordar la posible automatización completa del proceso de generación?

Debido a que este proceso de generación de código no es totalmente automático. Se debe proceder a terminar el modelo de forma manual.

### 4.6.1. Relaciones entre clases

En primer lugar, no se ha podido determinar la relación entre cada una de las clases. Por ello, se ha procedido a insertarlo de forma manual. El analista ha procedido a establecer las relaciones de cardinalidad entre las diferentes clases, en el modelado a mano, basándose en el modelo conceptual detallado en el apartado anterior.

Las relaciones entre las clases son estáticas a no ser que se indique lo contrario.

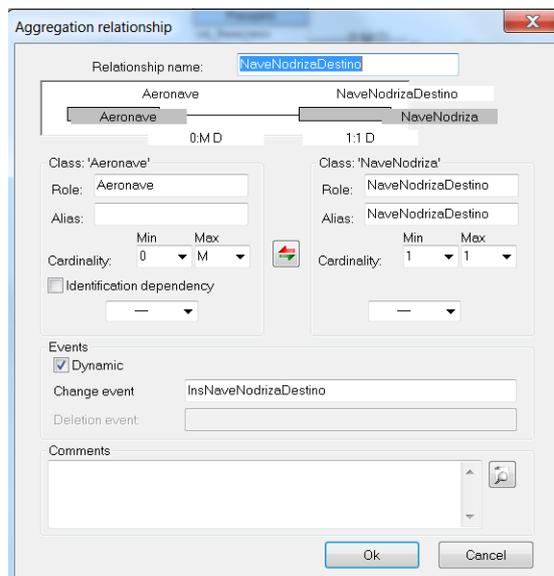


Ilustración 21 I-Relación Nave Nodrizas Destino

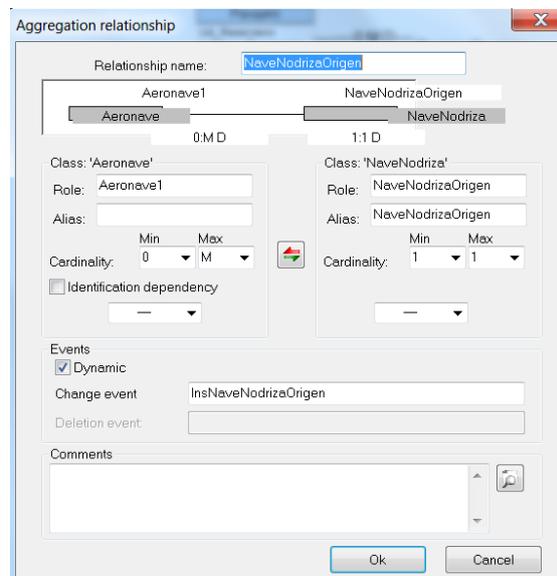


Ilustración 22 I-Relación Nave Nodrizas Origen

Estas son las relaciones entre Nave Nodrizas y Aeronave, el analista ha decidido la cardinalidad en el modelo conceptual, muchas Aeronaves pueden ser agregadas a una única Nave Nodrizas.

Como se puede observar ambas relaciones son dinámicas, las cuales han creado de forma automática un evento con su nombre, indicados anteriormente. Dichos eventos se utilizarán para insertar el origen y el destino actuales de la Aeronave.

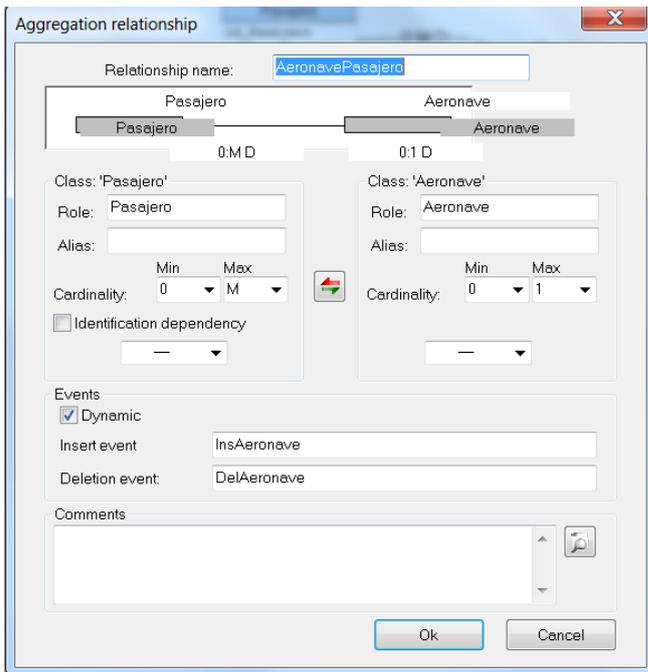


Ilustración 23 I-Relación Aeronave Pasajero

Como se puede observar la relación es dinámica, creando de forma automática un evento con su nombre, indicado anteriormente. Dicho evento se utilizará para subir y bajar al pasajero de la Aeronave.

Estas es la relación entre Aeronave y Pasajero, muchos Pasajeros pueden circular en una única Aeronave.

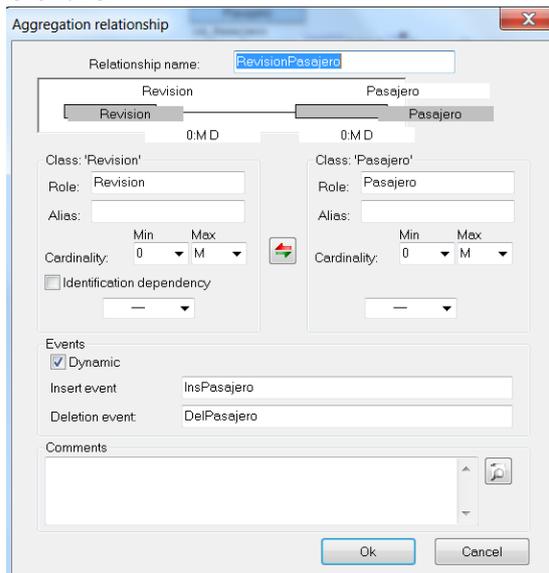


Ilustración 24 I-Relación Revisión Pasajero

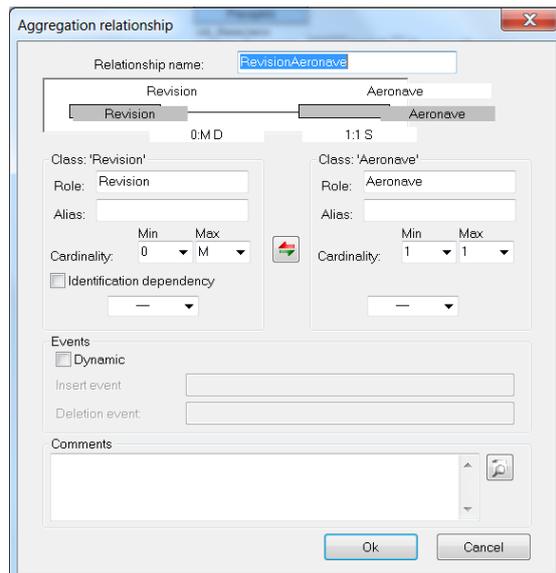


Ilustración 25 I-Relación Revisión Aeronave

Estas son las relaciones entre Revisión y Pasajero, muchos pasajeros pueden estar asignados a diferentes revisiones. Además, la relación es dinámica, los eventos creados se utilizarán para listar a los pasajeros durante la revisión.

La relación de Revisión con Aeronave, cada Aeronave puede tener muchas revisiones.

### 4.6.2. Diagrama resultante

A continuación, se muestra la imagen del diagrama resultante.

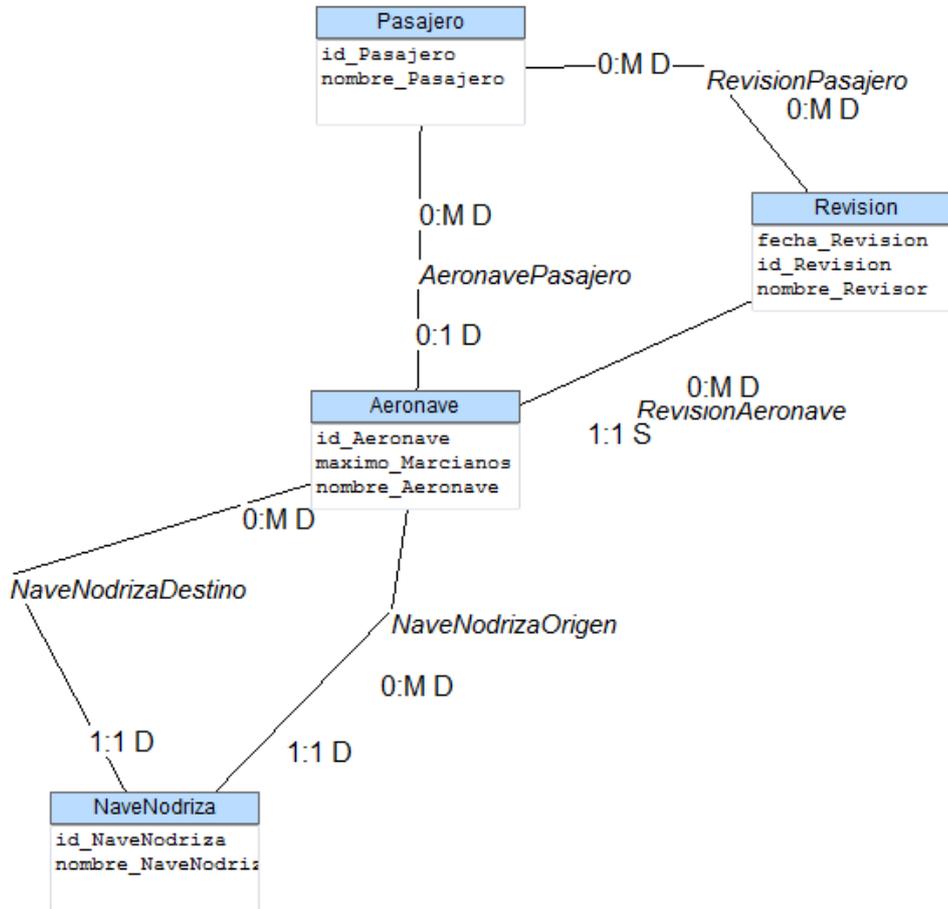


Ilustración 26 I-Diagrama de clases C1

Tras las inserciones de las distintas relaciones entre las clases y su cardinalidad, se observa el diagrama resultante de caso de estudio.

### 4.6.3. Transacciones

A la hora de crear la revisión de las distintas aeronaves, se ha procedido a crear una transacción que cumpla dicho cometido.

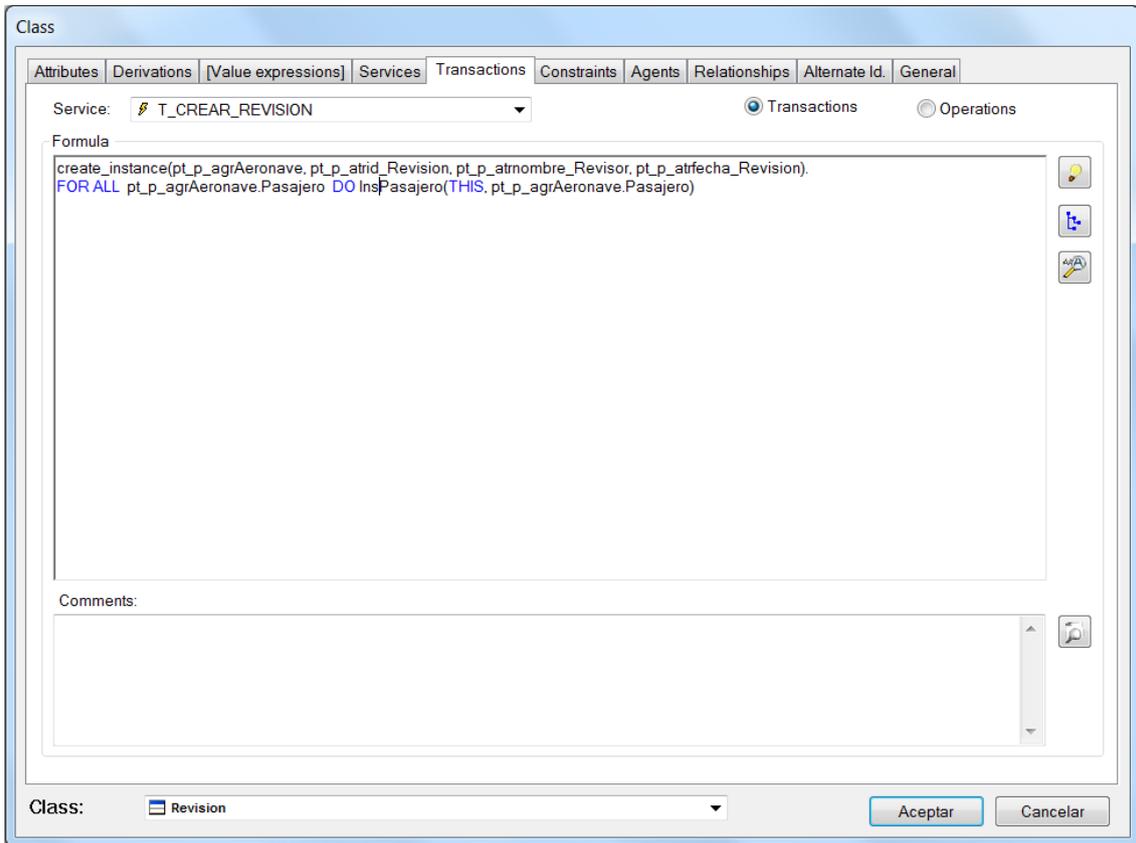


Ilustración 27 I-Transacción Revisión

Como se puede observar, la creación de las transacciones no es de forma automática. El analista debe proceder, mediante la inserción de una función en el modelado de Integranova, para obtener el resultado ideado para el evento de la revisión de las aeronaves.

## 5. Caso de Estudio 2

---

### 5.1. Motivación de este caso

Este documento presenta una demostración, que ejemplifica la derivación manual de un método OO Conceptual, tomando como entrada un modelo de requisitos de análisis de la comunicación.

La demostración de laboratorio corresponde a la gestión de todos los videoclubs de una cadena llamada APPV.

Esta demostración, ilustra la técnica de derivación, demuestra que la puesta en práctica de dicha técnica.

