

PROYECTO IS-1/00/2017: *Polaflix*

## Polaflix – Mapeo Objeto-Relacional

### Resumen

Este documento contiene un resumen de la fase de traducción objeto-relacional realizada sobre el modelo de dominio del sistema *Polaflix*, detallando los patrones de diseño que se han utilizado durante esta traducción, y el motivo por el que se han utilizado esos patrones en concreto.



## HISTORIAL DE CAMBIOS

Versión	Fecha	Cambio	Responsable
0.1	20-12-2016	Añadida Sección 2. Detallada la traducción de las clases: Serie, Temporada, Capitulo, Usuario, Visualización, Creador, Artista.	Isaac
0.2	27-12-2016	Añadido Resumen, Introducción y Sumario. Aclarado mecanismo de generación de claves para todas las relaciones.	Isaac
0.3	17-01-2017	Añadido esquema relacional obtenido.	Isaac
1.0	18-01-2017	Revisión Final.	Isaac



## 1. Introducción

El presente documento recoge un resumen del proceso seguido durante la traducción objeto-relacional del modelo de dominio del sistema de reproducción de series en línea, *Polaflix*.

El documento se compone de una sección principal -subdividida en varias secciones-, a la cual acompañan la presente introducción y su correspondiente sumario.

## 2. Transformación del modelo de dominio en un esquema de bases de datos relacional.

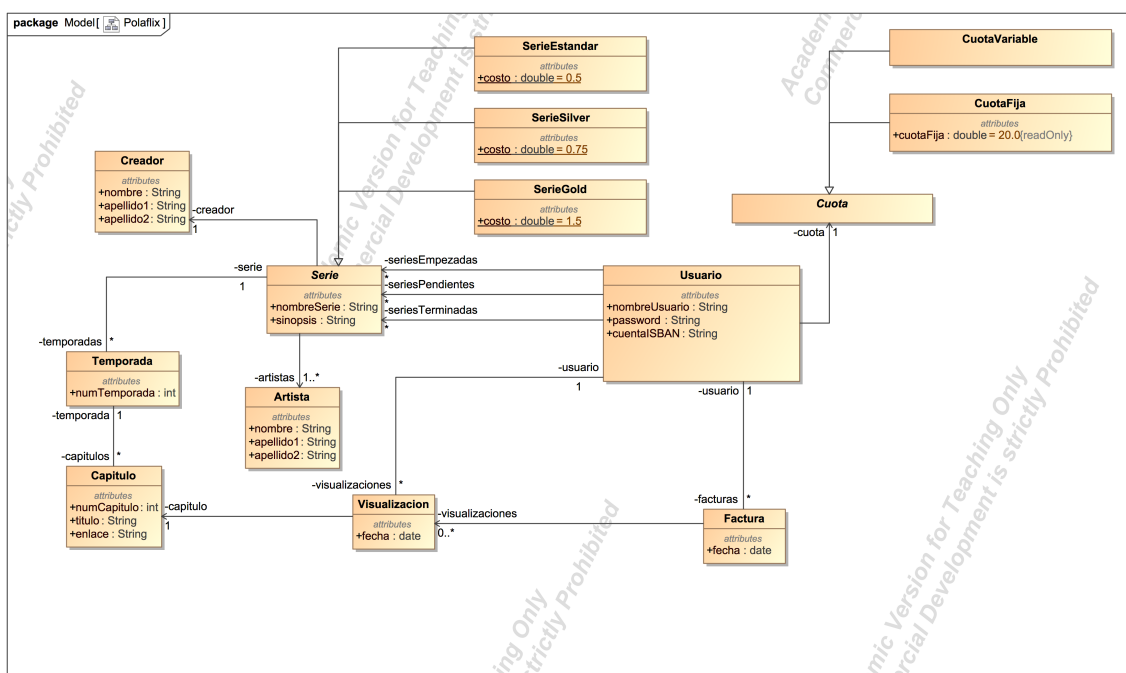


Figura 1. Modelo de Dominio

### 2.1 Serie

La clase Serie se transforma en la siguiente tabla:

<u>idSerie</u>	nombre	sinopsis	costoSerie	tipoSerie	creador
----------------	--------	----------	------------	-----------	---------

En este caso, se ha utilizado el patrón *Single Table*, indicando el tipo concreto de serie, en un campo. Se ha tomado esta decisión debido a que el número de atributos de las clases hijas es bajo, por lo que no se generan muchos valores nulos. De este modo se facilita la recuperación de Series –independientemente de su tipo concreto-, siendo esta una operación que se va a realizar un elevado número de veces. El creador de la serie se ha establecido aplicando el patrón *Foreign Key Mapping*.

Como identificador se ha establecido una clave artificial. No se ha establecido el nombre de la serie como identificador, ya que este pudiera modificarse posteriormente. De esta forma se permite además que existan dos series con el mismo nombre, ya que este es un caso que pudiera darse.

Para la relación *n a n* existente entre: *Serie* y *Artista* se ha utilizado el patrón *Association Table Mapping (Many to Many)*, creando para ello la siguiente relación.

<u>idArtista</u>	<u>idSerie</u>
------------------	----------------

Para esta relación, *Many to Many*, se ha establecido como clave primaria la unión de las claves primarias de ambos elementos. De esta forma se evita que aparezca varias veces el mismo par *idSerie-idArtista* con distintas claves primarias artificiales. Debido a que estas claves son artificiales, no debería haber problemas en el futuro ya que, por su naturaleza, estas no cambiarán.

## 2.2 Creador

La clase *Creador* se mapea en la siguiente tabla:

<u>idCreador</u>	nombre	apellido1	apellido2
------------------	--------	-----------	-----------

Como identificador de esta relación se ha establecido una clave artificial, ya que no puede conseguirse un identificador unívoco de cada creador a partir de sus atributos.

## 2.3 Temporada

La clase *Temporada* se traduce en la siguiente relación:

<u>idTemporada</u>	idSerie	numTemporada
--------------------	---------	--------------

Para esta traducción se ha utilizado el patrón *Foreign Key Mapping*, estableciendo una relación entre la serie y la temporada.

Como identificador de la relación se ha establecido una clave artificial, protegiendo de esta manera el modelo frente a posibles cambios en la identificación de las temporadas.

## 2.4 Capitulo

La clase *Capitulo* se traduce en la siguiente relación:

<u>idCapitulo</u>	numCapitulo	titulo	enlace	idTemporada
-------------------	-------------	--------	--------	-------------

Para esta traducción se ha utilizado el patrón *Foreign Key Mapping*, estableciendo una relación entre el *Capitulo* y la *Temporada*.

Como identificador de la relación se ha establecido una clave artificial, protegiendo de esta manera el modelo frente a posibles cambios en la identificación de los capítulos.

## 2.5 Usuario

La clase *Usuario* se traduce en la siguiente relación:

<u>idUsuario</u>	nombreUsuario	password	cuenta_ISBAN	cuota
------------------	---------------	----------	--------------	-------

La relación n a n existente entre *Usuario* y *Serie* se traduce –mediante la utilización del patrón *Association Table Mapping*- en la creación de la siguiente relación:

<u>idSerie</u>	<u>idUsuario</u>	estado
----------------	------------------	--------

El estado reflejado en la relación entre usuario y serie, tomará los valores “*Pendiente*”, “*Empezada*” o “*Terminada*”. Las columnas *idSerie* e *idUsuario* suponen la utilización del patrón *Foreign Key Mapping*.

Como identificador de la relación se ha establecido una clave artificial. Pese a que el nombre de usuario se entiende como unívoco, este pudiera cambiar, por lo que no es una buena opción como clave primaria.

