

Reporte: Práctica 02 Sistema de Archivos vs. Bases de Datos

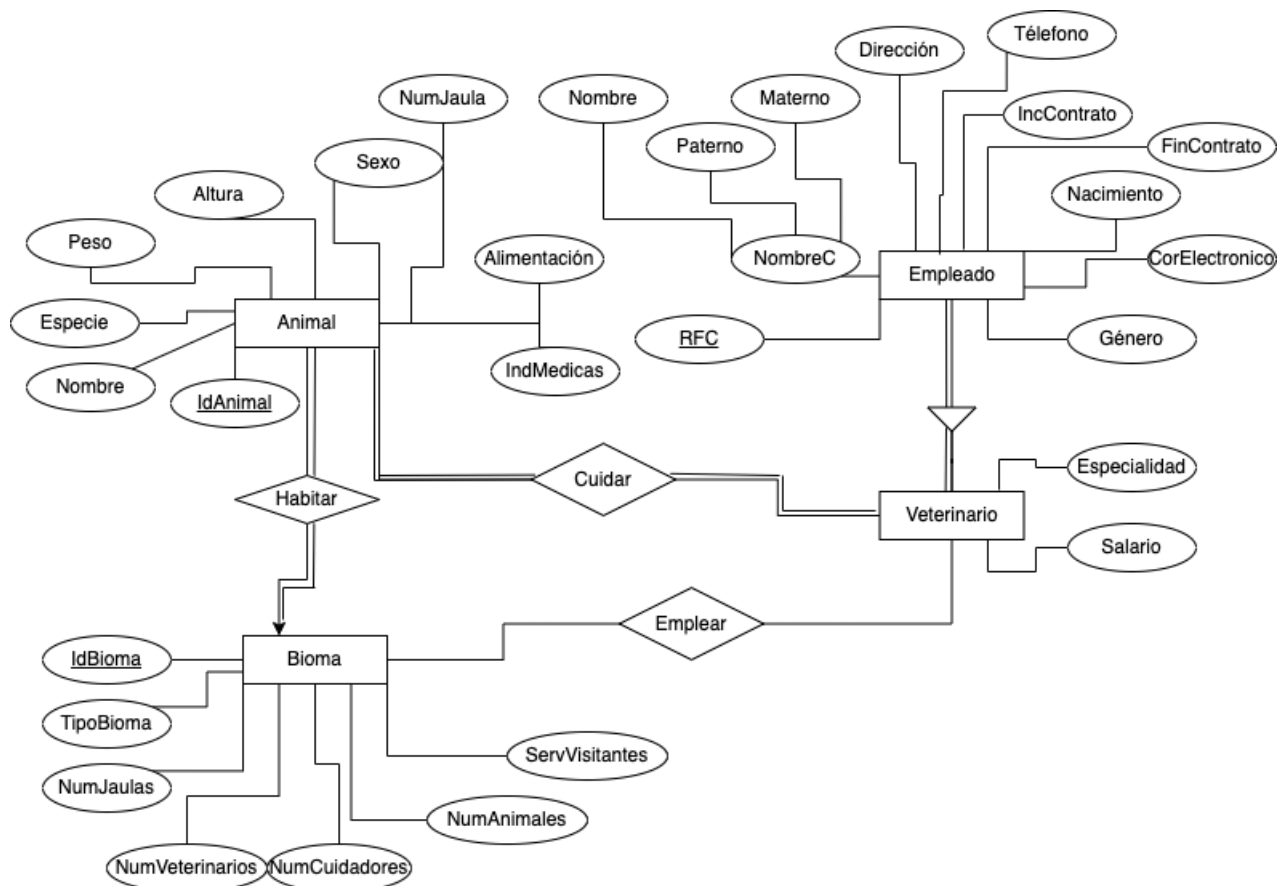
Análisis de requerimientos:

1 Enumerar los requerimientos candidato

- Almacenar información sobre los veterinarios, animales y biomas.
- Poblar la base de datos a través de la información recolectada en esta fase.
- Verificar los valores de las entidades.
- Agregar, consultar, editar y eliminar información de veterinarios, animales y biomas.
- Generar persistencia en archivos CSV, capaces de ser leídos y escritos.
- Tener bien organizados el sistema de archivos de cada entidad.
- Habilitar un menú de interacción para la persona usuaria.
- Recibir la llave de la entidad para poder recuperar la información completa de las entidades.
- Verificar que en los sitios donde sólo se reciban números no se ingresen valores no-numéricos.
- Implementar manejo de excepciones.

2 Compresión del contexto del Sistema

Para poder entender el contexto en el que vamos a desarrollar todo, elegimos trabajar con un diagrama de Entidad-Relación. Con la finalidad de que la petición en concreto de la práctica pueda cumplir las necesidades pedidas en esta entrega para el zoológico de Huitziltepec.



Elegimos representar todos los atributos listados en la práctica en estas tres entidades básicas para tener un funcionamiento sencillo de la información más pertinente, elegimos poner a empleado como un súperclase esperando que después esto funcione en términos de escalabilidad. Utilizamos participación total porque hasta ahora todo empleado es un veterinario.

3 Captura de requerimientos funcionales.

Aquí debemos revisar qué es lo que queremos que sea funcionalmente capaz de realizar nuestro proyecto.