## 5.2. Descripción contextual del caso

### 5.2.1. Funciones del sistema

#### **Crear nuevo Videoclub**

Se crea un Videoclub, perteneciente a la cadena de Videoclubs.

#### **Crear nueva película**

Cuando una película se crea, se debe asignar a un videoclub. Puede haber películas que se llamen igual, pero estén en distintos videoclubs. Esto son objetos distintos de la clase película, que tienen el mismo nombre, pero distinto código.

#### **Registrar socio**

Cuando un socio se registra en el videoclub, debe proporcionar todos sus datos personales. Esto le da derecho a alquilar películas en cualquier videoclub de la cadena (no sólo en el que se haya registrado).

#### **Alquilar película**

Cuando un socio alquila películas, se debe registrar en el sistema la fecha en la cual se ha llevado las películas y la fecha cuando las debe devolver. En un alquiler puede haber más de una película incluida. Todas las películas que forman parte de un alquiler tendrán la misma fecha de recogida y de devolución. La fecha de devolución se puede editar y modificar, pero sólo se puede poner una fecha posterior a la actual. El precio del alquiler se calcula como un derivado, sumando los precios de las películas que componen el alquiler.

#### **Realizar estudio estadístico**

Se desea mantener un estudio estadístico para ver lo que se gasta en la cadena de videoclubs cada uno de los socios. Para ello, una vez al mes, el administrador introduce el socio para el que desea calcular las estadísticas y el mes. Con estos datos, el sistema calcula el total de euros gastados por ese socio en ese mes. En un mes, no se puede almacenar más de un dato estadístico por socio.

#### **Crear Administrador**

Se desea que por cada Videoclub un administrador pertenezca al sistema para poder operar con el programa.

## 5.2.2. Características de los usuarios

El sistema será utilizado por los administradores de los Videoclubs de la cadena APPV.

## 5.2.3. Requisitos Específicos

### Crear nuevo Videoclub

#### Entrada

- Código del videoclub (autonumérico)
- Nombre del gerente (cadena, obligatorio y editable)
- Ciudad (cadena, obligatorio y editable)
- Calle (cadena, obligatorio y editable)
- Código postal (cadena 5 caracteres, obligatorio y editable)

#### Salida

Los datos del Videoclub quedan registrados en el sistema.

### Crear nueva película

#### Entrada

- Código de la película (autonumérico)
- Nombre (cadena, obligatorio y editable)
- Director (cadena, opcional y editable)
- Fecha de estreno (fecha, obligatorio y editable)
- Precio de alquiler (real, obligatorio y editable)

#### Proceso

Puede haber películas que se llamen igual, pero estén en distintos videoclubs.

#### Salida

Los datos de la película quedan registrados en el sistema.

### Registrar socio

#### Entrada

- Código del socio (autonumérico)
- Nombre (cadena, obligatorio y editable)
- Edad (entero, obligatorio y editable)

#### Salida

Los datos del socio quedan registrados en el sistema.

## Alquilar película

### Entrada

- Fecha de alquiler
- Nombre (cadena, obligatorio y editable)
- Edad (entero, obligatorio y editable)

### Salida

El alquiler del socio queda registrado en el sistema.

## Realizar estudio estadístico

### Entrada

- Código del socio
- Mes de la estadística

### Proceso

En un mes, no se puede almacenar más de un dato estadístico por socio.

### Salida

Con estos datos, el sistema calcula el total de euros gastados por ese socio en ese mes.

## Crear Administrador

### Entrada

- Código del administrador (autonumérico)
- Nombre (cadena, obligatorio y editable)

### Salida

Los datos del administrador quedan registrados en el sistema.

## 5.2.4. Formularios de negocio

### Crear nuevo Videoclub

Crear nuevo videoclub	
Código del videoclub	V0000001
Nombre del gerente	Pape García
Ciudad	Valencia
Calle	Av. Germanías
Código Postal	46007

Campo	Descripción
Código del videoclub	Código único para diferenciar los videoclubs entre si
Nombre del gerente	Nombre del gerente del videoclub
Ciudad	Ciudad a la que pertenece el videoclub
Calle	Calle a la que pertenece el videoclub
Código Postal	Código postal correspondiente al videoclub

### Crear nueva película

Crear nueva película	
Código de la película	P0000001
Nombre	Mad max
Director	George Miller
Fecha de estreno	05/02/1980
Precio de alquiler	2.00 €

Campo	Descripción
Código de la película	Código único para diferenciar las películas entre si
Nombre	Nombre de la película
Director	Director de la película
Fecha de estreno	Fecha de estreno de la película
Precio de alquiler	Precio del alquiler de la película

## Registrar socio

Registrar socio	
Código del socio	S0000001
Nombre	Paco Pérez
Edad	25

Campo	Descripción
Código del socio	Código único para diferenciar los socios entre si
Nombre	Nombre del socio
Edad	Edad del socio

## Alquilar película

Alquilar película	
Código de alquiler	A0000001
Fecha de recogida	06/08/2016
Fecha de devolución	10/08/2016
Total a pagar	5.50€

Campo	Descripción
Código de alquiler	Código único para diferenciar los alquileres entre si
Fecha de recogida	Fecha de recogida del alquiler
Fecha de devolución	Fecha de devolución del alquiler
Total a pagar	Importe total del alquiler

## Realizar estudio estadístico

Realizar estudio estadístico	
Código de la estadística	E0000001
Fecha de creación	05/05/2017
Total gastado	25.00€

Campo	Descripción
Código de la estadística	Código único para diferenciar las estadísticas entre si
Fecha de creación	Fecha de creación de la estadística
Total gastado	Total gastado en el mes referente a la estadística

## Crear Administrador

Crear administrador	
Código del administrador	AD0000001
Nombre	Pepe Domingo

Campo	Descripción
Código del administrador	Código único para diferenciar los administradores entre si
Nombre	Nombre del administrador

## 5.3. Análisis comunicativo de negocio

### 5.3.1. Introducción

Se pretende automatizar la gestión de una cadena de videoclubs.

El trabajo que actualmente se realiza a mano ha sido dividido en procesos de negocio. La siguiente tabla introduce dichos procesos y los empleados que participan en ellos.

Proceso	Subproceso	Empleado encargado
Crear nuevo videoclub		Empleado
Crear nueva película		Empleado
Registrar socio		Empleado
Alquilar película		Empleado
Realizar estudio estadístico		Empleado
Crear administrador		Administrador del sistema

### 5.3.2. Diagramas de comunicación

#### Crear nuevo videoclub

El proceso de creación del videoclub, el empleado del sistema introducirá los datos del videoclub, tras ello estará creado en el sistema.

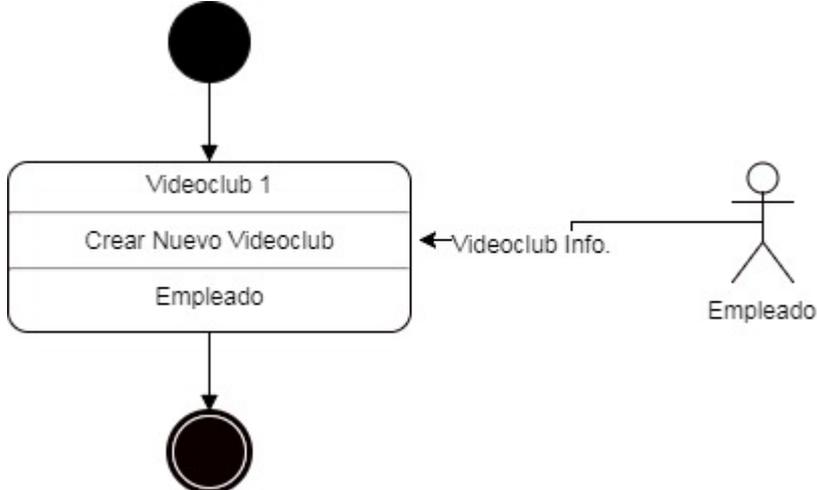


Ilustración 28 Crear nuevo videoclub

#### Crear nueva película

El proceso de creación de la nueva película, es idéntico al del videoclub.

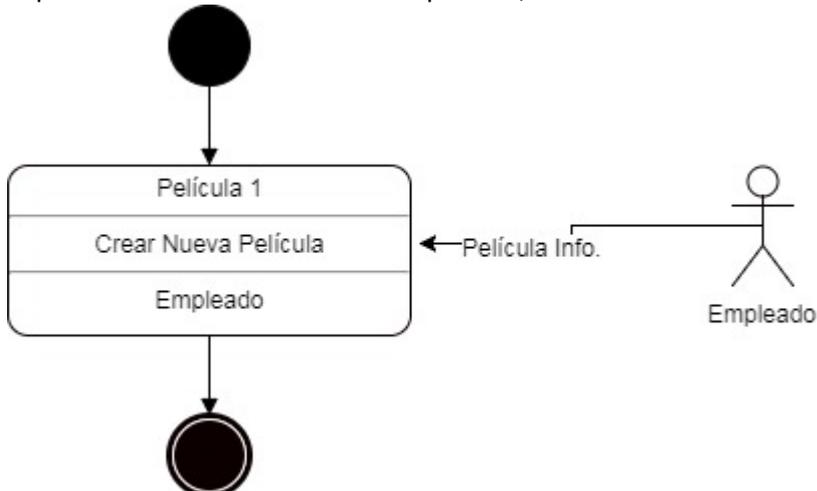


Ilustración 29 Crear nueva película

## Registrar socio

El proceso de registrar socio, es idéntico a los dos procesos de creación anteriores.

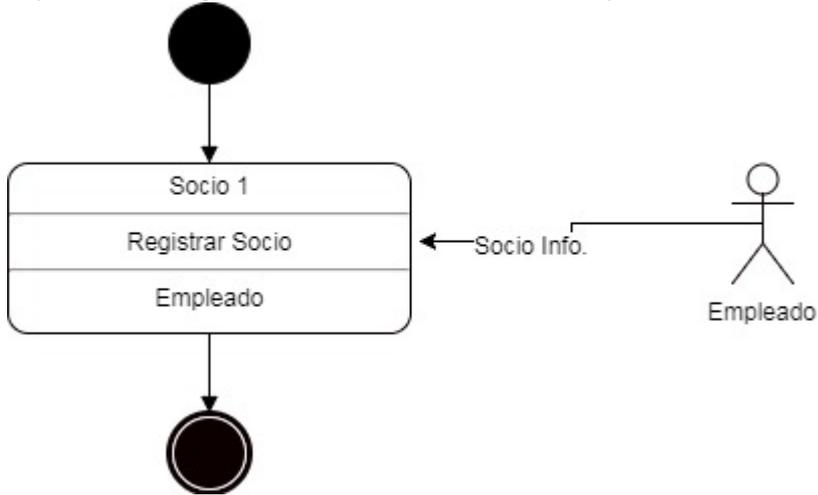


Ilustración 30 Registrar socio

## Alquilar película

El proceso de alquiler de película difiere de los anteriores en la necesidad de comprobar si el socio está registrado en el sistema, para poder proceder a la asignación de la película a su persona, además en cualquier momento antes de la devolución de la película se podrá modificar la fecha de devolución de la misma. Los dos procesos que se observan en la imagen tienen la misma función que los anteriores, la inserción de la información del socio y la asignación de la película al socio.

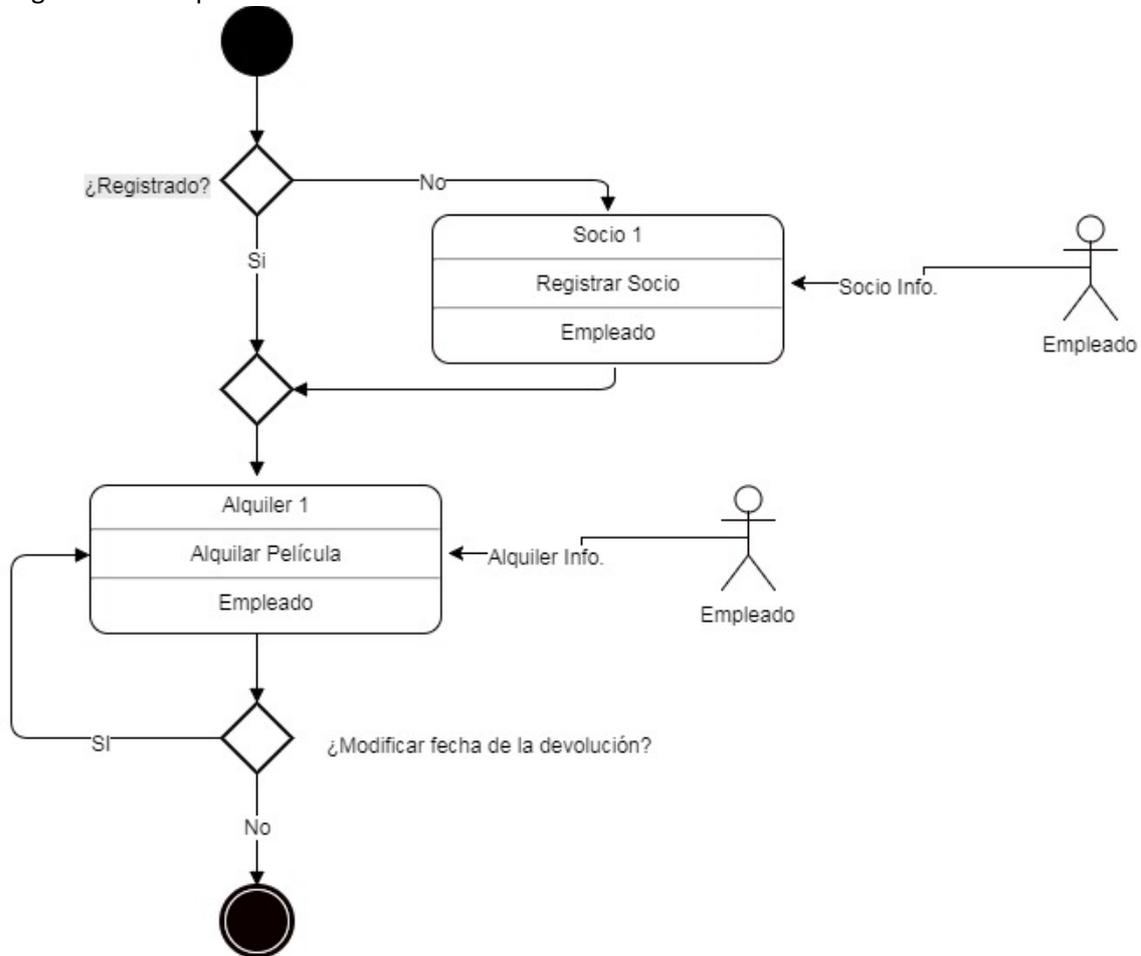


Ilustración 31 Alquilar película

## Realizar estudio estadístico

Este proceso, tiene como función principal la obtención por parte del sistema de un listado de los gastos realizados por el socio durante el mes indicado, pero existe la precondition de que solamente se puede realizar un estudio al mes.

