

Entidades

- Usuario. Porque tiene identidad propia y a pesar de que cambien sus atributos, el usuario seguirá siendo el mismo. Por ejemplo, si cambiamos su cuenta bancaria o contraseña, sigue siendo el mismo usuario.
- Capítulo. Porque representa un elemento único en el sistema, aunque se cambien sus atributos como el nombre o la descripción, el capítulo seguirá siendo el mismo. Por ello, tendrá que ser monitorizado.
- Serie. Porque al igual que capítulo son elementos únicos, no tendremos dos réplicas de una serie en nuestro sistema. Estas podrán modificar su descripción o título por ejemplo, pero seguirán siendo la misma serie. Además de tener muchas relaciones y componer otras entidades.
- Factura. Porque representa un objeto único y que tiene muchas probabilidades de ser modificado, al ir añadiendo dinámicamente las visualizaciones que se hacen. Aunque solo se aplica en las facturas actuales.

Objetos Valor

- Actor. Porque representa un concepto dentro del dominio y una vez que se crea no tiene previsto modificarse. Únicamente sirve para representar un dato, actualmente con un atributo pero que podría tener más en un futuro.
- Creador. Análogo a Actor.
- Capítulo Visto. Porque no tiene una existencia independiente, si no que surge de la relación entre los capítulos y el usuario que ve los mismos.
- Cuenta Bancaria. Porque he creado la clase para representar un dato como es el IBAN (y posibles futuros datos bancarios) que es propio del usuario. Ya que este tipo de dato no existe de forma nativa. Por tanto, puedo tener dos usuarios con una misma cuenta bancaria sin problemas.
- Categoría. De la misma forma que antes, el propósito de categoría es únicamente representar un tipo de dato que indica el precio de la serie. Tendremos muchas veces objeto de categoría duplicados sin necesidad de monitorizar cada uno.
- Temporada. Su única función surge de la necesidad de agrupar los capítulos, pero no se trata de una clase con identidad propia. Se identifica únicamente por el conjunto de sus atributos que definen la relación entre el capítulo y la serie.

Servicios

- Servicio Polaflix. Porque realiza operaciones relacionadas con todo el dominio de la aplicación como el cobro de facturas o la búsqueda de series. Estas operaciones no son propias de ninguna otra clase, sino que interactúan con varias entidades.

Agregado 1 “Usuario”

Entidades y objetos valor propios del usuario que confirma la raíz de este agregado. Que confirma los clientes de la aplicación.

- Usuario (raíz). Se trata de la raíz ya que tiene una identidad global, que se trata del nombre de usuario. Y el resto de los elementos son derivados del usuario.
- Cuenta Bancaria. Es totalmente propiedad del usuario.
- Factura. Cada usuario tiene la suya y si eliminamos el usuario, sus facturas no se cobrarán ni existirán (puede que en un historial, pero no activamente).
- Capítulo Visto. Porque es más dependiente del usuario, ya que sin él, no tendría sentido una visualización. Y sin embargo, si no tenemos la serie que se ha visualizado, la aplicación tendrá que mantener el cobro ya que el servicio se ha consumido.

Agregado 2 “Serie”

Entidades y objetos valor propios de la serie y capítulos. Que conforman el producto que ofrece la aplicación a los clientes.

- Serie (raíz). Es la raíz ya que, al igual que con el usuario tiene identidad global y básicamente, de ella dependen todo el resto de los elementos del agregado.
- Temporada. Su única referencia está con la serie y los capítulos por lo que se encuentra totalmente dependiente del agregado.
- Capítulo. Se trata de uno de los elementos más importantes del agregado ya que es otra de las entidades que confirman el producto de la aplicación. Pero su existencia depende de serie por lo que no es la raíz.
- Actor. Simplemente se utiliza como objeto valor para representar una característica de la serie. Por tanto, debe formar parte de este agregado.
- Creador. Análogo a Actor.
- Categoría. Directamente es un atributo de la serie, por lo que debe estar en su agregado.

Repositorios

Tendremos uno para cada raíz de los agregados para poder realizar las operaciones básicas CRUD. Por tanto solo tenemos dos:

- Usuario
- Serie

Con esta configuración de agregados se respeta la propiedad en la que, entre agregados, las referencias solo son entre raíces o entre elementos internos y la raíz.