En este caso vamos a requerir lo siguiente:

- a) Para facilitar el recuperar la información vamos a definir clases en Java que representen las entidades, los atributos serán a su vez atributos de la clase que nos faciliten modelar cada instancia de una entidad como un objeto. De manera que tenemos a definir 3 clases, correspondientes a Animal, Bioma y Veterinario. Pero notamos que Veterinario a su vez debe ser una subclase de Empleado, el cual posiblemente tenga más clases que hereden de la misma, así que definimos en total 4 clases que representan instancias contando la superclase Empleado.
- b) Vamos a generar un proxy que se encargue de gestionar el cómo se modifican, buscan y obtienen los datos guardados en un CSV. Éste a su vez se coordinará para mostrar en terminal una serie de opciones que permitan al usuario tener un mayor control respecto a cómo agregar, consultar, editar y eliminar indicando por números en terminal. Una vez seleccionada una opción, la misma terminal solicitará y dará un espacio para proporcionar los datos pertinentes para su opción.
- c) Los CSV se van a guardar en carpetas especialmente diseñadas para cada una de las tres entidades, el programa revisa las líneas para generar las coincidencias en caso de una búsqueda utilizando los ID propios de cada entidad. Habrá únicos archivos CSV generados por cada entidad, de manera que debemos ocupar un singleton para garantizarlo, si no existen; se crea pero si existe el archivo mantenemos esa unicidad del documento, esto tiene como objetivo evitar la redundancia de datos al menos a nivel de archivos. Lo cual puede ocasionar una gestión desastrosa del sistema.
- d) Los datos se verifican, de manera que no haya datos incorrectos, si bien esto se puede manejar por excepciones, también se da el caso de que el programa cuando sea proporcionado un dato incorrecto sea capaz de recuperarse y pedir en terminal una entrada correcta.
- e) Los métodos de búsqueda para las entidades necesariamente deben realizarse a través del identificador único que hemos definido para cada clase/entidad del sistema.

4 Captura de requerimientos no funcionales.

■ Requerimientos no funcionales asociados a requerimientos funcionales.

En esta práctica no se nos notificó de requerimientos no funcionales específicos al pequeño modelo que debíamos implementar a nivel código, funcional. Es decir, no hay especificaciones propias de la empresa o reglas específicas que deban tomarse en cuenta para gestionar el sistema o los datos que alteren la funcionalidad del sistema.

■ Requerimientos no funcionales asociados a requerimientos no funcionales.

Aquí se nos solicitó que el sistema tuviera coherencia cuidando el cómo abstraemos las entidades y que también el sistema fuera robusto, de manera que notificara cuando algo falló o pudiera recuperarse.

Preguntas:

1 Menciona 5 diferencias entre almacenar la información utilizando un sistema de archivos a almacenarla utilizando una base de datos.

■ Consultas y búsqueda:

- Sistema de archivos: Al buscar y acceder a datos específicos en un sistema de archivos puede ser más lento y menos eficiente, ya que generalmente se realiza a través de navegación en carpetas y búsqueda manual de archivos.
- Base de datos: Las bases de datos están diseñadas para consultas eficientes y permiten la indexación de datos, lo que acelera la búsqueda y recuperación de información.

■ Seguridad y control de acceso:

- Sistema de archivos: En un sistema de archivos, la gestión de permisos de usuario y grupos se realiza principalmente a nivel de archivos y carpetas, lo que puede ser menos detallado y más complejo de administrar.
- Base de datos: Las bases de datos ofrecen un control de acceso más granular y controlado a nivel de registros y campos, lo que permite una gestión más precisa de quién puede acceder y modificar los datos.

■ Redundancia de datos:

- Sistema de archivos: En un sistema de archivos, la redundancia de datos es común. Esto significa que los mismos datos pueden estar duplicados en varios archivos o ubicaciones. Cuando se actualiza un dato en un lugar, puede ser necesario actualizarlo en otros lugares manualmente, lo que aumenta el riesgo de inconsistencias y errores.
- Base de datos: Los sistemas de bases de datos aseguran que esta redundancia sea mínima, lo cual implica también un menor consumo de espacio en disco duro. Los sistemas de gestión de bases de datos (DBMS) utilizan técnicas de normalización para organizar los datos de manera eficiente y reducir la duplicación.

■ Escalabilidad:

- Sistema de archivos: La escalabilidad de un sistema de archivos puede ser limitada, especialmente cuando se trata de grandes cantidades de datos o acceso simultáneo de múltiples usuarios.
- Base de datos: Las bases de datos están diseñadas para escalar tanto vertical como horizontalmente, lo que significa que pueden manejar grandes volúmenes de datos y cargas de trabajo de manera más efectiva y escalable.

■ Consistencia de datos:

- Sistema de archivos: Los sistemas de archivos no ofrecen garantías de consistencia de datos ni control de concurrencia de forma predeterminada, lo que puede llevar a problemas de integridad de datos en entornos multiusuario.
- Base de datos: Las bases de datos están diseñadas para garantizar la integridad de los datos, con mecanismos de control de concurrencia y transacciones que aseguran que los datos permanezcan coherentes incluso en entornos de acceso concurrente.

2 Describe cual es mas conveniente utilizar (sistema de archivos o base de datos).

La elección de utilizar un sistema de archivos o una base de datos va a depender de las necesidades y requerimientos de lo que estemos haciendo. Para ambos enfoques tenemos ventajas y desventajas. A menudo, las aplicaciones modernas utilizan una combinación de ambos enfoques, almacenando archivos en el sistema de archivos y metadatos relacionados en una base de datos para un mejor rendimiento y gestión de datos.

Referencias:

[Enlace web referencia.](#)

[Enlace web referencia.](#)

[Enlace web referencia.](#)