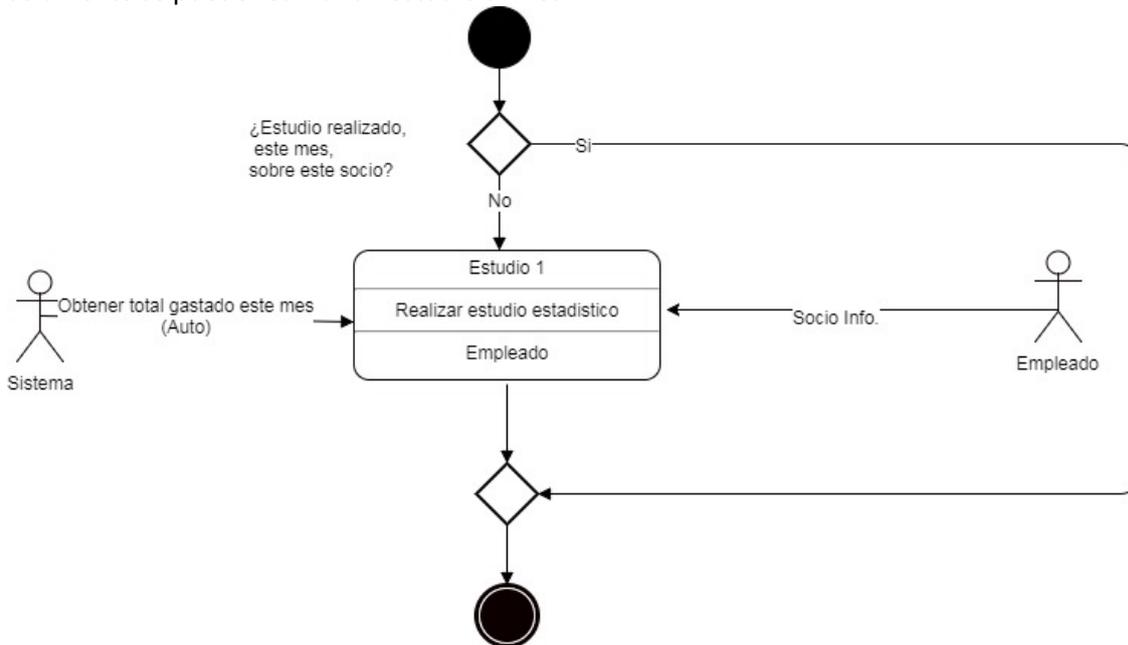


Ilustración 32 Realizar estudio estadístico

## Crear administrador

El proceso de creación del administrador, es idéntico a los procesos de creación anteriores.

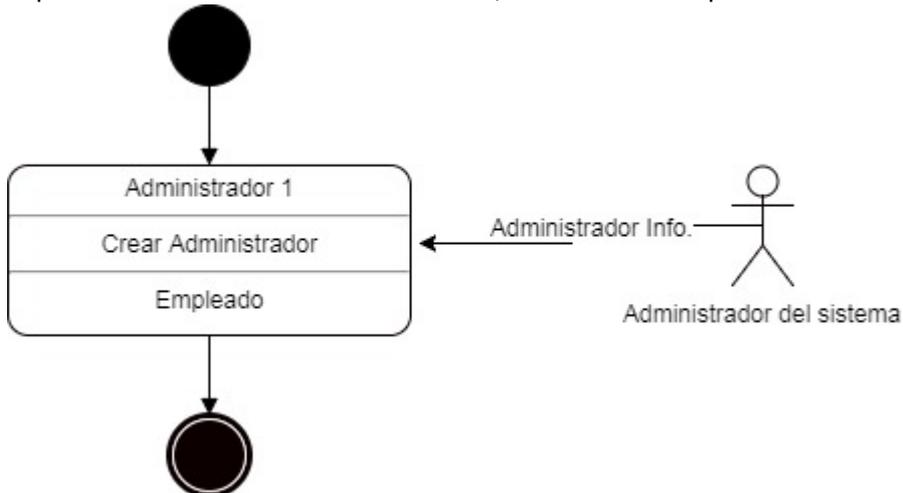


Ilustración 33 Crear Administrador

### 5.3.3. Descripción de eventos

#### Crear nuevo videoclub

##### *Información general*

Esta funcionalidad almacena los datos de los videoclubs existentes. De cada nave nodriza sólo se desea almacenar el nombre, un identificador y los datos referentes a la localización del local del videoclub.

##### *Requisitos generales*

###### **Actores responsables:**

- Empleado

###### **Formulario**

Crear nuevo videoclub	
Código del videoclub	V0000001
Nombre del gerente	Pape García
Ciudad	Valencia
Calle	Av.\ Germanías
Código Postal	46007

##### *Requisitos comunicativos*

Campo	Operación	Dominio	Ejemplo
VIDEOCLUB =			
< Código V. +		Text	V0000001
Nombre +		Text	Pape García
Ciudad +		Text	Valencia
Calle +		Text	Av.\ Germanías
C.P.		Num	46007
>			

Campo	Descripción
Código del videoclub	Código único para diferenciar los videoclubs entre si
Nombre del gerente	Nombre del gerente del videoclub
Ciudad	Ciudad a la que pertenece el videoclub
Calle	Calle a la que pertenece el videoclub
Código Postal	Código postal correspondiente al videoclub

###### **Restricciones estructurales**

El código del videoclub ha de ser único

###### **Restricciones contextuales**

Los videoclubs son identificados por su código.

*Reacción a los requisitos*

**Tratamiento**

El videoclub queda creado.

## Crear nueva película

### Información general

Esta funcionalidad pretende almacenar todas las películas existentes. En el sistema se almacena una ficha de cada película, asignada a un videoclub.

### Requisitos generales

#### Actores responsables:

- Empleado

#### Formulario

Crear nueva película	
Código de la película	P0000001
Nombre	Mad max
Director	George Miller
Fecha de estreno	05/02/1980
Precio de alquiler	2.00 €

### Requisitos comunicativos

Campo	Operación	Dominio	Ejemplo
PELICULA =			
< Código P. +	I	Text	P0000001
Nombre +	I	Text	Mad max
Director +	I	Text	George Miller
Fecha de estreno +	I	Date	05/02/1980
Precio de alquiler	G	Text	2.00 €
>			

Campo	Descripción
Código de la película	Código único para diferenciar las películas entre si
Nombre	Nombre de la película
Director	Director de la película
Fecha de estreno	Fecha de estreno de la película
Precio de alquiler	Precio del alquiler de la película

#### Restricciones estructurales

El código de la película ha de ser único.

Los códigos de las películas van relacionadas a la copia de la película física. Es decir, por cada película, un código.

#### Restricciones contextuales

Las películas son identificadas por su código.

### Reacción a los requisitos

#### Tratamiento

La película queda creada.

## Registrar socio

### *Información general*

El sistema debe proporcionar la funcionalidad para registrar los socios que alquilan en el videoclub. Una vez que un socio se registra en uno de los videoclubs de la cadena, puede alquilar en cualquier videoclub de la cadena sin volver a registrarse.

### *Requisitos generales*

#### **Actores responsables:**

- Empleado

#### **Formulario**

Registrar socio	
Código del socio	S0000001
Nombre	Paco Pérez
Edad	25

### *Requisitos comunicativos*

Campo	Operación	Dominio	Ejemplo
SOCIO =			
< Código S.+		Text	S0000001
Nombre +		Text	Paco Pérez
Edad		Num	25
>			

Campo	Descripción
Código del socio	Código único para diferenciar los socios entre si
Nombre	Nombre del socio
Edad	Edad del socio

#### **Restricciones contextuales**

El código del socio ha de ser único.

### *Reacción a los requisitos*

#### **Tratamiento**

El socio queda creado.

## Alquilar película

### Información general

Para registrar que un socio realiza un alquiler de películas, basta con indicar en el sistema en la fecha de recogida y la fecha de devolución.

### Requisitos generales

#### Actores responsables:

- Empleado

#### Formulario

Alquilar película	
Código de alquiler	A0000001
Fecha de recogida	06/08/2016
Fecha de devolución	10/08/2016
Total a pagar	5.50€

### Requisitos comunicativos

Campo	Operación	Dominio	Ejemplo
A_PELICULA = < Código A_P.+	I	Text	A0000001
Fecha R. +	I	Date	06/08/2016
Fecha D. +	I	Date	10/08/2016
Total	G	Text	5.50€
>			

Campo	Descripción
Código de alquiler	Código único para diferenciar los alquileres entre si
Fecha de recogida	Fecha de recogida del alquiler
Fecha de devolución	Fecha de devolución del alquiler
Total a pagar	Importe total del alquiler

#### Restricciones contextuales

El código del alquiler ha de ser único.

El alquiler se puede modificar la fecha de devolución, pero siempre será una fecha posterior a la fecha indicada actualmente en el sistema.

### Reacción a los requisitos

#### Tratamiento

El alquiler queda creado.

## Realizar estudio estadístico

### Información general

Se desea mantener un estudio estadístico para ver lo que se gasta en la cadena de videoclubs cada uno de los socios. Para ello, una vez al mes, el administrador introduce el socio para el que desea calcular las estadísticas y el mes. Con estos datos, el sistema calcula el total de euros gastados por ese socio en ese mes. En un mes, no se puede almacenar más de un dato estadístico por socio.

### Requisitos generales

#### Actores responsables:

- Empleado

#### Formulario

Realizar estudio estadístico	
Código de la estadística	E0000001
Fecha de creación	05/05/2017
Total gastado	25.00€

### Requisitos comunicativos

Campo	Operación	Dominio	Ejemplo
ESTUDIO =			
< Código E. +	I	Text	E0000001
Fecha +	I	Date	05/05/2017
Total	G	Text	25.00€
>			

Campo	Descripción
Código de la estadística	Código único para diferenciar las estadísticas entre si
Fecha de creación	Fecha de creación de la estadística
Total gastado	Total gastado en el mes referente a la estadística

#### Restricciones contextuales

El código del estudio ha de ser único.

No se puede realizar más de un estudio sobre el mismo socio en el mismo mes.

### Reacción a los requisitos

#### Tratamiento

El estudio queda insertado en el sistema.

## Crear administrador

### Información general

El sistema debe proporcionar la funcionalidad para registrar los administradores de los videoclubs.

### Requisitos generales

#### Actores responsables:

- Administrador del sistema

#### Formulario

Crear administrador	
Código del administrador	AD0000001
Nombre	Pepe Domingo

### Requisitos comunicativos

Campo	Operación	Dominio	Ejemplo
ADMINISTRADOR = < Código AD. + Nombre >	I I	Text Text	AD0000001 Pepe Domingo

Campo	Descripción
Código del administrador	Código único para diferenciar los administradores entre si
Nombre	Nombre del administrador

#### Restricciones contextuales

El código del administrador ha de ser único.

### Reacción a los requisitos

#### Tratamiento

El administrador queda insertado en el sistema.

## 5.4. Modelo conceptual aplicado a la metodología-OO

### 5.4.1. Derivación del modelado de objetos

Antes de aplicar las directrices de transformación, el diagrama de eventos comunicativos debe ser pre-procesado. Para ello se adoptan las siguientes acciones:

El diagrama de eventos comunicativos se extiende con eventos comunicativos de otros procesos.

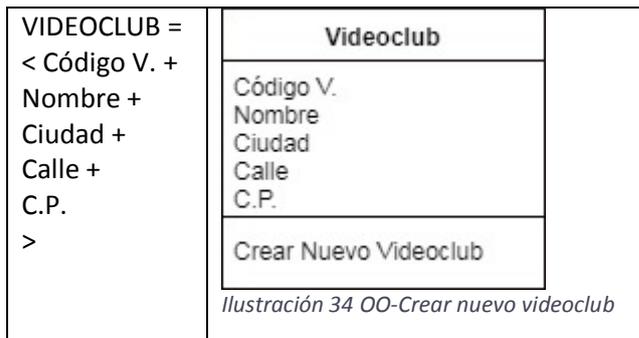
El diagrama de evento comunicativo extendido se elimina las relaciones de precedencia de bucle back (como, así como otros elementos) hasta que se obtiene un conjunto ordenado de eventos comunicativos.

El árbol dirigido raíz resultante es procesado para obtener una lista ordenada de eventos. Los sucesos se procesan en el orden definido, aplicando las directrices de derivación.

#### Crear nuevo videoclub

La estructura de mensajes del evento comunicativo Videoclub, consiste en una Agregación llamada VIDEOCLUB.

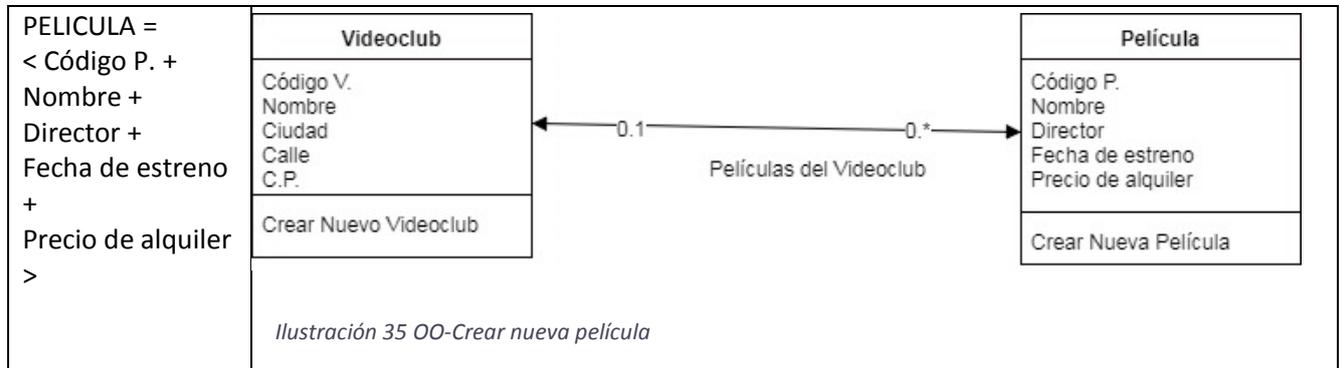
Especifica los datos del Videoclub correspondiente. Además, se agrega un servicio de creación denominado Crear nuevo videoclub a la clase; Este servicio tiene tantos argumentos entrantes como atributos de clase se han derivado.



