

Domain-Driven Design

El modelo de dominio se conforma de dos aggregates, uno de ellos lo conforman las clases Usuario, Factura, TipoFactura, FacturaService y VisualizacionCapitulo, donde el aggregate root es Usuario. El otro aggregate se conforma por Serie, Temporada, Capitulo, Categoria, Persona y SerieService, donde el aggregate root es Serie.

Ambos aggregate root son entity, junto con las clases Temporada, Capitulo, Factura, Categoria, TipoFactura y VisualizacionCapitulo y FacturaService y SerieService son dos service.

Siguiendo las reglas de Domain-Driven Design los entity tienen identificadores únicos, el Usuario tiene su nombre de usuario, la Factura la fecha y el usuario conforman el identificador. En la Serie podría ser el nombre, pero es probable que haya dos series con mismo nombre por lo que igual es conveniente hacer un identificador artificial. Para la temporada y capítulo se podría conformar el identificador con la serie y el número de temporada para Temporada y añadiendo el número de capítulo para Capitulo. Categoria parte del nombre de la categoría como identificador, al igual que TipoFactura que será el nombre del tipo de factura y VisualizacionCapitulo tendrá como identificador la fecha de visualización junto con el usuario y el identificador global del capítulo conformado por los números de Capitulo, Temporada y Serie. Las operaciones de ambos service toman elementos del dominio como Serie o Factura para poder operar y no tienen estado, es decir, en todo momento que se aplique, el resultado será el mismo.

Respecto a los aggregate y aggregate root, los root tienen sentido global en el dominio y los elementos externos al root tienen sentido local respecto a este, Temporada y Capitulo tienen sentido dentro de la Serie y Factura tiene sentido dentro del Usuario. Los elementos externos a un aggregate identifican al root, este es el caso de Usuario y VisualizacionCapitulo que referencian a Serie y toman de ella todos sus datos.

