

Finalizado ▾

# Ciencia Participativa y Juegos

- Poder hacer que la aplicación móvil recolecte los datos y no el usuario

Datos = Muestra

? ( La ciencia participativa (o ciencia ciudadana) es una metodología de investigación que propone integrar a **personas** que no están vinculadas a **instituciones** de investigación a **proyectos de investigación en sus diferentes etapas**, desde el **diseño del proyecto** hasta la **recolección de datos**. Estos proyectos poseen una gran diversidad de actividades y áreas. Existe un tipo particular de proyectos de ciencia participativa en donde la actividad principal es mediada por una **aplicación móvil** y con la misma **se recolectan datos** geográficamente posicionados. Por ejemplo, se recolecta información sobre árboles, y de cada árbol se obtiene una **foto**, la **ubicación** y **algunas características específicas** como la especie y el estado de las hojas.

List

Checkear

¿Proyecto tiene una lista de actividades?

Para que estos **proyectos** posean mayor participación del público en general, una de las alternativas es sumarle algunas **características de juegos**. De esta forma, la participación en los proyectos, además de perseguir los **objetivos** científicos, contempla participar de alguna **actividad lúdica**. Las actividades lúdicas se describen de diferentes maneras, una de ellas es incorporando elementos de ludificación conocidos como **game elements**. Un **game element** es un elemento al que se lo vincula con algún **propósito específico** de lógicas de juego, por ejemplo cumplir un **desafío**, obtener una cantidad de monedas, ganar medallas, sumar puntos, etc.

GameElement  
como interface o  
clase abstracta

Desafío es una clase

Lista de : gustos, intereses y  
motivaciones para persona

Sin embargo, la incorporación de elementos de juegos en estos proyectos no es tan sencilla. Las **personas** poseen diversidad de **gustos**, **intereses** y **motivaciones**. Aunque agregar elementos de ludificación a los proyectos parece una buena alternativa, los mismos deben ser adecuados a los gustos de cada persona. **Por ejemplo, si a una persona que no le gusta caminar se le presenta como desafío del juego recolectar muestras para las que debe caminar muchas cuadras, puede ocurrir que en lugar de motivarla a caminar, por el contrario, lo que haga es que abandone la aplicación.** Es por este tipo de razones que los elementos de ludificación **deben adecuarse a las particularidades de cada persona.**

Posible patrón  
¿Strategy?

La problemática ligada a este trabajo práctico es modelar y programar una solución orientada a objetos que permita identificar elementos de ludificación a participantes de proyectos de ciencia participativa acorde a sus preferencias.

El dominio en general

## Proyectos ✓

El proyecto es el que dirige todo/ organiza

Los proyectos de ciencia participativa pueden ser muy diversos. En este caso no se tendrán en cuenta los detalles organizativos, ni la definición de los protocolos de recolección. Sin embargo hay una serie de caracterizaciones que son de importancia. En principio de un proyecto se debe poder conocer el **nombre** y una **descripción**. Además debe ser posible conocer a qué **categorías específicas** pertenece, por ejemplo si está vinculado a la

astronomía, la biología, etc. El listado de categorías es común para todos los proyectos y cada proyecto puede pertenecer a una o más categorías.

Vincula la clase usuario y muestra

A los proyectos debe ser posible **suscribir usuarios** como participantes activos. También es importante para un proyecto conocer cuáles fueron las muestras que se han recolectado y que pertenecen a ese proyecto (ver sección **Muestras**). Básicamente a un proyecto se deben poder agregar muestras y conocer las muestras que se han recolectado.

→ No es observer

## Muestra ✓

Relación entre muestra y usuario

- Agarro una muestra y le pregunto al proyecto si le interesa, en el caso que sea correcto la va a registrar y sino no pasa nada

Una muestra es la unidad de trabajo básica que aportan los usuarios a los **proyectos** de ciencia participativa. La información a registrar de cada muestra depende del interés del proyecto. Por ejemplo, en un proyecto de recolección de información sobre árboles, la información que se pide es **la ubicación** (geo coordenadas), **una foto del árbol, una foto de alguna hoja y tipo de árbol**. Una muestra estará formada por las respuestas a esa información por cada árbol que se recolecta incluyendo además el **usuario** que realizó la muestra.