## Crear nueva película

La estructura de mensajes del evento comunicativo Película, consiste en una Agregación llamada PELICULA.

Especifica los datos de la película correspondiente. Además, se agrega un servicio de creación denominado Crear nueva película a la clase; Este servicio tiene tantos argumentos entrantes como atributos de clase se han derivado.

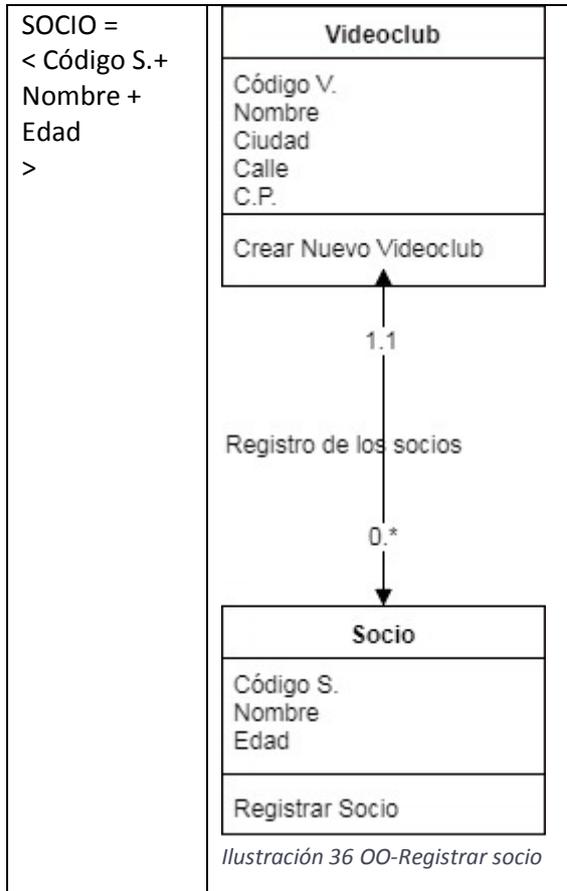


Se establece una relación entre Videoclub y Película correspondiente, para indicar a que Videoclub pertenece la película.

## Registrar Socio

La estructura de mensajes del evento comunicativo Socio, consiste en una Agregación llamada SOCIO.

Especifica los datos del socio correspondiente. Además, se agrega un servicio de creación denominado Registrar socio a la clase; Este servicio tiene tantos argumentos entrantes como atributos de clase se han derivado.

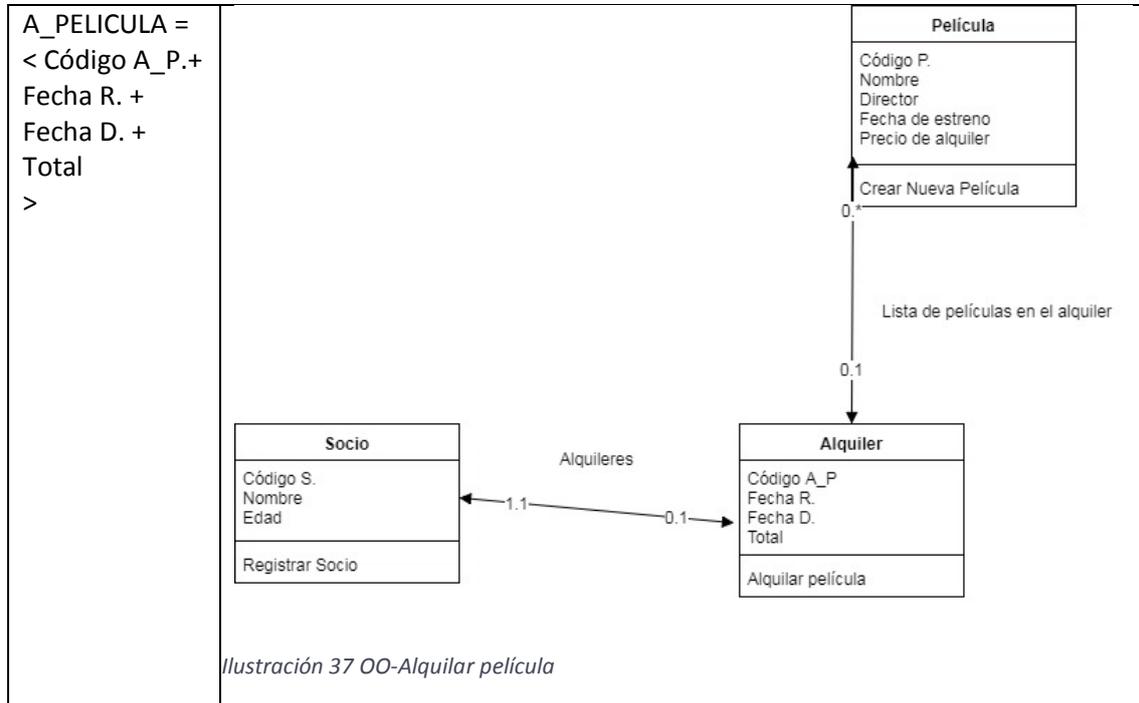


Se establece una relación entre Videoclub y Socio correspondiente, para indicar en que Videoclub se han registrado los socios.

## Alquilar película

La estructura de mensajes del evento comunicativo Alquiler, consiste en una Agregación llamada A\_PELICULA.

Especifica los datos del alquiler correspondiente. Además, se agrega un servicio de creación denominado Alquilar película a la clase; Este servicio tiene tantos argumentos entrantes como atributos de clase se han derivado.

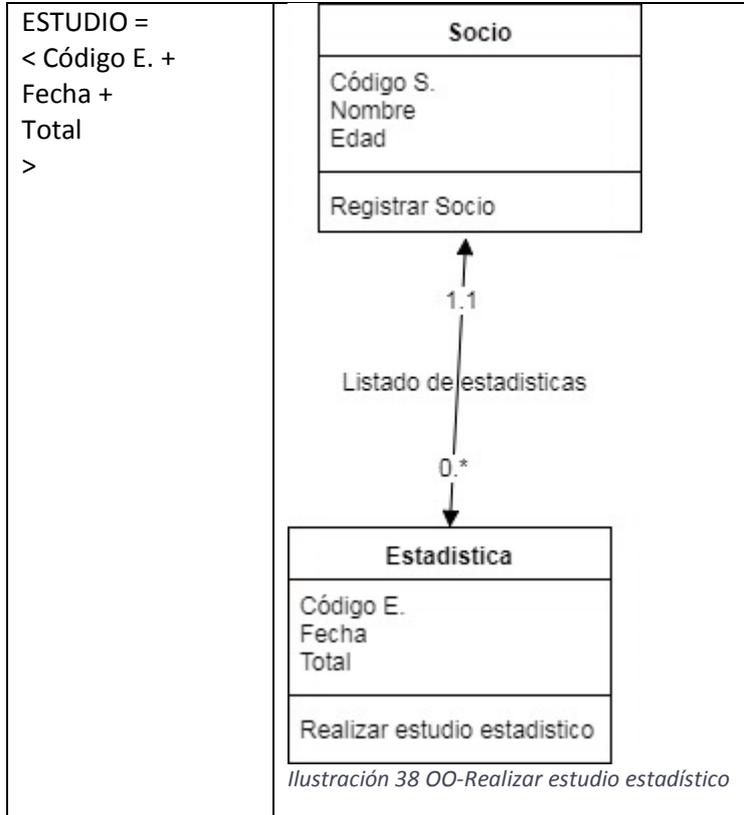


Se establece una relación entre Alquiler y Socio correspondiente, para indicar que Socio ha realizado el alquiler. Además, se establece una relación entre Alquiler y Película para indicar que películas se han alquilado en el proceso de alquiler.

## Realizar estudio estadístico

La estructura de mensajes del evento comunicativo Estadística, consiste en una Agregación llamada ESTUDIO.

Especifica los datos del socio correspondiente. Además, se agrega un servicio de creación denominado Realizar estudio estadístico a la clase; Este servicio tiene tantos argumentos entrantes como atributos de clase se han derivado.

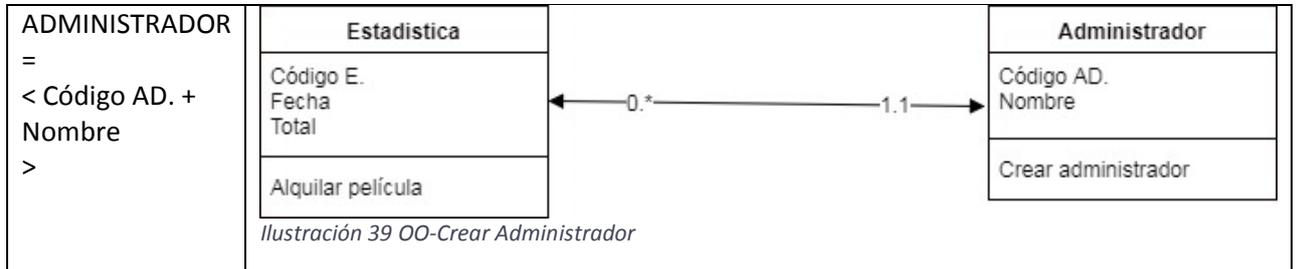


Se establece una relación entre Estadística y Socio correspondiente, para indicar sobre que Socio se ha realizado la estadística.

## Crear Administrador

La estructura de mensajes del evento comunicativo Administrador, consiste en una Agregación llamada ADMINISTRADOR.

Especifica los datos del socio correspondiente. Además, se agrega un servicio de creación denominado Crear administrador a la clase; Este servicio tiene tantos argumentos entrantes como atributos de clase se han derivado.



Se establece una relación entre Estadística y Administrador correspondiente, para indicar las estadísticas que crea el administrador.

## 5.4.2. Diagrama de clases

Teniendo en cuenta todo lo explicado en el punto anterior, aquí se muestra el diagrama de clases resultante.

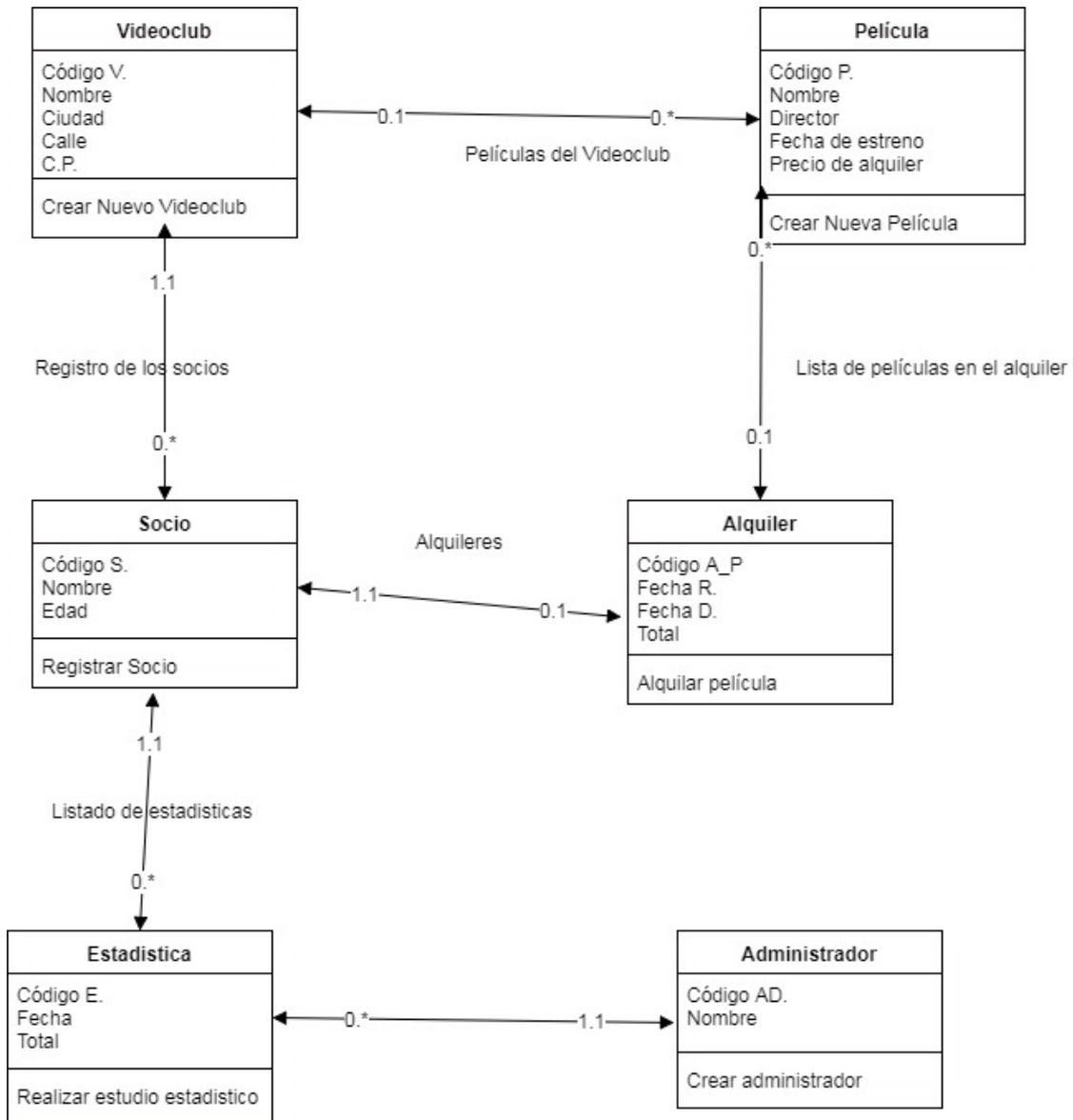


Ilustración 40 OO-Diagrama de clases C2

## 5.5. Construcción del modelo conceptual en Integranova

En la siguiente serie de imágenes se ilustra, la parte automatizable del modelo de clases creado utilizando a la herramienta Integranova

### Crear nuevo videoclub

La siguiente imagen muestra la clase Videoclub, junto a sus atributos.

