

## 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.
- Categoría. Porque la idea es tener una única instancia de cada categoría, de forma que si modificamos el precio de una categoría, este se vea reflejado en todas las series que tengan esa categoría. Es decir, no queremos tener objetos duplicados. Por ello, se trata de una entidad.
- Temporada. Podemos tener dos temporadas que tengan los mismos atributos, pero si pertenecen a series diferentes deben ser temporadas diferentes, no se consideran temporadas iguales. Por ello, son entidades, tienen una identidad propia y no pueden duplicarse.

## 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. Además, como decisión de diseño, así podrán tener dos usuarios la misma cuenta bancaria y si alguno decide cambiarla, que no le afecte al otro.
- Visualizaciones. Se utiliza para guardar los capítulos vistos de los usuarios. Por ello, puede ser que dos usuarios hayan visto los mismos capítulos pero no por ello se trate de la misma entidades, sino que son objetos diferentes que guardan datos de los usuarios. Es decir, podremos tener instancias duplicadas sin que provoque problemas.

## 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 debe una identidad global al no poder tener dos usuarios iguales duplicados en el sistema. Lo cual provocaría problemas. 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).
- Capitulo 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.
- Visualizaciones. Porque al igual que los capítulos vistos, depende del usuario y es un dato que forma parte de su entidad y vida útil.

### **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 no puede estar duplicada. 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.
- Capitulo. 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.