## 2.6 Visualización

La clase *Visualización* se traduce en la siguiente relación:

<u>idVisualizacion</u>	idCapitulo	idUsuario	fecha
------------------------	------------	-----------	-------

Para esta traducción se ha utilizado el patrón *Foreign Key Mapping*, estableciendo relaciones entre la *Visualización* y:

- *Capítulo* (a través de su identificador).
- *Usuario* (a través de su identificador).

Como identificador de la relación se ha establecido una clave artificial. Pudiera haberse establecido como clave principal la unión *idCapitulo-idUsuario*, ya que ambas son artificiales y el conjunto resultante es una clave unívoca, sin embargo, si se estableciera esta clave, no podría registrarse más de una visualización de un capítulo para un mismo usuario. Este problema podría solventarse uniendo la fecha a la clave primaria. Sin embargo, se utiliza una clave artificial para proporcionar independencia entre la lógica de negocio y el esquema de la base de datos.

## 2.7 Artista

La clase *Artista* se traduce en la siguiente relación:

<u>idArtista</u>	nombre	apellido1	apellido2
------------------	--------	-----------	-----------

Como identificador de esta relación se ha establecido una clave artificial, ya que no puede conseguirse un identificador unívoco de cada artista a partir de sus atributos.

## 2.8 Factura

La clase *Factura* se traduce en la siguiente relación:

<u>idFactura</u>	fecha	idUsuario
------------------	-------	-----------

Como identificador de esta relación se ha establecido una clave artificial, ya que no puede conseguirse un identificador unívoco a partir de sus atributos.

El usuario a quién pertenece esta factura se ha establecido aplicando el patrón *Foreign Key Mapping*.

Para la relación 1 a n existente entre *Factura* y *Visualización* se ha utilizado el patrón *Association Table Mapping*, creando para ello la relación:

<u>idFacturaVisualizacion</u>	idVisualización	idFactura
-------------------------------	-----------------	-----------

Como identificador de esta nueva relación se ha establecido una clave artificial, ya que en este caso no tiene sentido la utilización de una clave formada por el resto de atributos (*idVisualización-idFactura*).

El esquema resultante es el siguiente:

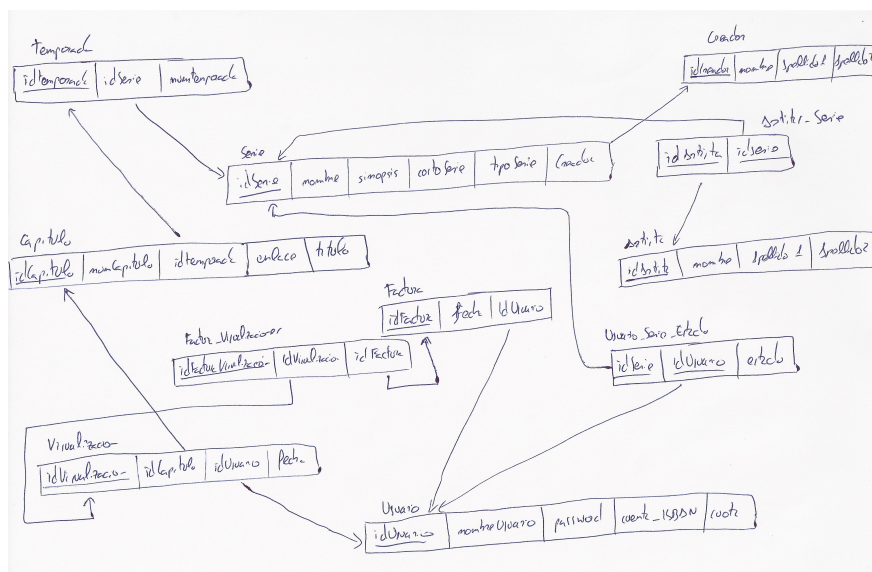


Figura 2. Esquema Relacional obtenido.



### 3. Sumario

Este documento ha recogido el resumen del proceso seguido para convertir el modelo de dominio del sistema *Polaflix* en un esquema relacional que pueda ser persistido en una base de datos de este tipo.