Name	Attribute type	Data type	Id	Size	Default v...	Request...	Nulls
id_VideoClub	Constant	Autonumeric	Y...			Yes	No
Gerente	Variable	Text	No			Yes	No
Ciudad	Variable	Text	No			Yes	No
Calle	Variable	Text	No			Yes	No
Codigo_Postal	Variable	Text	No			Yes	No

Ilustración 41 I-Atributos Videoclub

Como se puede observar se han creado cinco atributos:

El identificador del videoclub, el cual es un atributo constante y del tipo autonumérico.

El nombre del gerente del videoclub, atributo variable de tipo Text.

La ciudad, atributo variable de tipo Text.

La calle, atributo variable de tipo Text.

El código postal, atributo variable de tipo Text.

Dichos atributos han tenido que ser insertados a mano, por lo cual se ha tenido que proceder al uso de un analista para que lo añadiese.

En la siguiente imagen se muestran los servicios de la propia clase, los tres son creados por defecto por Integranova.

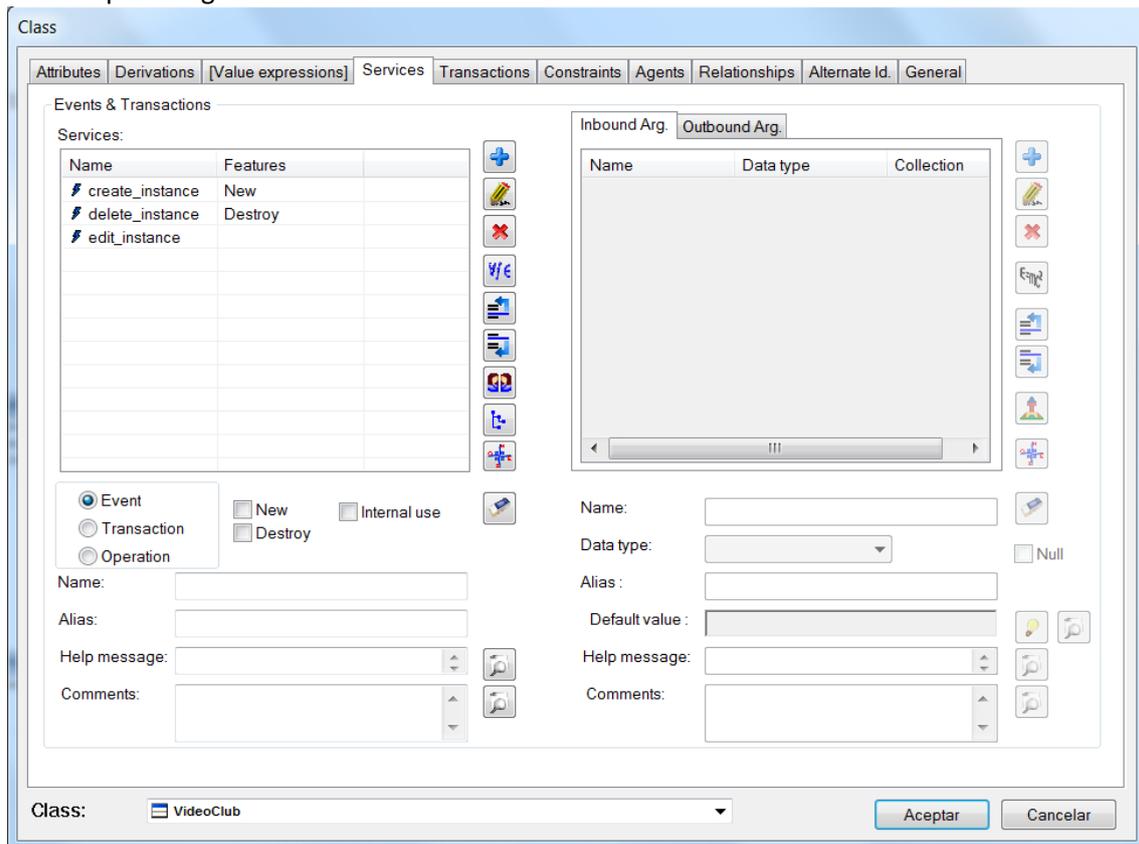


Ilustración 421-Servicios Videoclub

Gracias a Integranova los procesos de creación, edición y destrucción, se han creado al crear la clase en el modelador. Debido a ello, no es necesario insertarlos a mano.

## Crear nueva película

La siguiente imagen muestra la clase Película, junto a sus atributos.

Class

Attributes | Derivations | [Value expressions] | Services | Transactions | Constraints | Agents | Relationships | Alternate Id. | General

Attributes

Name	Attribute type	Data type	Id	Size	Default v...	Request...	Nulls
id_Pelicula	Constant	Autonumeric	Y...			Yes	No
Nombre	Variable	Text	No			Yes	No
Director	Variable	Text	No			Yes	Yes
Fecha_Estreno	Variable	Date	No			Yes	No
Precio_Alquiler	Variable	Real	No			Yes	No

Name:  Attribute type:

Alias:  Data type:

Help Message

Comments

Class: Película

Aceptar Cancelar

Ilustración 43 I-Atributos Película

En la imagen se reflejan los distintos atributos creados a mano por el analista.

En la siguiente imagen se muestran los servicios de la propia clase. Como sucede con la clase Videoclub, los servicios son creados por Integranova.

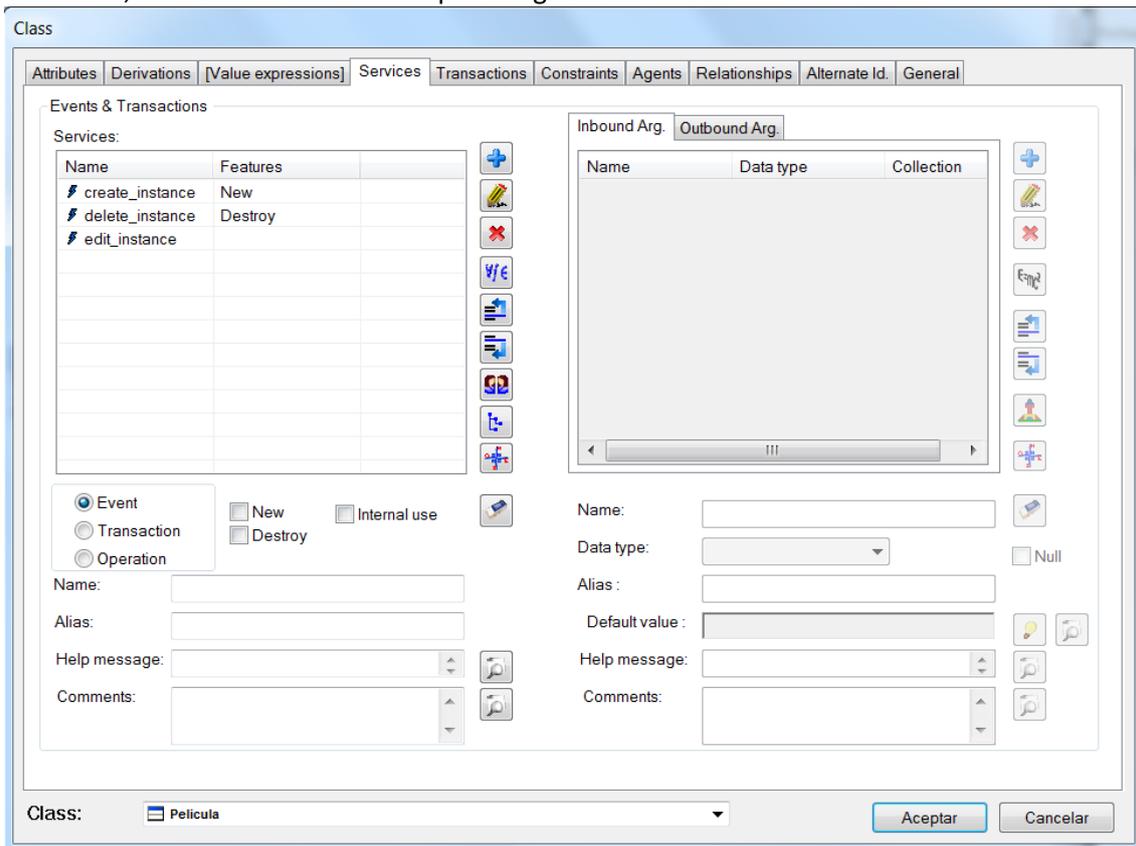


Ilustración 44 I-Servicios Película

Se repite el mismo proceso que se ha producido con Videoclub, ninguno de sus servicios se ha tenido que crear, Integranova lo ha realizado automáticamente.

## Registrar socio

La siguiente imagen muestra la clase Socio, junto a sus atributos.

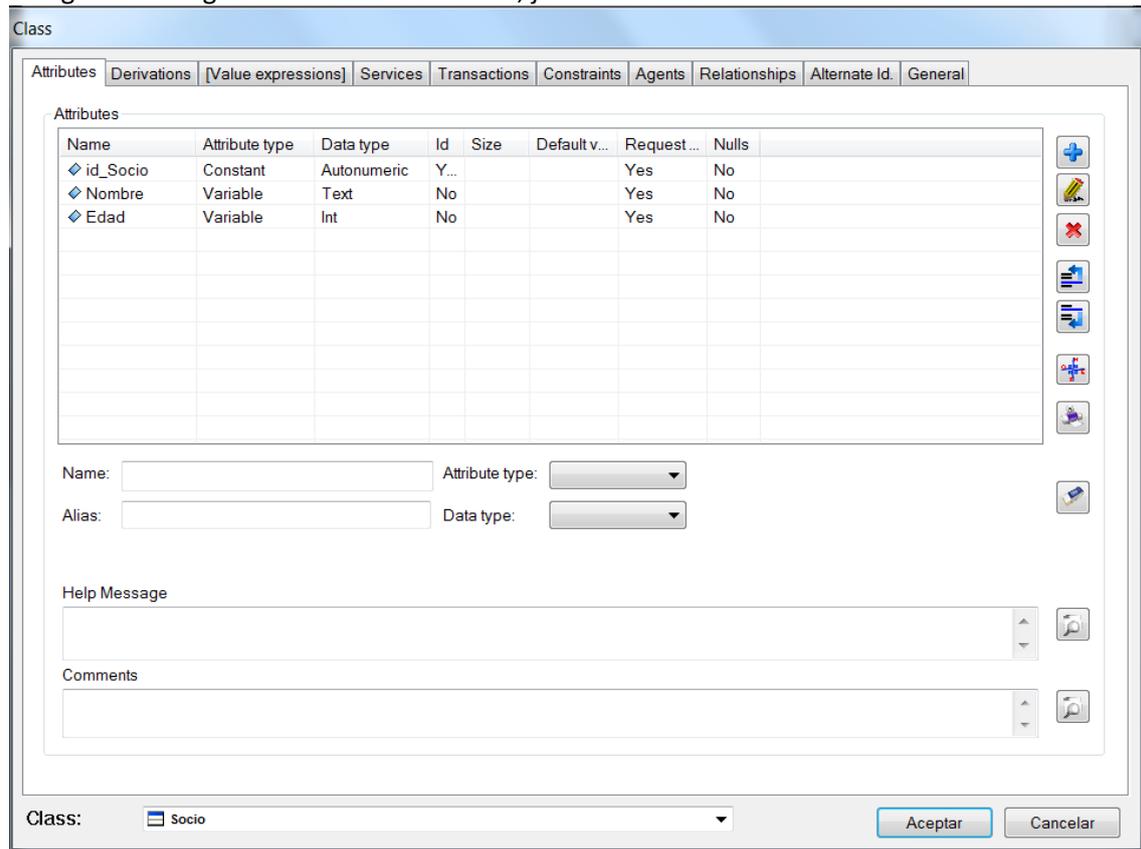


Ilustración 45 I-Atributos Socio

En la imagen se reflejan los distintos atributos creados a mano por el analista.

En la siguiente imagen se muestran los servicios de la propia clase. Como sucede con la clase Videoclub, los servicios son creados por Integranova.

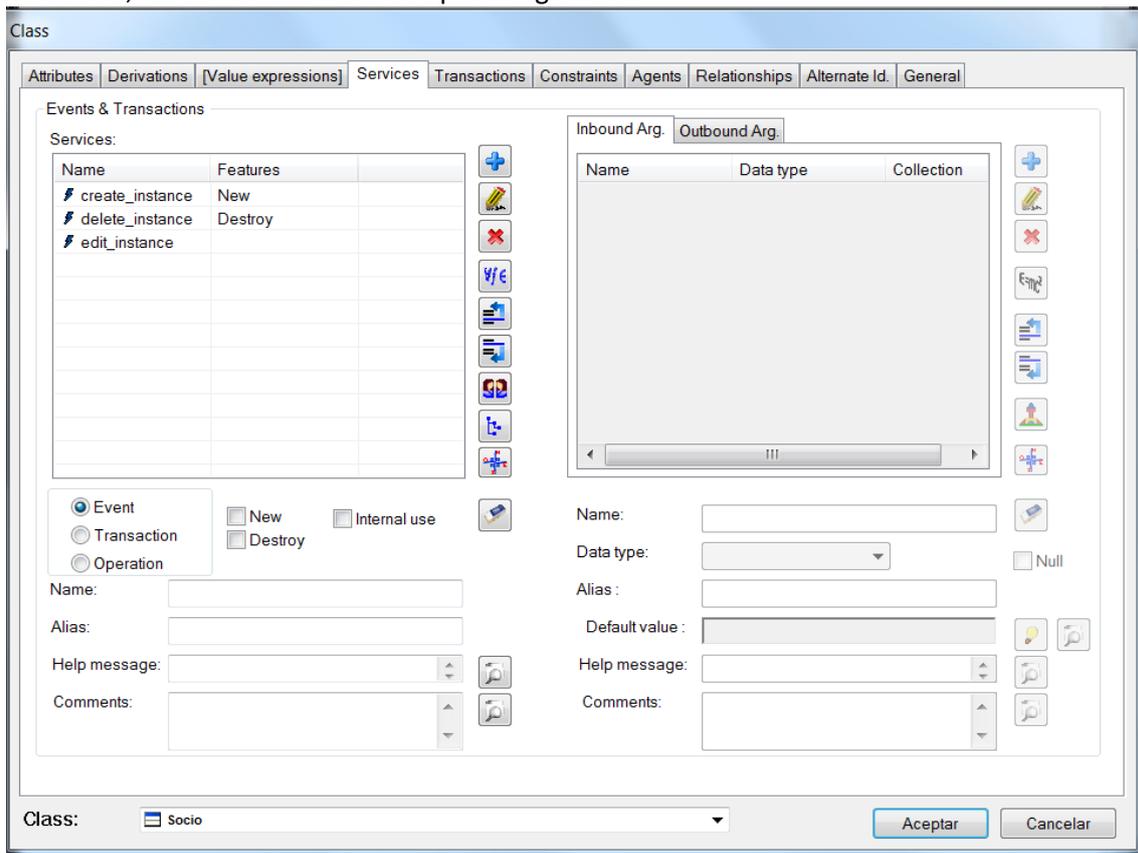


Ilustración 46 I-Servicios Socio

Se repite el mismo proceso que se ha producido con Videoclub, ninguno de sus servicios se ha tenido que crear, Integranova lo ha realizado automáticamente.

