Backend de Aplicaciones 2024 Q2

Guía de Ejercicios - Semana 06

Marcar como hecha

En la presente guía de ejercicios se presentan enunciados para escribir las primeras líneas de código en Java, es decir los enunciados propuestos tienen que ver con poner en práctica los elementos revisados en la presente semana de clases.

1. Simulacro 1

Simulacro de Parcial 1

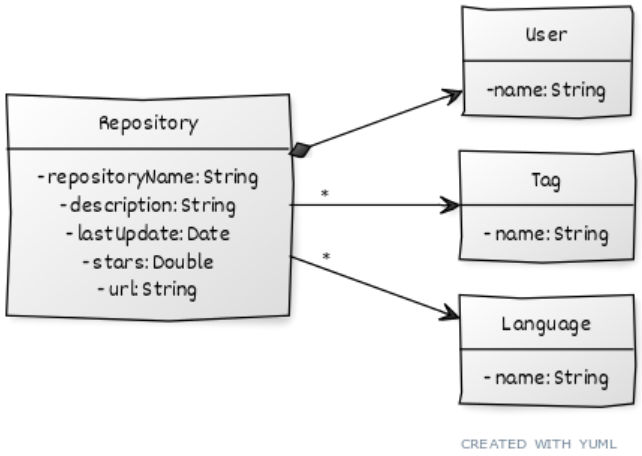
A continuación, se presenta un enunciado Tipo Parcial para práctica y entrenamiento. Hemos tratado de ser lo más fieles posible a lo que será el enunciado de parcial, aunque evidentemente cambiará el dominio del problema, lo que podría generar algunas variaciones o situaciones diferentes a las que aquí se presentan.

Datos:

Usted dispondrá del archivo `Repositories.txt`, que puede descargar haciendo clic aquí. Este archivo es un archivo de texto separado por pipes `|` (se utilizan pipes debido a que hay columnas que tienen atributos separados por comas). Dicho archivo contiene los datos de una lista de repositorios de código y tiene la siguiente estructura:

- **USER_NAME (TEXT)**: Nombre del usuario propietario del repositorio.
- **REPOSITORY_NAME (TEXT)**: Nombre del repositorio, es único y debe ser validada su unicidad.
- **DESCRIPTION (TEXT)**: Descripción del repositorio larga del repositorios y el proyecto.
- **LAST_UPDATE (DATE)**: Fecha de la última actualización del repositorio.
- **LANGUAGE (TEXT)**: Lenguajes utilizados en el repositorio, es una lista separada por comas de los lenguajes identificados por nombre.
- **STARS (REAL)**: Número de estrellas que ha recibido el repositorio.
- **TAGS (TEXT)**: Etiquetas asociadas al repositorio, es una lista separada por comas de las etiquetas identificadas por nombre.
- **URL (TEXT)**: URL del repositorio.

A partir de los datos mencionados, usted debe importar estos a un modelo de objetos en memoria que debería responder al siguiente diagrama de clases:



Requisitos mínimos:

- Todas las clases deben implementar los atributos para manejar las relaciones que se presentan en el diagrama.
- Todas las clases deben implementar constructores y métodos de acceso a los atributos, teniendo en cuenta lo que puede o no puede cambiarse en las entidades de dominio.
- Todas las clases deben implementar algún mecanismo de comparación de acuerdo con las necesidades del modelo.
- Todas las clases deben implementar el método `toString` para mostrar los elementos básicos de la clase.
- La clase `Repository` debe gestionar y exponer la cantidad de estrellas que tiene el repositorio como un número decimal entre 0 y 5 redondeado a un decimal entendiendo como 5 el máximo número de estrellas existente y manejando el resto de forma proporcional. Y además debe exponer la última actualización como la cantidad de días desde la fecha actual.
- La clase `User` debe mantener un listado de los repositorios que le pertenecen y debe exponer la cantidad total de repositorios y el número de estrellas que ha recibido en todos sus repositorios.
- La clase `Language` debe exponer la lista de repositorios en los que ha sido utilizado y la cantidad total de repositorios que lo usan.

Requerimientos:

1. Cargar todos los datos de los repositorios en una colección, asociando cada repositorio a los objetos necesarios, y teniendo en cuenta que cada uno de los objetos asociados debe existir una y solo una vez en la memoria.
2. Utilizar estructuras de datos que permitan mantener el esquema de objetos en memoria de forma tal que no se repitan instancias de los objetos relacionados como usuarios, lenguajes y etiquetas.
3. Determinar e informar la cantidad total de repositorios importados y el número total de estrellas acumuladas por todos los repositorios.
4. Generar un archivo de texto separado por comas con la lista de lenguajes y la cantidad de repositorios que utilizan cada uno de ellos y la suma de estrellas del conjunto de repositorios que utilizan el lenguaje.
5. Mostrar por pantalla la lista de usuarios ordenados alfabéticamente junto con la cantidad de repositorios que tienen y el número total de estrellas que han recibido.
6. Cargar todos los datos montados en memoria en una base de datos utilizando JPA.

Base de datos:

Usted puede organizar la estructura de la base de datos según crea pertinente, teniendo en cuenta que las tablas deben representar el modelo de objetos y que todos los datos importados deben estar cargados y accesibles en la base de datos.

El modelo de datos montado debe permitir al menos las siguientes consultas:

- Repositorios por usuario.
- Repositorios por lenguaje.
- Repositorios por etiqueta.
- Consultar la lista de repositorios de un usuario específico y la lista de lenguajes que utiliza en todos sus repositorios.

◄ Repositorio Semana 6 para Descarga

Ir a...

Material de Clase - Diseño de APIs ►

Descargar la app para dispositivos móviles