Está asociada la muestra al proyecto

Una lista de características (List<String>)

En este trabajo, de las muestras solamente nos interesará registrar **las coordenadas geográficas, el usuario que la realizó, la fecha y hora en que se realizó**. La **información específica de la muestra** (como la foto del árbol o el tipo del árbol) no se modelarán en esta oportunidad.

Acá aclara las variables que tiene la muestra

Aclaración, entonces es una Lista de String

## Usuarios = PERSONAS ✓

Los usuarios son las personas que efectivamente realizan la **recolección de muestras**. Además de las muestras que han recolectado y los **proyectos en los que participan** es importante poder conocer de alguna forma cual es el **perfil** de ese usuario.

Lista<Muestra>

Es una variable

Lista<Proyecto>proyectosEnLosQueParticipa

Los **perfiles** de usuario intentan representar **los gustos, preferencias y comportamientos** específicos que tiene un usuario, **ya sea antes de comenzar a usar la aplicación, como durante la aplicación**. El perfil del usuario será importante para la recomendación de los elementos de ludificación, detallado más adelante.

→ Listas de String

Esto quiere decir que se puede cambiar los gustos e ir agregando nuevos o como sacando (Add, remove)

### ACTIVIDAD LUDICA

## Elementos de ludificación - Desafíos ✓

Los elementos de ludificación son artefactos que permiten incorporar algunas mecánicas y dinámicas de juegos a, en este caso, la participación de los usuarios en los proyectos de ciencia participativa. Existe una diversidad de elementos de ludificación como por ejemplo Desafíos, Insignias, Monedas, Puntos, etc. En este caso los **elementos utilizados** serán los **Desafíos**.

↳ Lista<Desafíos> elementosDeLudificacion

Un desafío es una actividad o serie de actividades que debe poder completarse. De un desafío se debe poder conocer lo siguiente:

## LA CLASE DESAFIO TIENE :

**Área:** el área geográfica donde deben efectuarse las muestras. El área está definida por una **latitud** y una **longitud** que representa el centro de la misma y un radio que determina la distancia desde el centro. → Clase

**Restricciones temporales:** entre dos fechas particulares, durante la semana, durante el fin de semana, también puede combinar, por ejemplo durante los fines de semana entre una fecha y otra fecha. ← Dos variables de tipo fecha, la primera es fechaDeInicio y la segunda fechaDeCierre ambas conforman la restricción

**Cantidad de muestras recolectadas:** un número que indica la cantidad necesaria para batir el desafío. ← int

**Dificultad:** Un número entre 1 y 5, donde 1 es muy fácil y 5 es muy difícil. ← Enum

**Recompensa:** un número que indica el puntaje obtenido como recompensa. ← int

De los desafíos es importante poder indicar, para un usuario, qué desafíos completó, qué porcentaje de completitud posee, momento en que lo superó, saber si lo completó o no y si aún no, el porcentaje de completitud. INTERFAZ

[ Para usuarios : List<Desafio> desafiosCompletados  
Para el desafio : List<IParticipantes> personas

Los desafíos deben ser aceptados por un usuario para que se empiece a contabilizar. Al momento de aceptarlo puede ser que algunas acciones realizadas previamente sean compatibles con el desafío. Sin embargo, las recolecciones que se toman en cuenta para el mismo serán solamente aquellas que hayan sido realizadas posteriormente a la aceptación. Los usuarios además podrán votar cuánto les gusta o gustó realizar este desafío indicando un valor entre 0 y 5, donde 0 significa que no le gustó nada y 5 que le gustó mucho.