## Alquilar película

La siguiente imagen muestra la clase Alquiler, junto a sus atributos.

Name	Attribute type	Data type	Id	Size	Default v...	Request ...	Nulls
id_Alquiler	Constant	Autonumeric	Y...			Yes	No
Fecha_recog...	Constant	Date	No			Yes	No
Fecha_devol...	Variable	Date	No			Yes	No
Total_Pagar	Derived	Real	No				

Ilustración 47 I-Atributos Alquilar Película

En la imagen se reflejan los distintos atributos creados a mano por el analista.

En este caso el campo Total\_Pagar es derivado, el cual se tendrá que insertar su fórmula de obtención de los datos.

En la siguiente imagen se muestran los servicios de la propia clase. Como sucede con la clase Videoclub, los servicios son creados por Integranova.

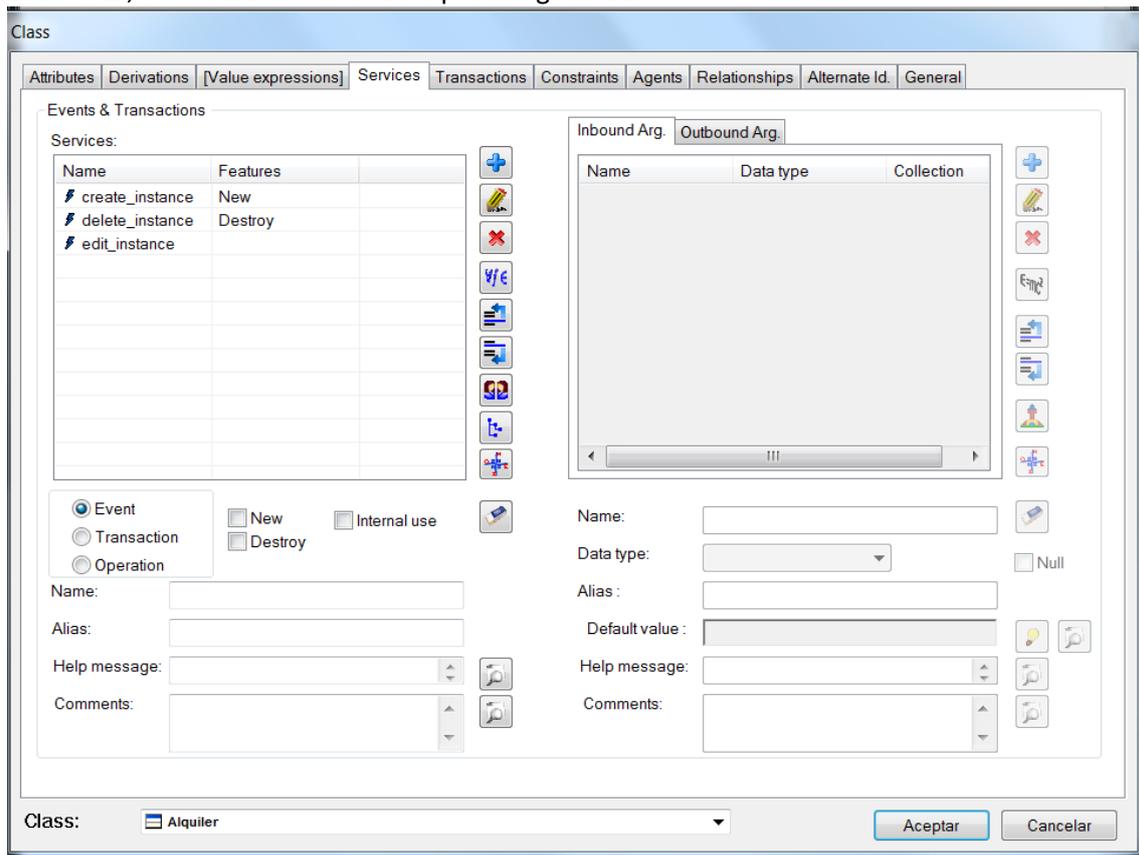


Ilustración 48 I-Servicios Alquiler Película

Se repite el mismo proceso que se ha producido con Videoclub, ninguno de sus servicios se ha tenido que crear, Integranova lo ha realizado automáticamente.

## Realizar estudio estadístico

La siguiente imagen muestra la clase Estadística, junto a sus atributos.

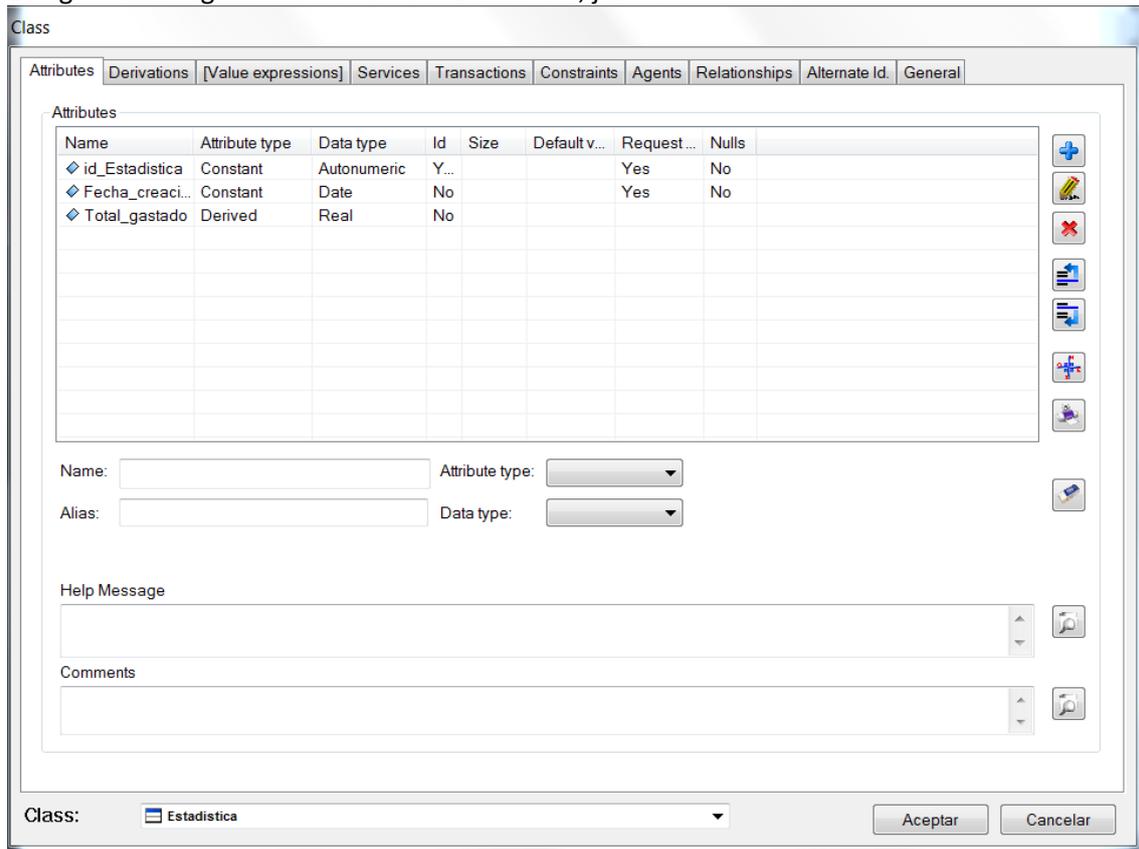


Ilustración 49 I-Atributos Realizar estudio estadístico

En la imagen se reflejan los distintos atributos creados a mano por el analista.

En este caso el campo Total\_gastado es derivado, el cual se tendrá que insertar su fórmula de obtención de los datos.

En la siguiente imagen se muestran los servicios de la propia clase. Como sucede con la clase Videoclub, los servicios son creados por Integranova.

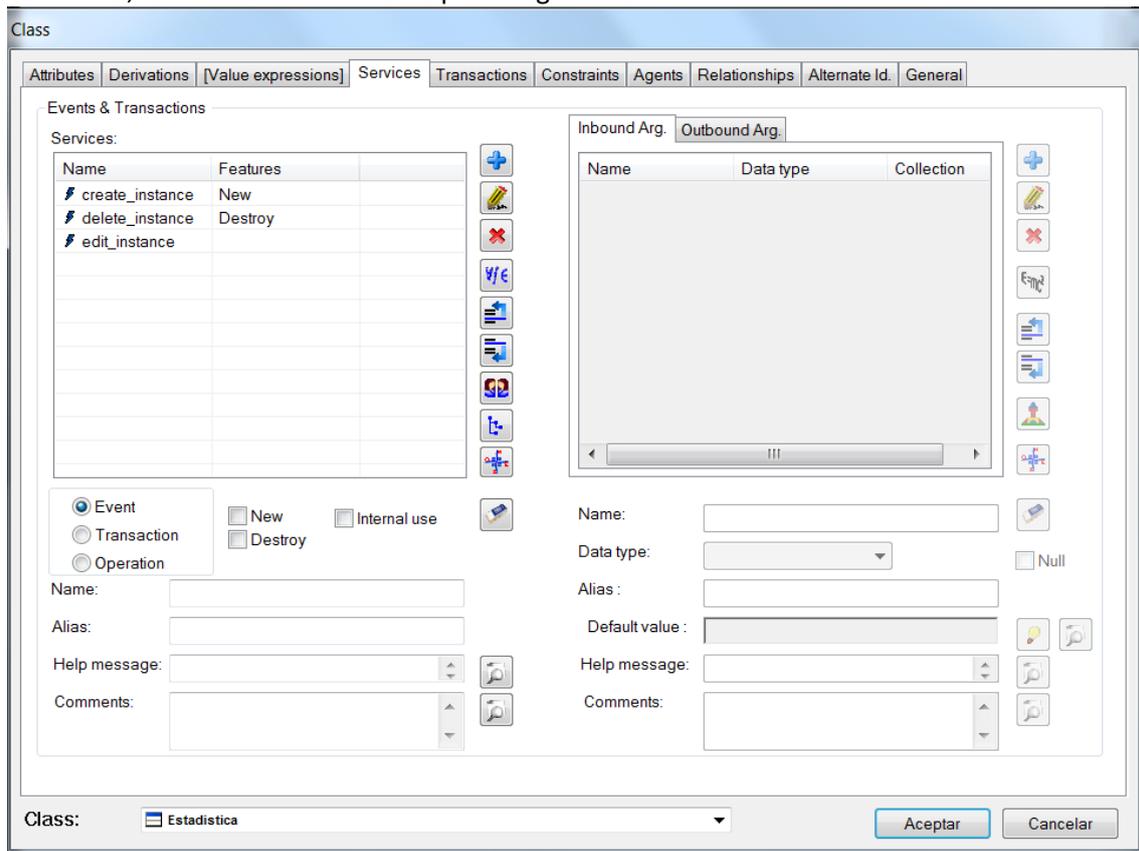


Ilustración 50 I-Servicio Realizar estudio estadístico

Se repite el mismo proceso que se ha producido con las anteriores clases, ninguno de sus servicios se ha tenido que crear, Integranova lo ha realizado automáticamente.

## Crear administrador

La siguiente imagen muestra la clase Socio, junto a sus atributos.

Name	Attribute type	Data type	Id	Size	Default v...	Request ...	Nulls
◆ id_Administra...	Constant	Autonumeric	Y...			Yes	No
◆ Nombre	Variable	Text	No			Yes	No

Name:  Attribute type:

Alias:  Data type:

Help Message

Comments

Class:

Ilustración 51 I-Atributos Administrador

En la imagen se reflejan los distintos atributos creados a mano por el analista.

En la siguiente imagen se muestran los servicios de la propia clase. Como sucede con la clase Videoclub, los servicios son creados por Integranova.

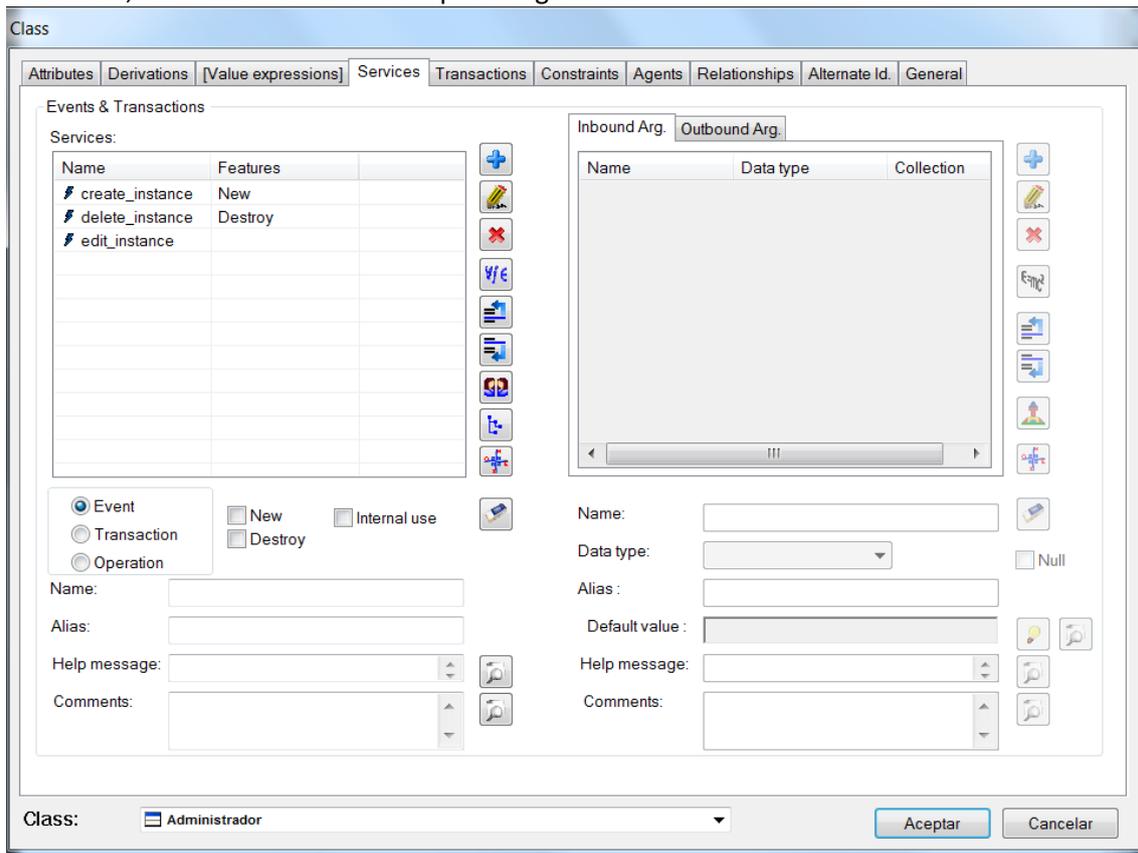


Ilustración 52 I-Servicios Administrador

Se repite el mismo proceso que se ha producido con las anteriores clases, ninguno de sus servicios se ha tenido que crear, Integranova lo ha realizado automáticamente.

