

## Tópicos Avanzados – 2019

### Trabajo práctico 2

En este trabajo le solicitaremos que mejore el sistema CuOOra trabajado en el TP2 de la materia Técnicas y Herramientas dictada este año (ver [link](#)). A modo de repaso, CuOOra es un sitio de Preguntas y Respuestas que intenta responder preguntas realizadas por usuario. Los propios usuarios del sistema son quienes responde las preguntas realizadas por otros usuarios.

En este trabajo se le solicita implementar los siguientes requerimientos aplicando los patrones de diseño del paradigma orientación a objetos presentados en clases.

Los requerimientos son los siguientes:

- Actualmente el sistema no permite exportar las preguntas y sus respuesta al formato JSon ni XML. Se le solicita que implemente los mecanismos para generar esos archivos. No se descarta exportar la información en otros formatos en el futuro. Por ejemplo, el resultado esperado para ambos formatos es el siguiente:

<pre>{   "preguntas": [     {"id": 1, "titulo": "cómo escribir un string en formato JSON",     "respuestas": [       {"id": 1, "texto": "podes usar un Stream"},       {"id": 2, "texto": "revisaste la librería NEOJSON o FUEL?"}     ]   } ]</pre>	<pre>&lt;?xml version="1.0"?&gt; &lt;preguntas&gt;   &lt;pregunta id="1"&gt;     &lt;titulo&gt;"cómo escribir un string en XML&lt;/titulo&gt;     &lt;respuestas&gt;       &lt;respuesta id="1"&gt;         podés usar un Stream       &lt;/respuesta&gt;       &lt;respuesta id="2"&gt;         revisaste la librería NEOJSON o FUEL?       &lt;/respuesta&gt;     &lt;/respuestas&gt;   &lt;/pregunta&gt; &lt;/preguntas&gt;</pre>
Caso JSon	Caso XML

- Personalización del Home. Cada usuario puede elegir una manera en la que CuOOra le recomienda preguntas al ingresar al Home, se deberán implementar tres opciones de búsqueda:
    - Social: retorna las preguntas de la última semana realizadas por los usuarios a los que sigue, ordenadas por mayor cantidad de votos.
    - Tópicos: retorna las preguntas de la última semana realizadas para los tópicos de interés del usuario, ordenadas por fecha de creación.
    - Más relevantes: retorna las preguntas del día, ordenadas de manera descendente según su cantidad de respuestas.
- El usuario puede cambiar su configuración y por defecto utiliza "Social".

- Estados en la pregunta. En esta segunda versión de la app, queremos llevar la solución a un siguiente nivel de robustez en donde cada pregunta posee un estado bien definido:
  - Creada: una pregunta al ser instanciada se inicia en este estado en donde no se pueden recibir respuestas. Desde este estado solo se puede pasar a estado Activa.
  - Activa: en este estado puede recibir respuestas. Desde este estado solo se puede volver al estado Creada si la pregunta no tiene respuestas cargadas, y siempre se puede pasar a Cerrada.
  - Cerrada: una vez que se alcanza este estado no se pueden recibir respuesta ni volver a estados anteriores.
- Debido al éxito de la aplicación se debe implementar una mejora que permita optimizar el acceso a la información relacionada a las preguntas. Es decir, solo leer la información estrictamente necesaria que se necesita presentar. El sistema posee 100.000 usuarios y un promedio de 100 preguntas cada usuario con 50 respuestas cada una, por lo cual usted deberá utilizar el objeto SluggishDAO (provisto por la cátedra) para almacenar y recuperar los objetos del sistema. Los mensajes soportados son:

#put:anObject  
"almacena un objeto en memoria"

#retrieveAllByClass:aClass  
"retorna todos los elementos almacenados en memoria de una clase demorando 1 segundo"

#select:aBlock forClass: aClass  
"retorna elementos que satisfacen un bloque demorando 200ms por cada objeto de la colección "

Se le solicita las siguientes tareas:

- Diseño de diagrama de clases UML. Realice el diseño de esta nueva versión de CuOOra que implemente los requerimientos mencionados anteriormente. Esta entrega deberá ser aprobada por la cátedra.
- Describir los patrones aplicados y los roles de las clases intervinientes.
- Implementar el nuevo diseño del sistema en Pharo (solo modelo de dominio), implemente Tests para probar el desarrollo.