1 Justificación del Diseño Inicial

Diagrama de Clases del Dominio - Entrega 1

Al momento de diseñar la solución, identificamos las siguientes clases; Vianda, Vulnerable, Heladera, PersonaJuridica, PersonaFisica y Colaboracion.

Originalmente pensamos que PersonaJuridica y PersonaFisica podían llegar a tener un comportamiento parecido y aplicar herencia, aunque la misma podría llegar a ser forzada ya que no alcanzan a tener lo mencionado.

Elegimos modelar las colaboraciones como una única clase Colaboración, que contiene todos los datos posibles y que además contiene un enumerado con el que, en un futuro, podremos distinguir cada una de las donaciones en la implementación hecha. El enumerado contiene los cuatro tipos de colaboraciones posibles, pudiendo elegir uno de ellos dependiendo del tipo de persona que lo realice, y este nos permite agregar fácilmente más comportamientos (extensibilidad). En el caso de que una colaboración no precise de uno de los campos, los mismos no serían inicializados. En caso de que cambien los campos necesarios (ya sea que se agreguen campos o se quiten, para una colaboración existente o una nueva), tener una única clase Comportamiento con todos los campos resulta más mantenible.

Diagrama de Casos de Uso

En el Diagrama de Casos de Uso identificamos 3 actores: PersonaFisica, PersonaJuridica y ONG. Consideramos que tanto las personas físicas como las jurídicas pueden registrarse si así lo desean. Además, cada uno de ellos puede colaborar a partir de los casos de uso que le corresponden a sus posibles colaboraciones. Estos casos incluyen el caso de uso “completar formulario”, ya que es un paso necesario para cualquier tipo de colaboración. Este último caso de uso también incluye el de registro, debido a que lo consideramos necesario para poder realizar el formulario.

También relacionamos a los 3 actores con el caso de uso que permite la visualización del mapa de las heladeras, ya que lo evaluamos como algo necesario para los 3. Finalmente, incluimos una serie de casos de usos para la ong, que le permiten administrar heladera, administrar colaborador y dar de alta personas vulnerables.

En lugar de crear los casos de uso dar de baja heladera, dar de alta heladera y modificar heladera, elegimos crear el caso de uso administrar heladera que contemple estos 3 casos, porque permite ampliar las posibilidades una vez realizado el documento de especificación de casos de uso (y así ser visualmente más claro). Esto mismo también se aplica al caso de uso administrar colaborador.

2 Justificación de la Entrega 2

Diagrama de Clases del Dominio

Agregamos nuevas clases necesarias para representar los cambios en el dominio.

A las personas en situación vulnerable les agregamos un atributo Tarjeta, siendo la Tarjeta una clase que representa a las tarjetas plásticas utilizadas para acceder a las heladeras. Las tarjetas tienen un código y una cantidad de usos máxima, y también están compuestas por un RegistroUso, que permite registrar la fecha y la heladera en la que fue utilizada una tarjeta, y un InformacionRegistro, que registra al colaborador que le entregó la tarjeta y el vulnerable al que pertenece la misma. La distribución de las tarjetas, al ser otro tipo de colaboración, la agregamos al enum TipoColaboracion (en este caso DISTRIBUIR_TARJETAS). A estas tarjetas se les puede consultar si es posible su uso en una heladera.

Luego, a las Heladeras les agregamos un modelo y sensores, que permiten conocer el estado en tiempo real de cada heladera (con el atributo estaActiva y su método getter) por si se da el caso en el que una heladera sufra algún desperfecto. Además, la heladera permite calcular la cantidad de meses que estuvo activa la heladera, para algo que explicaremos más adelante. El modelo de la heladera cuenta con el nombre del modelo y un sensor de temperatura. El sensor de temperatura conoce la heladera a la que pertenece, y contiene las temperaturas máxima y mínima de la heladera, al igual que contiene la última temperatura registrada por el sensor. Este sensor puede activarse o desactivarse según si ocurre algún desperfecto. El sensor de movimiento cuenta con los métodos necesarios para activarse y desactivarse en caso de algún suceso inesperado y que permiten enviar una alerta al sistema en caso de que una heladera este siendo robada.

Se agregó la clase Tecnico, incluyendo todos los campos que el sistema debe registrar de los mismos.

Adicionalmente, agregamos un Reconocimiento a los colaboradores. Este consta de puntos que pueden ser intercambiados por diversos productos y servicios ofrecidos por empresas asociadas a la ONG (desarrollado más adelante). Aparte de los puntos, el registro cuenta con una Fórmula para calcular los puntos obtenidos por colaboración a partir de una serie de coeficientes establecidos a priori y los cuales pueden ser cambiados. También permite reducir los puntos, en caso de que el colaborador los intercambie.

Como mencionamos anteriormente, agregamos las empresas asociadas como un TipoColaborador, y las estas pueden colaborar con la ONG al hacer ofertas (siendo esto un nuevo tipo de colaboración)

Diagrama de Casos de Usos

Agregamos los casos de uso de “Administrar técnico”(actor ONG), “Entregar tarjetas” (actor PersonaFísica), “Ofrecer producto” (actor Empresa Asociada), “Cargar colaboradores masivamente” (actor ONG), “Canjear Puntos” (actores PersonaFísica y PersonaJurídica) y “Solicitar recomendación de puntos de colocación” (actor PersonaJurídica).

Recomendación de puntos

Se empleó el uso de mocks dentro de lo que es Postman, en este caso se creó un mock server para poder simular las peticiones de una mock collection creada actualmente, que la misma contiene la información acerca de 50 puntos (cada uno conteniendo latitud, longitud y referencia). Es posible hacer una consulta GET acerca de cualquiera de los 3 atributos, o incluyendo cada uno.

Carga masiva de colaboradores

Lo que hicimos para poder realizar la carga masiva de los colaboradores es la implementación de DTOs (Data Transfer Objects), los cuales sirven para interpretar las cosas que se están trayendo desde el archivo CSV, para posteriormente cargarlo en cada Colaborador y Colaboración.

Adicionalmente, se implementó el Controller para el Colaborador, que es quien maneja los DTOs (básicamente transforma entradas en salidas), delegando la decisión de crear un colaborador en caso de que no se encuentre en el repositorios, llamando a Service. El Service es el encargado de crear las instancias de las clases las cuales involucran los DTOs.