## 5.6. ¿Cómo abordar la posible automatización completa del proceso de generación?

Debido a que este proceso de generación de código no es totalmente automático. Se debe proceder a terminar el modelo de forma manual.

### 5.6.1. Relaciones entre clases

En primer lugar, no se ha podido determinar la relación entre cada una de las clases. Por ello, se ha procedido a insertarlo de forma manual. El analista ha procedido a establecer las relaciones de cardinalidad entre las diferentes clases, en el modelado a mano, basándose en el modelo conceptual detallado en el apartado anterior.

Las relaciones entre las clases son estáticas a no ser que se indique lo contrario.

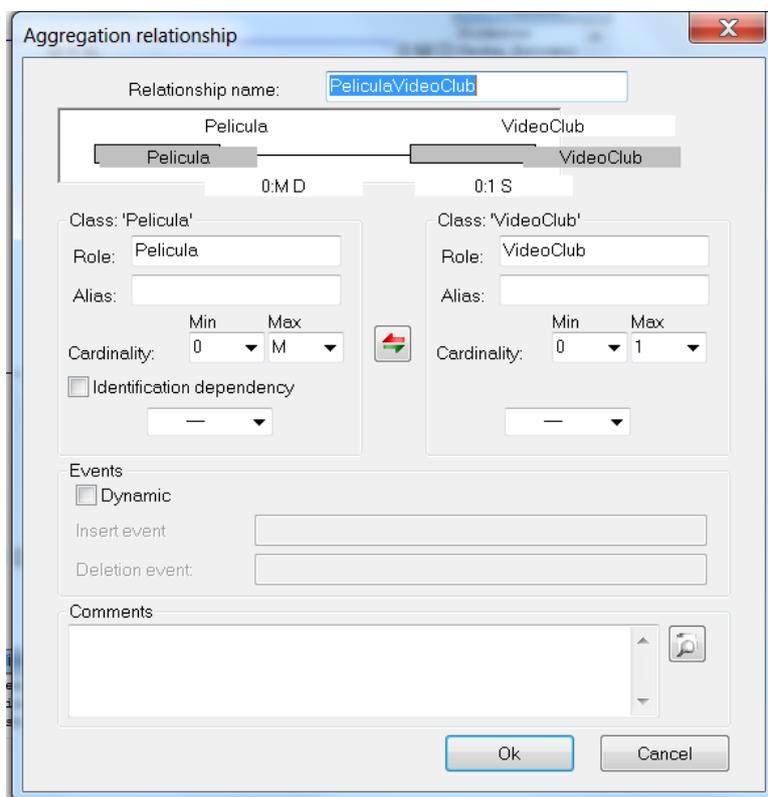


Ilustración 53 I-Relación Película Videoclub

Estas son las relaciones entre Película y Videoclub, el analista ha decidido la cardinalidad en el modelo conceptual, muchas Películas pueden ser agregadas a un único Videoclub.

Esta es la relación entre Videoclub y Socio, muchos socios pueden estar asignados a un único Videoclub

Ilustración 54 I-Relación Socio Videoclub

Además, se muestra las relaciones entre Alquiler y Socio, muchos alquileres pueden ser realizados por un socio

Ilustración 55 I-Relación Socio Alquiler

Ilustración 56 I-Relación Película Alquiler

Alquiler y Película, muchas películas solamente pueden estar asignadas a un alquiler activo.

Esta es la relación entre Estadística y Socio, a un socio se le pueden hacer muchas estadísticas.

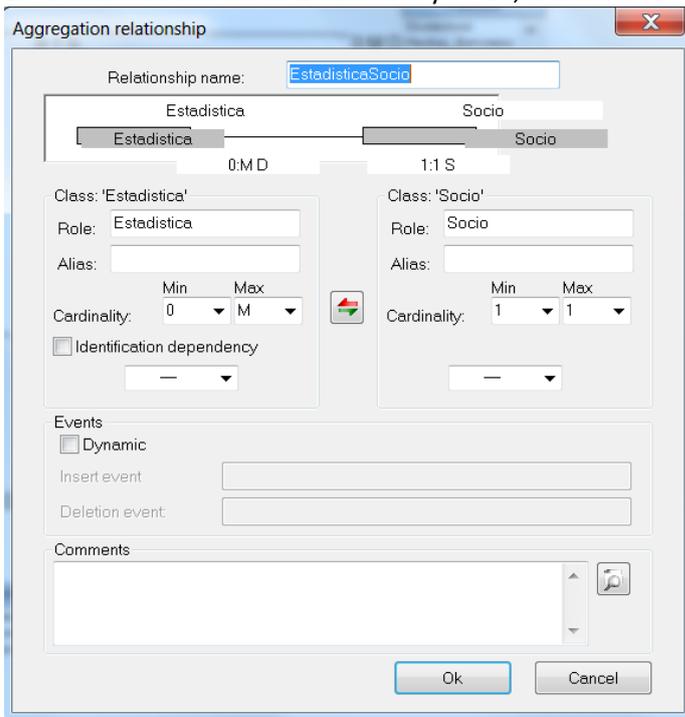


Ilustración 57 I-Relación Estadística Socio

Por último, se muestra la relación entre Estadística y Administrador, un administrador puede hacer muchas estadísticas.

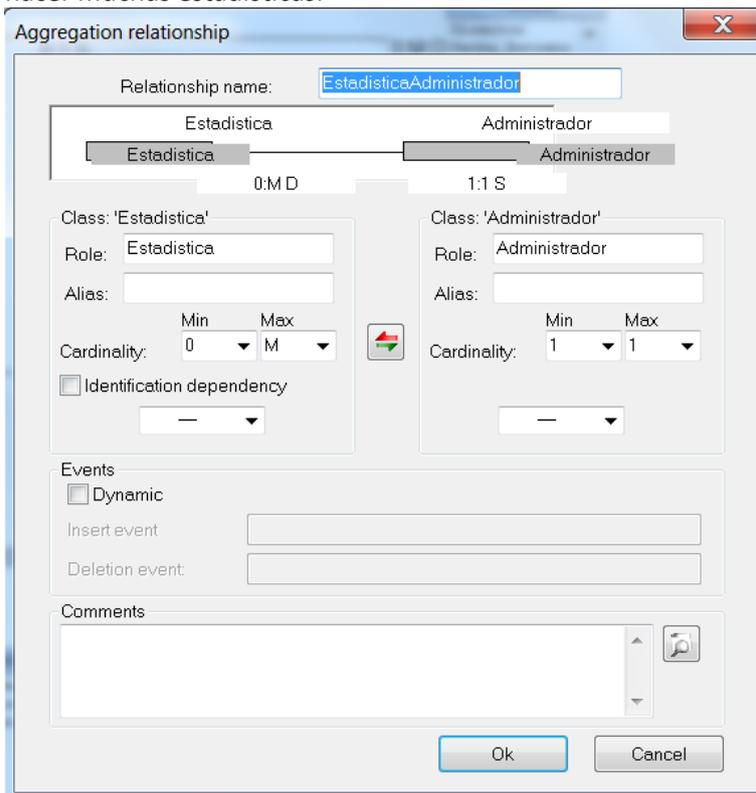


Ilustración 58 I-Relación Estadística Administrador

## 5.6.2. Diagrama resultante

A continuación, se muestra la imagen del diagrama resultante.

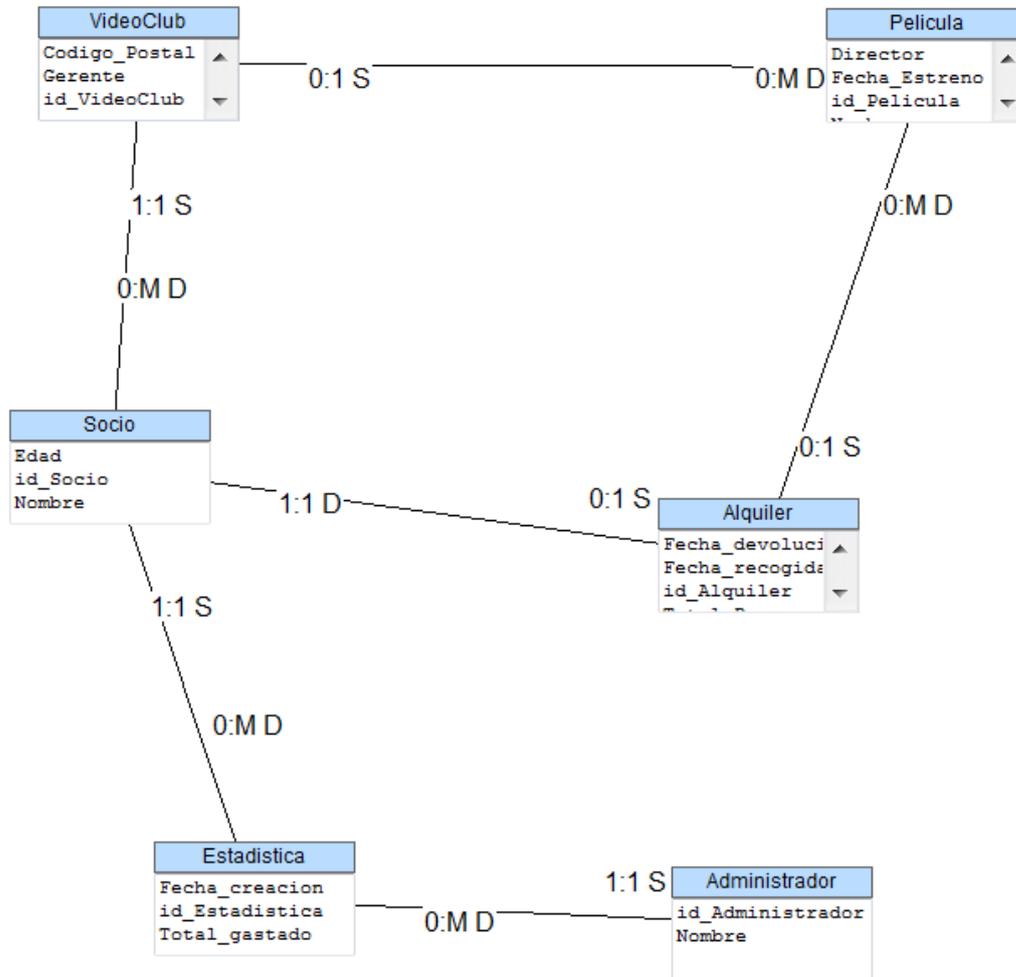


Ilustración 59 I-Diagrama de clases C2

Tras las inserciones de las distintas relaciones entre las clases y su cardinalidad, se observa el diagrama resultante de caso de estudio.

### 5.6.3. Atributos derivados

En este caso de estudio, se derivan varios atributos que llevan una dependencia.

Como se puede observar, la creación de las fórmulas para los atributos derivados tampoco es automática. El analista debe proceder, mediante la inserción de una fórmula en el modelado de Integranova, para obtener el resultado ideado para el atributo de Total\_Pagar.

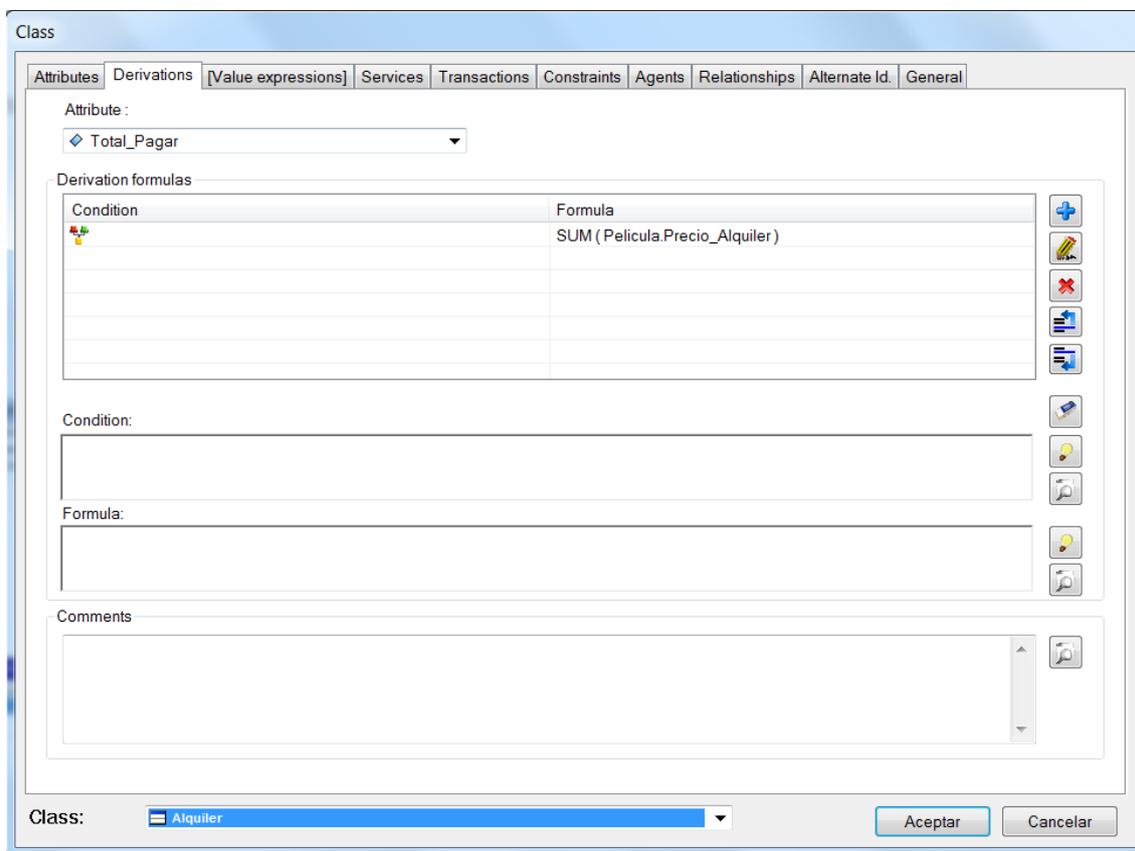


Ilustración 60 I-Derivación Total a Pagar

Aquí, se observa la variable Total\_Pagar que deriva de la suma del precio de alquiler de todas las películas alquiladas en esa instancia de alquiler.