● Acordarse de ver la relacion Usuario/Participante con desafio

## Recomendación de elementos de ludificación

Como indicamos anteriormente, los usuarios pueden indicar sus preferencias sobre las características de los desafíos. Las preferencias que el usuario debe poder configurar son: cantidad de muestras a recolectar, dificultad y la recompensas preferidas. Esto será utilizado para evaluar los desafíos más acordes a los gustos del usuario. Es importante notar que las preferencias son formas de determinar un valor numérico para una característica del desafío y que cada usuario puede tener diferentes preferencias para diferentes características del desafío.

Clase Preferencia con una variable de instancia "P" que relacione Usuario con Preferencia.

## Recomendación de desafíos

OJO con recomendar proyectos ya finalizados!

Existen diferentes formas para la recomendación de desafíos para los usuarios del sistema. Aquí va una serie de alternativas las cuales pueden cambiarse durante el ciclo de vida del sistema en forma dinámica. La forma de recomendación se configura por usuario, de esta forma cada usuario puede tener formas diferentes y también pueden configurar que se cambie en cualquier momento cómo sugerirle desafíos.

## Mis preferencias en juego

Esta forma de recomendación selecciona los 5 desafíos con mejor coincidencia (que no ha tomado el usuario anteriormente). La coincidencia de un desafío para un usuario se calcula

Un metodo

metodo

sumando los valores absolutos de las diferencias, para cada característica, entre el valor asignado al desafío y la preferencia del usuario en esa característica. Cuanto más baja sea esa suma mayor es el nivel de coincidencia.

Favoritos → ¿Subclase de preferencia?

Selecciona los 20 desafíos con **mayor coincidencia** y **los ordena por similitud** al desafío realizado que más le gustó (si hay más de uno con el mismo puntaje selecciona uno de ellos al azar). La medida de similitud entre desafíos se define de la siguiente forma: promedio de distancia entre cantidad de muestras recolectadas, dificultad y recompensa de dos desafíos. Por ejemplo: **Tiene diferente comportamiento.**

Como se implementa

| Desafío    | cantidad de muestras recolectadas | dificultad | recompensa |
|------------|-----------------------------------|------------|------------|
| A          | 15                                | 2          | 25         |
| B          | 30                                | 4          | 10         |
| Diferencia | 15                                | 2          | 15         |

Distancia entre A y B =  $(15+2+15)/3 = 10,66$

Una vez ordenado por similitud recomienda los primeros 5.

Patron composite

Buscar proyectos ⇒ STREAMS

La búsqueda de proyectos se debe poder realizar utilizando filtros **ligados a las categorías a los que pertenecen los proyectos**. Debe ser posible combinar las siguientes formas de búsqueda:

- Que incluya ciertas categorías. → Lo tiene la clase proyecto?
- Que excluya ciertas categorías. →
- Que el título del proyecto incluya un texto particular. → Quien debe tener el filtrado?
- Combinar los elementos anteriores utilizando conectores lógicos como el and, or como así también la negación de términos y expresiones.

## Tareas a Realizar

Se deberán completar y entregar los siguientes puntos:

1. Un diseño de la solución completa utilizando diagrama de clases UML.
2. Documentación en un archivo PDF que incluya los integrantes del grupo y sus direcciones de email, las decisiones de diseño, detalles de implementación que merezcan ser explicados, patrones de diseño utilizados y los roles según la definición de Gamma et. al.

3. Implementación completa en lenguaje JAVA que incluya test de unidad con un 95 % de cobertura del código de dominio (Sin los tests)
4. Todo lo anterior debe estar alojado en un repositorio de acceso por parte de los docentes donde se pueda realizar un seguimiento del trabajo.

**Nota:** no se requiere la implementación de interfaces gráficas de ningún tipo. La participación de los integrantes del proyecto debe ser equitativa en función del registro del repositorio git y los commits con los usuarios correspondientes (las discrepancias pueden generar desaprobaciones personales del tp).

- Agregar excepciones
- Hay que revisar, no todo hay que codearlo, hay que verificar si de verdad lo necesitamos
- UML : Falta implementar los mensajes mas rebuscados y verificar



Luego con el paso de que mientras codeemos vemos si se modifica

Pero tiene que estar aunque sea el %90