



Justificaciones de Diseño - TPA1 DDS

Requerimientos de dominio

Para cumplir con el requerimiento de permitir el alta, baja y modificación de colaboradores, decidimos implementar las clases `PersonaHumana` y `PersonaJuridica`, con los atributos correspondientes a cada tipo de colaborador. Estas clases tienen un método `colaborar(colaboracion)` que polimórficamente permitirá realizar colaboraciones.

Ambos tipos de persona cuentan con un objeto usuario asignado que incluye un nombre de usuario y una contraseña. Para modelar las formas en que una persona puede colaborar al registrarse, decidimos utilizar un set que contiene las formas de colaboración, las cuales están representadas por un enum (`FormaColaboracionHumana` y `FormaColaboracionJuridica` respectivamente). Además, ambas clases también incluyen una lista que registra su historial de colaboraciones

Modelamos las colaboraciones con distintas clases que implementan dos interfaces, `ColaboracionHumana` y `ColaboracionJuridica`, ambas con un método `realizarColaboracion()`, que efectúa los cambios producidos por esa colaboración.

Implementamos la clase colaboración `DonacionDeVianda`, permitiendo que, al indicar las viandas y la heladera, se delegue a esta última la responsabilidad de agregar las viandas a la misma mediante el método `ingresarViandas(List<Vianda> viandas)`.

Para permitir el registro de personas en situación vulnerable, creamos la clase `PersonaVulnerable`. En esta clase, agregamos un atributo de tipo `Usuario` que permite a cada persona vulnerable tener un nombre de usuario y contraseña únicos para acceder al sistema.

Para cumplir el requerimiento de permitir el alta, baja y modificación de heladeras creamos una clase singleton, `MapaHeladeras`, que contiene una lista de todas las heladeras existentes. Esta clase posee métodos para agregar, listar y quitar heladeras. Que ésta clase sea un singleton nos permite que, dentro de la colaboración de `HacerseCargoHeladera`, el método `realizarColaboración` pueda crear una heladera nueva y agregarla al mapa mediante la instancia global.

Las heladeras son modeladas con una clase que contiene su ubicación, nombre, capacidad, fecha de creación, y una lista de las viandas que contiene, teniendo métodos para ingresar una lista de viandas a la heladera y para sacar una cantidad de viandas (ambos usados en la colaboración de distribución de viandas).

Requerimientos de seguridad

Para cumplir con el requerimiento de permitir que el usuario se pueda registrar, decidimos modelar una clase llamada Usuario en donde se encuentran los atributos usuario y contraseña, también se encuentra desarrollado un método llamado "verificarContraseña(contraseña)" en donde se valida la contraseña ingresada.

Para cumplir con los requerimientos de seguridad en cuanto a la contraseña del usuario según OWASP desarrollamos la comprobación dentro del constructor de la clase Usuario. Ahí, comparamos la contraseña del usuario con un archivo que contiene las 10.000 peores contraseñas para verificar si no está entre ellas y también si su longitud es mayor a 8.

Casos de Uso TPA1

Grupo 08 | Primera entrega

