



Programación con Herramientas Modernas

Trabajo práctico 2021

Dominio y precondiciones

El dominio sigue siendo el mismo de la entrega 0. Asumimos que están implementadas las interfaces de usuario de los casos de uso de la entrega anterior.

Parte 1: Persistencia

La aplicación deberá implementar la persistencia mediante un componente de acceso a datos (Repositorio) que trabaje contra un motor de base de datos relacional. Dado que en las próximas entregas lo volveremos a utilizar, lo más recomendable es que simplemente cambie el componente de acceso a datos de la forma más dinámica posible.

El grupo podrá elegir entre

- crear las entidades en un motor de base de datos relacional, utilizar un script de manipulación de datos mediante SQL y hacer el mapeo mediante JDBC
- Utilizar un framework de mapeo OR/M (Spring Boot o Hibernate) y generar el juego de datos mediante un script SQL o bien mediante llamadas a queries del lado de objetos (en los tests o en el Bootstrap que provee el Framework). *(opción sugerida por la cátedra)*

Responsabilidad grupal

- Definición del modelo de objetos
- Definición de los componentes de acceso a datos
- Definición del script inicial con el juego de datos provisto en la entrega anterior.

Responsabilidad individual

- **Integrante 1 y 2** : Consulta de Preguntas, edición y respuesta.
- **Integrante 3**: Login y perfil de usuario.

Cada integrante debe ser responsable de encontrar los componentes adecuados para cada caso de uso asignado: objetos de dominio que colaboran, controllers, serializadores y vista.

En caso de haber menos integrantes se conversará con el tutor los casos de uso a entregar.

Parte 2: Componentes en la base de datos

Se pide que implementen componentes en la base de datos para resolver los siguientes requerimientos:

1. Conocer las preguntas que creó un determinado usuario.
2. Llevar un control de las modificaciones de texto que se hagan sobre una pregunta, de manera de saber: a) la fecha en la que se hizo la modificación, b) el texto anterior, c) el texto nuevo
3. saber qué usuarios tienen más de 3 respuestas.
4. Evitar que el puntaje de un usuario tome un valor nulo en la base (por fuera de la interfaz de usuario).

Debe decidir cómo modelar cada uno de los casos. Las opciones son:

- vistas
- triggers
- stored procedures
- constraints

Cada integrante será responsable de un punto y deberá justificar su decisión y explicar su resolución en la entrega presencial.