Este segundo atributo, conlleva una mayor complejidad puesto que lleva una condición incluida. Se encarga de calcular el total de dinero gastado por un socio en un determinado mes del año.

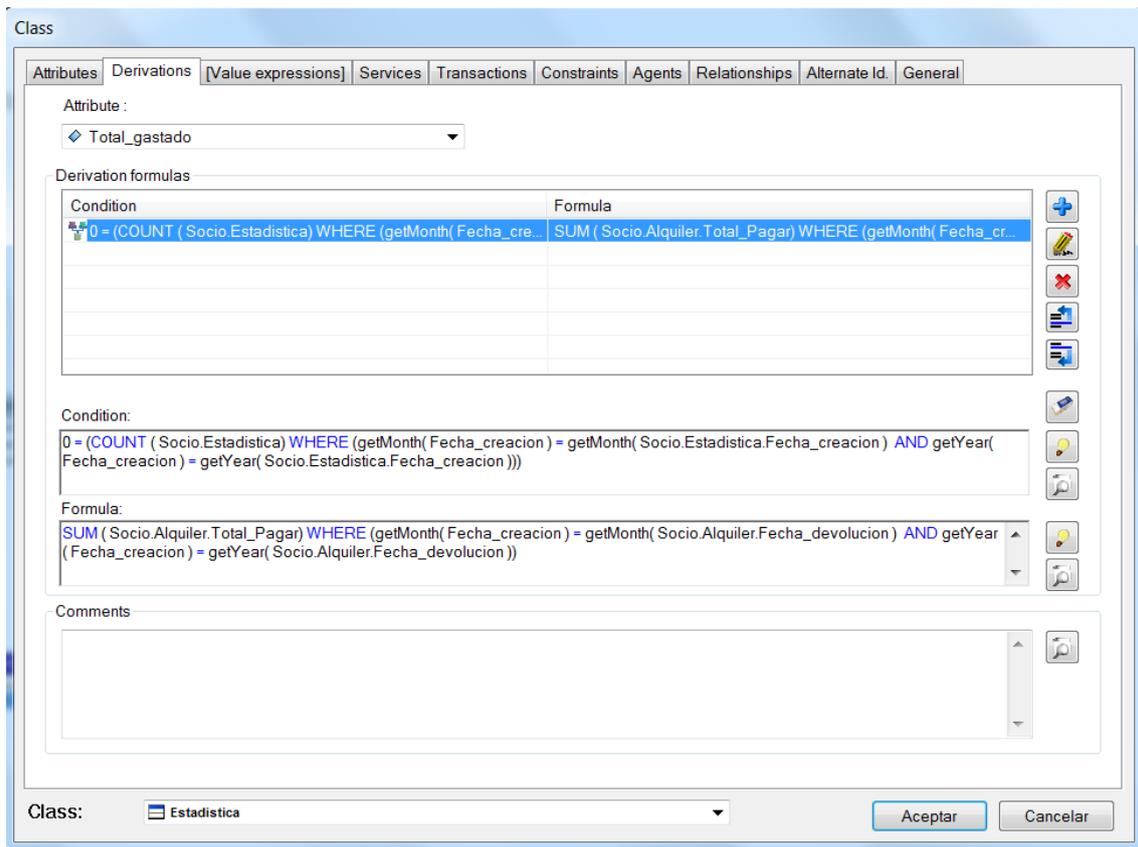


Ilustración 61 I-Derivación Total gastado

## 6. Análisis de automatización del proceso

---

En este punto, se procederá a realizar un análisis del proceso que se ha llevado a cabo para tratar los dos casos de estudio.

### 6.1. Proceso de generación semiautomática

Aplicando este método se han realizado distintas tareas con distintos grados de complejidad.

#### 6.1.1. Descripción contextual del caso

En ambos casos de estudio, se procede un estudio preliminar realizado por un analista, el profesor de ISI. Se ha realizado la adaptación de los requisitos a los formularios de negocio.

Si se tuviese que resolver un problema completo implicaría tener a un analista adaptando las necesidades del cliente, a una descripción comprensible para el equipo de trabajo.

#### 6.1.2. Análisis comunicativo de negocio

Para esta parte del proceso, se ha debido realizar de forma manual por el analista todo el trabajo, siguiendo las pautas del método aplicado. Debido a ello, esta parte no está automatizada, necesitando de un analista que se dedique a proceder con ello.

En primer lugar, se ha procedido a analizar los distintos eventos comunicativos. Obteniendo las interacciones con estos mismos, un único empleado en la mayoría de los casos. Además, se han realizado los diagramas de comunicación que reflejan las interacciones de los eventos con el medio, ya sea por parte de un empleado o la salida producida por el evento en sí. Finalmente, se ha procedido a analizar al detalle cada evento por separado, indicando los requisitos de estos, a nivel general y comunicativo, incluyendo la reacción del sistema a los requisitos del evento.

### **6.1.3. Modelo conceptual aplicado a la metodología-OO**

Igual que en la anterior fase, esta necesita de un analista que trabaje de forma manual para adaptar el modelo. Ambos casos de estudio lo han requerido.

Se ha aplicado el modelo a la metodología-OO, se ha extendido el diagrama de eventos comunicativos con eventos comunicativos de otros procesos. Se han eliminado las relaciones de precedencia de bucle back. Finalmente, el resultado es procesado para obtener una lista ordenada de eventos

### **6.1.4. Modelado en Integranova**

Esta ha sido la última fase en la que se ha tenido que aportar de forma manual detalle. Al contrario que las anteriores, esta no ha requerido de un esfuerzo total por parte del analista, puesto que la herramienta de modelado ha facilitado la tarea del modelado del propio. Ambos casos de estudios solo ha sido necesarios añadir poco detalle que la herramienta no proporciona por si sola.

Se han añadido las relaciones entre clases, la cardinalidad entre estas. Además, se han definido los atributos, todo esto partiendo del modelo anterior. Respecto al resto de detalle, incluir las fórmulas para los atributos derivados, añadir las precondiciones en los servicios y añadir las transacciones junto con el código que define su función.

## 6.2. Proceso de generación automática

La gran mayoría del proceso de creación de software ha sido realizada en la parte del análisis.

### 6.2.1. Modelado en Integranova

Durante el modelado de la solución, no se ha tenido que proceder a crear los diferentes servicios básicos que tienen las clases; creación, edición y eliminación.

La herramienta de modelado de Integranova procedía de forma automática a realizar esas tareas, facilitando el proceso del software.

A su vez, la creación del diagrama de clases ha sido mayoritariamente creado por la herramienta, para ambos casos de estudio.

### 6.2.2. Modelo XML

Para el proceso de validación y creación del XML, no ha habido la necesidad de realizar ningún esfuerzo. A su vez, durante la validación del modelo si existiese algún fallo, la propia herramienta lo hubiese indicado.

### 6.2.3. Cliente Integranova Star

Para crear el proyecto definitivo, que aporte la solución deseada, se realiza una serie de pasos para elegir las diferentes características que se quiere incorporar al proyecto, como el lenguaje de programación, la plataforma donde se va a implementar...

Este proceso es totalmente automático, no se requiere que el usuario que trata con el cliente programar nada.

### 6.2.4. Proceso de la generación de código

Finalmente, la herramienta devuelve un proyecto totalmente funcional, sin que el usuario tenga que modificar nada, con el cual ya se puede proceder a trabajar con él, una vez instalado en el ordenador deseado.

# 7. Conclusiones

---

Este apartado está dedicado a las conclusiones obtenidas una vez realizado todo el proceso del software.

## 7.1. Ventajas

Las distintas ventajas que tiene esta metodología respecto al proceso de desarrollo aplicando la metodología tradicional.

### 7.1.1. Precisión de la solución

Gracias que esta metodología requiere de una mayor dedicación al análisis del problema, implica que va a ser más preciso una vez el producto esté terminado.

### 7.1.2. Reducción de los costes de trabajo

Puesto que la herramienta en sí, realiza todo el código no es necesaria la participación de un mayor personal. A su vez, los tiempos de trabajo son menores, puesto que te ahorras todo el tiempo de programación, que es mucho mayor que el tiempo extra que se dedica al proceso de análisis y modelado del problema.

### 7.1.3. Cambios en el sistema

Puesto que la herramienta en sí, produce todo el código, si es necesario cualquier cambio o ampliación de la solución no requiere de un coste tan grande que si se tuviese que programar la aplicación.

## 7.2. Desventajas

Las distintas desventajas son aplicadas en el campo de la programación tradicional donde el trabajo de analista está muy distanciado del de programador.

### 7.2.1. Curva de aprendizaje

Para un analista/programador la curva de aprendizaje e esta metodología de trabajo no es considerablemente alta. Pero partiendo del punto de vista de un programador tradicional, a su vez, tendría que aprender a diseñar los distintos modelos. Con lo cual, si no se parte de una base, se tiene una serie desventaja respecto a la programación tradicional, puesto que no todo el mundo es capaz de reducir su eficiencia a corto plazo para obtener beneficios a largo plazo.

### 7.2.2. Modelo de negocio

La herramienta de Integranova está adaptada para trabajar con sistemas de formulario. Este proceso, de momento, no es trasladable a otros entornos; Smartphone, videojuegos... Puesto que exigen una mayor interacción con el usuario.

## 7.3. Trabajo futuro

Como trabajo futuro, se debería mejorar la herramienta para que pudiese comprender el trabajo previo que se ha realizado antes de trabajar con el modelado de Integranova.

### 7.3.1. Paso del modelo conceptual a Integranova

En un futuro, Integranova podría implementar la lectura de modelo conceptuales, para no repetir trabajo.

#### Lectura del diagrama de clases

A partir del diagrama de clases el analista ha insertado en el modelo atributos, relaciones entre clases, cardinalidad de las relaciones y servicios que tendrá la aplicación.

Todo este detalle, se ha añadido al modelador de Integranova de forma manual, sin la necesidad de transformarlo.

Si este proceso se automatiza, solamente habría que añadir las precondiciones de los servicios, las fórmulas de los atributos derivados y el código de las transacciones.

Así, el proceso de creación de un modelo funcional se simplificaría de ahorrando un mayor tiempo al grupo de trabajo.

## 7.4. Crítica personal

Tras la realización de este TFM, me he encontrado con varios aspectos de trabajo que me gustaría valorar de forma personal.

### 7.4.1. Aspectos positivos

La mayor ventaja que observo es la abstracción del proceso, debido a que la mayoría de la realización del programa se hace en el PIM, si me equivoco en algún aspecto, solamente lo tengo que modificar a ese nivel, y volver a generar el modelo PSM y obtener el programa con el código incluido. Esto implica que el tiempo que se invierte en cambios es increíblemente pequeño una vez detectado el error. Debido a ello, es muy fácil y sencillo, aplicar esta metodología para casi cualquier trabajo.

### 7.4.2. Aspectos negativos

A su vez, la única pega que le veo a este método es la dependencia respecto a la empresa de generación de código. Puesto que ellos te devuelven el proyecto totalmente ejecutable. Si se han equivocado en algo, aplicar ingeniería inversa sobre el proyecto, implicaría un gran coste, además de que, si modificas algo dentro del código generado y pide una nueva generación de código sobre la antigua versión, no respeta el código modificado, sino que te envía un proyecto totalmente nuevo, con lo que te tocaría volver a implementar las modificaciones que hubiesen hechas en la versión anterior.

## 8. Referencias

---

- [1] S. Martínez, L. Lamoth, R. Moreno, and N. Jacho, “Análisis de la Transformación de Modelo CIM a PIM en el Marco de Desarrollo de la Arquitectura Dirigida por Modelos ( MDA ),” vol. 36, no. 3, 2015.
- [2] S. Espana, A. Gonzales, and P. Oscar, “Communication Analysis : a Requirements Engineering,” *Proc. 21st Int. Conf. Adv. Inf. Syst. Eng. CAiSE 2009*, pp. 1–15, 2007.
- [3] S. España, “Communication Analysis : an introduction ( part 2 ) How to apply CommunicaFon Analysis in this course,” no. part 2.
- [4] J. Gómez, C. Cachero, and O. Pastor, “Conceptual modeling of device-independent Web applications,” *IEEE Multimed.*, vol. 8, no. 2, pp. 26–38, 2001.
- [5] C. Chen, X. Chen, and H. Liu, “Data Model Driven Architecture (DMDA),” *Fourth Int. Work. Adv. Comput. Intell.*, pp. 269–273, 2011.
- [6] J. Duarte, M. González, L. Cernuzzi, and O. Pastor, “Evaluation of software development through an MDA tool: A case study [Evaluación del desarrollo de software mediante una herramienta MDA: Un caso de estudio],” *IEEE Lat. Am. Trans.*, vol. 6, no. 3, pp. 252–259, 2008.
- [7] J. Rodriguez, “Ingenieria de Modelos con MDA estudio comparativo de optimalJ y ArcStyler,” p. 106, 2004.
- [8] “Integranova MES,” 2011.
- [9] L. H. Zhong, “Model driven software architecture evolution information capture,” *Proc. - 2015 2nd Int. Conf. Inf. Sci. Control Eng. ICISCE 2015*, pp. 313–317, 2015.
- [10] R. Pros-tr--, “Research Centre in Software Production Methods Informe Técnico / Technical Report COMMUNICATION ANALYSIS AND THE OO-METHOD : MANUAL DERIVATION OF THE CONCEPTUAL MODEL THE SUPERSTATIONERY CO . LAB DEMO Authors ( in alphabetical order ): Sergio España , Art,” no. December 2010, 2011.