

GeSoc

Decisiones de diseño

**Entrega 3**

[Introducción 2](#_Toc46436265)

[Categorización de entidades 3](#_Toc46436266)

[Etiquetado de egresos 7](#_Toc46436267)

[Calendarización de Validador de Egresos 8](#_Toc46436268)

# Introducción

En esta entrega se agregaron unas pocas entidades al dominio de la aplicación para cumplir con los requerimientos solicitados.

El punto más importante y el que más debates generó es el relacionado con la categorización de entidades.

# Categorización de entidades

* Se decidió modelar a la categoría de cada tipo de entidad como una clase. Si bien consideramos que sería válida la utilización de herencia para extender el comportamiento de cada entidad, creemos que conceptualmente la categoría de una entidad es un **atributo** de esta.
* Considerando que el comportamiento de cada categoría varía según distintos factores (por ejemplo si es categoría de una entidad base o jurídica), decidimos que la clase Categoría siga los lineamientos del patrón Decorator; permitiendo al usuario configurar si esta categoría debe validar el egreso y la entidad base antes de ser añadidas, a través de los atributos booleanos validaEgreso y validaEntidadBase respectivamente y, en caso de tratarse de la categoría de una entidad base, permitimos que el usuario configure si esta entidad permite ser añadida a una entidad jurídica por medio del atributo booleano permiteAgregarseAEntidadJuridica.
* Para indicar si la categoría corresponde a ONG, Judicial o Industria Agropecuaria, se utilizó un enum “TipoCategoria”, que es atributo de cada categoría. Si a futuro se precisa desarrollar alguna diferenciación a la hora de tomar decisiones entre una categoría del tipo ONG, Judicial o Industria Agropecuaria, vemos como necesario que cada categoría tenga un enum asociado para que el mismo pueda ser usado, por ejemplo, en un condicional, según el caso específico.
* Para el método “validarEgresoParaEntidad”, se decidió que el monto máximo fijo permitido a una entidad se encuentre dentro de su categoría, entendiendo que este monto depende entonces de la categoría específica y no de la entidad en sí. También, creemos que debe tomarse el precio total del nuevo egreso para decidir si se acepta o no, ya que si tomamos la suma de los precios de egresos anteriores asignados a la entidad podríamos estar sumando egresos que ya han sido pagados. Este método se llama desde la entidad (jurídica o base) antes de agregar un nuevo egreso.
* Para el método “validarEntidadBase” se decidió, debido a que no se explica cuáles son los criterios para validar o rechazar la adhesión de una entidad base a una entidad jurídica, que haya un número máximo de entidades base que una entidad jurídica puede tener asociada. Este número depende, nuevamente, de la categoría de la entidad jurídica, y se verifica si el mismo fue superado o no dentro del método. También se comprueba si la entidad base admite ser añadida a una entidad jurídica chequeando el valor del booleano que se encuentra en su categoría. Como tampoco hay definiciones sobre el criterio por el cual una entidad base admite ser o no añadida a una entidad jurídica, se decidió utilizar por el momento un booleano cuyo valor es configurado por el usuario según su criterio. Este método se llama desde la entidad jurídica antes de agregar una nueva entidad base.

Captura de pantalla de un celular

Descripción generada automáticamente

# Etiquetado de egresos

* Con esta entrega, cada egreso podrá tener una etiqueta asociada.
* Modelamos la “Etiqueta” como enum, dado que por el momento no tendrá asociado un comportamiento complejo.
* Los valores iniciales de etiquetas serán:
  + AMOBLAMIENTO
  + INDUMENTARIA
  + PROVEEDOR\_ELIMPERIO
  + HOGAR
  + DECORACION
* Dado que las entidades conocen sus egresos (y estos a sus etiquetas), decidimos que el mensaje a través del cual, dada una etiqueta, se recuperen todos los egresos de una entidad, está situado dentro de las entidades.

# Calendarización de Validador de Egresos

* Se creo una nueva entidad, TareaProgramadaValidadorDeEgresos que, mediante un Trigger, ejecuta periódicamente el método “execute” de la clase batchValidacionEgresos.
* El trigger fue seteado inicialmente como "0/5 \* \* \* \* ?", es decir, se ejecutará el job desde el segundo 0 de iniciado el programa, cada segundos segundos, indefinidamente.
* Sobre este punto surgió la inquietud respecto a la “vida” del job. Es decir, la ejecución del job va a comenzar una vez se inicie el método main de la nueva entidad. Entendemos que en posteriores entregas haremos una invocación del programa, desde el sistema operativo.