

Parcial Único Backend de Aplicaciones 2024

Turno 1

En este parcial vamos a retomar el ejercicio de las mascotas virtuales "Tamagotchi" estudiado durante el cursado. Para los efectos de este parcial, el problema se describe de la siguiente manera:

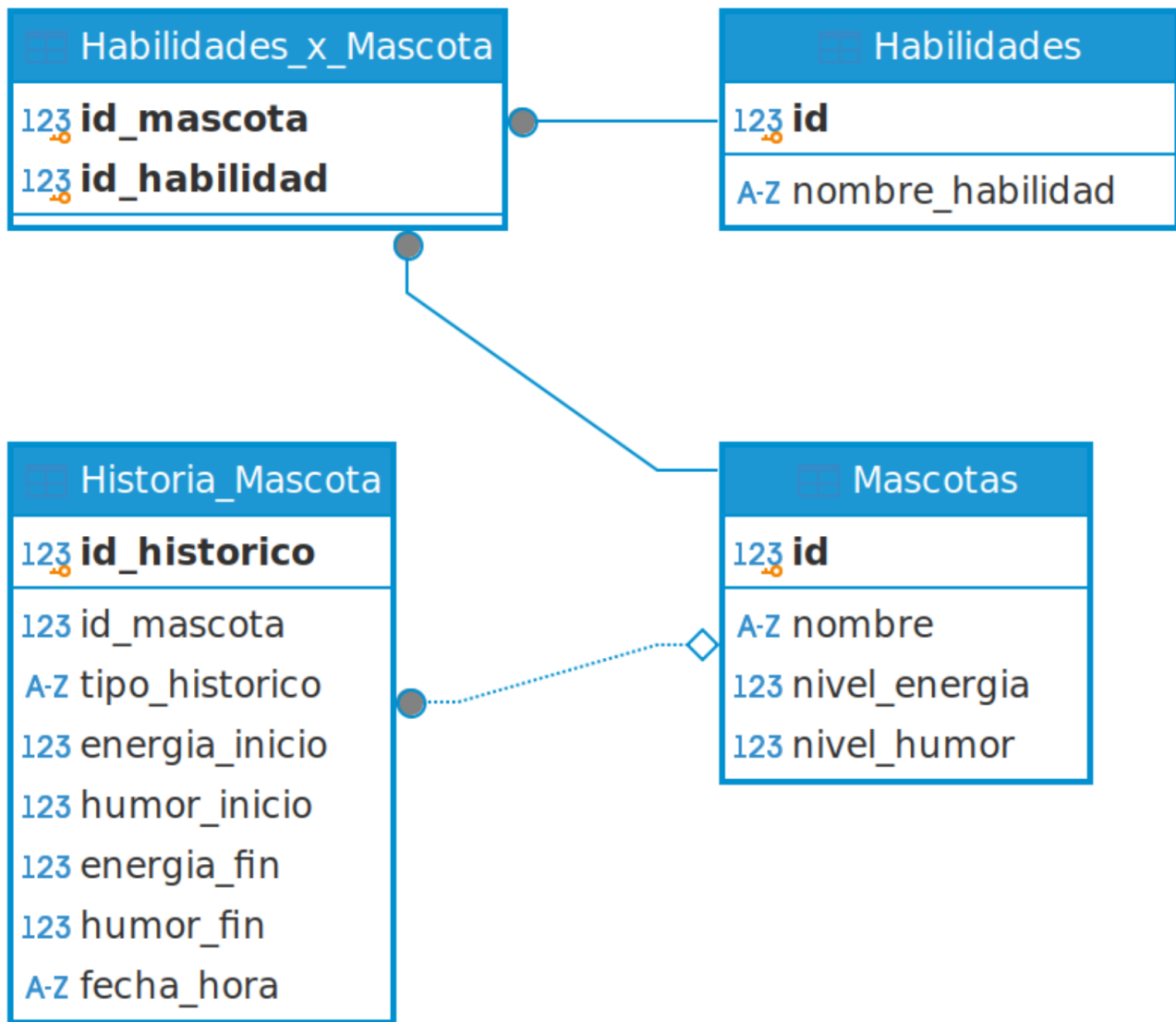
Cada mascota tiene un Nombre, un Nivel de energía: Entero del 0 al 100, un Nivel de humor: Entero del 1 al 5 (siendo 1 muy malo y 5 excelente) y una lista de habilidades (cada una identificada por una cadena), además de mantener una colección con el histórico de las acciones que realizó a lo largo de su vida (ingerir, ejercitar o demostrar una habilidad). Finalmente, para los fines de este parcial, aplican las siguientes reglas:

- Una mascota con energía 0 muere y no puede realizar más acciones.
- Una mascota puede ingerir alimentos. Cada ingesta incrementa en 10 unidades su energía y en 1 unidad su humor, si no hizo más de 3 ingestas consecutivas; a partir de la 3ra ingesta consecutiva únicamente incrementa 5 unidades su energía y no incrementa su humor. No puede realizar más de 5 ingestas consecutivas porque muere al hacerlo (su energía debe bajar a 0 en este caso)..
- Una mascota puede realizar actividad deportiva. Cada ejercitación decrementa en 10 unidades su energía.
- Una mascota puede demostrar una habilidad. Cada vez que se demuestra una habilidad, el nivel de humor aumenta en 1 unidad, siempre y cuando la habilidad que se le pide demostrar esté dentro de las habilidades que posee. No hay reglas adicionales para las habilidades.

Contexto del parcial

- Se provee un archivo CSV con los datos de las mascotas que deben procesarse. Cada mascota en el archivo está representada por una línea, donde cada atributo está separado por comas y donde las habilidades de la mascota están separadas por punto y coma (👉).
- Se provee una base de datos creada, con una estructura que debe ser respetada para la grabación de datos solicitada
- Se provee el esqueleto del proyecto, junto con el archivo persistence.xml para trabajar con JPA

Estructura de la base de datos



{border-effect=line}

Requerimientos

1. Deben crearse las clases necesarias para dar soporte al problema.
2. La clase Mascota (debe llamarse así) debe implementar los métodos que el alumno considere necesarios pero, además:
 1. `public float getEstadoFisico():` Retorna el cociente entre la cantidad de ejercitaciones y la cantidad de ingestas realizadas por la mascota durante toda su vida.
 2. `public boolean estaViva():` Retorna si la mascota está viva o no
 3. `public int getRachaFitness():` Retorna la mayor cantidad de ejercitaciones consecutivas que realizó la mascota.
 4. `public void demostrarHabilidad(String habilidad):` Simplemente imprime la cadena "Mirá como puedo <habilidad>!". Pero si la habilidad no está dentro de las habilidades de la mascota, arroja una `IllegalArgumentException`.
3. Si una mascota está muerta, cualquier intento de actividad debe resultar en una excepción no chequeada del tipo `MascotaMuertaException`.

4. Al iniciar el programa, debe levantar los datos de todas las mascotas del archivo .csv y deben mantenerse todas las mascotas en memoria, en la/las colecciones que considere necesarias. La clase que contiene a todas las Mascotas se debe llamar ServicioMascotas y debe tener, al menos, los siguientes métodos:
 1. public void guardarVivas(): Debe almacenar en la base de datos, utilizando JPA, todas las mascotas que están vivas, dentro de las mascotas que se tiene en memoria..
 2. public List<Mascota> getFelices(): Debe retornar desde la base de datos el listado de todas las Mascotas con un nivel de humor mayor a 3.
5. En base a las salidas esperadas del programa, se deben imprimir por consola:
 1. Listado de las mascotas vivas, junto a su estado físico y su racha fitness.
 2. Listado resultante del método "getFelices()" luego de haber grabado los datos en la base de datos.
 3. El alumno puede realizar las pruebas que quiera, puede hacerle realizar actividades a las mascotas que desee (variando su estado), siempre y cuando se impriman estos resultados por consola.

Recomendaciones y aclaraciones

- Se brindó el material con Proyecto y Base de datos para que el alumno tuviera una visión previa al día del parcial y que no haya sorpresas, la idea de la cátedra es que durante el parcial codifiquen lo que se pide y entreguen cada uno a su leal saber y entender.
- SOLO tienen que escribir código java y configurar la clase que implemente el método main en el plugin del jar para que el programa se pueda ejecutar con `java - jar` ya que hemos quitado todos los elementos de configuración del parcial para eliminar posibles problemas que consumen tiempo valioso en esta instancia
- Por esto:
 - Se espera un trabajo individual y de producción propia, SIN la utilización de bloques previos NI herramientas externas, por esto nos enfocamos en evaluar programación
 - No debe acceder a repositorios ni tampoco a internet y ni usar cualquier tipo de acceso fuera del equipo en el que trabaja puesto que hacerlo habilita al aplazo y análisis de posibles medidas disciplinarias mayores
 - No está permitido el uso IA para la escritura de código más allá de los plugins vistos en clase para trabajo con JPA o base de datos.

La cátedra de Backend les desea todo el Éxito en esta semana de Parcial!!!