Entregas - 1° Cuatrimestre

Sábado 23/5/15

El objetivo de esta entrega es diseñar los siguientes casos de uso:

- Planificar comida
- Generar recomendación por perfil
- Listar recetas para usuario
- Armar grupos de usuario

Observaciones generales:

- Cuando se dice que debe generarse un error, debe ser una excepción que tenga sentido interpretar, por ejemplo no debe ser NullPointerException.
- Los assertTrue, assertNull y los que no arrojen información sobre lo que esta fallando, deben tener una descripción del error generado (por ejemplo: “el usuario X no está en el grupo Y”). En general el assertEquals es más descriptivo, pero recuerden que mientras más se aclara el error más fácil es luego hacer un refactor.
- Documentar cualquier definición que sea necesaria para definir los tests (por ejemplo alguna situación ambigua o condiciones que consideran que no tengan sentido).
- Por ahora suponemos que existen las clases:
    - RepositorioUsuario: donde están todos los usuarios y grupos
    - RepositorioRecetas: donde están las recetas precargadas

Usuario:

- Que no haya 2 usuarios con el mismo e-mail (vamos a tomar que el nombre de usuario es el e-mail)
- Validación de datos obligatorios:
    - Nombre
    - Edad (>18)
    - E-Mail (e-mail valido)
- Agregar preferencia (por ahora son strings, son como tags)
    - Si se agrega una preferencia que ya estaba no debe tener efecto

Grupo:

- No puede haber 2 grupos con el mismo nombre
- Al agregar un usuario:
    - El mismo es miembro del grupo
    - El grupo es uno de los grupos del usuario
    - Si se agrega 2 veces el mismo usuario debe generarse un error
- El usuario puede compartir recetas al grupo si es miembro
- Listar recetas de grupo
- Al quitar un miembro:
    - Si se trata de quitar un usuario que no es miembro debe generarse un error
    - Las recetas del usuario que compartió ya no deben estar disponibles para el grupo

Planificar comida (Definir donde asignar las responsabilidades):

- Consultar una planificación realizada (Ej: que voy a desayunar el 15/05?: nada / receta X)
- Verificar el efecto de una replanificación (por otra receta o si no se come nada)
- No se puede planificar/replanificar más allá de una semana (hacia adelante o atrás)
- No se puede planificar una receta en un horario que no corresponde (desayuno, merienda, almuerzo, cena)
- Las recetas disponibles para un usuario son:
    - Las que creo + las x defecto + las que están en otro grupo
    - Antes que nada deben respectarse las restricciones:

- Los vegetarianos no se le debe listarse ninguna receta que tenga como ingrediente “carne”
- Los diabéticos no se le debe listar ninguna receta que tenga como ingrediente “azúcar”

Domingo 14/6/15

Se deben generar al menos 1 caso por escenario principal y otro por cada alternativo (y condiciones de borde que detecten).

Generar Receta:

- No debe permitir que existan dos recetas iguales (2 recetas son iguales cuando tiene el mismo nombre y la misma persona)
- Validación de datos obligatorios:
  - Nombre de receta
  - Ingredientes (lista de valores) (deben ser 2 o más)
  - Procedimiento (Explicación de la receta)
  - Dificultad de la receta (lista de valores: 1 a 5 estrellas)
  - Temporada recetario
  - Calorías total de la receta

Calificar Receta:

- No se debe permitir calificar una receta de un grupo al que no se pertenece
- La calificación de la receta es visible solo para los miembros del grupo (listarCalificacionesReceta)
- Se puede modificar la calificación de la receta en cualquier momento pero no se debe poder calificar más de una vez la misma receta
- Se debe poder pedir un ranking de recetas por grupo (por promedio)
- Si un usuario comparte una receta en más de un grupo, las calificaciones no deben mezclarse.

Modificar o Eliminar Receta:

- Eliminar una receta no debe influir en los historiales de los usuarios
- Si un usuario se borra de un grupo, no debe influir en el historial de los usuarios
- Al modificar una receta no se deben presentar cambios en el historial (Ej: Si se cambian los ingredientes y/o las calorías)