



Nombre y Apellidos:

Instrucciones. La tarea constará de diferentes puntos que crearan un modelo de clases y un programa que desarrollaremos en la unidad siguiente.

La tarea se entregará en el aula virtual en un ZIP **tareaUnidad15NombreApellidos.Zip** conteniendo los ejercicios contestados. Crear un proyecto en eclipse **tareaUnidad15NombreApellidos** con el modelo que se ofrece en la práctica.

Criterios de evaluación

RAE 8. Utiliza bases de datos orientadas a objetos, analizando sus características y aplicando técnicas para mantener la persistencia de la información.

e) Se han programado aplicaciones que almacenen objetos en las bases de datos creadas. Indicador 82

f) Se han realizado programas para recuperar, actualizar y eliminar objetos de las bases de datos. Indicador 83

g) Se han realizado programas para almacenar y gestionar tipos de datos estructurados, compuestos y relacionados. Indicador 84

h) Se han realizado programas para almacenar y gestionar tipos de datos estructurados, compuestos y relacionados. Indicador 85

RAE 9. Gestiona información almacenada en bases de datos relacionales manteniendo la integridad y consistencia de los datos.

d) Se han creado programas para recuperar y mostrar información almacenada en bases de datos. Indicador 89

e) Se han efectuado borrados y modificaciones sobre la información almacenada. Indicador 90

f) Se han creado aplicaciones que ejecuten consultas sobre bases de datos. Indicador 91

g) Se han creado aplicaciones para posibilitar la gestión de información presente en bases de datos relacionales. Indicador 92



Nombre y Apellidos:

Detalles de la tarea de esta unidad.

Enunciado.

Tarea Unidad 15.

Introducción

En esta **tarea vamos a aplicar lo aprendido en la unidad e integrarlo en la tarea de la unidad 15. Vamos a migrar la aplicación obtenida en la unidad 14 de ficheros a una base de datos relacional y/o orientada a objetos.**

Condiciones de Entrega

Entregamos un zip que contendrá:

1. El proyecto **final**. El *proyecto completo funcionando*. El nombre del proyecto **tarea15NombreyApellidos**
2. **Pantallazos** probando cada ventana con las acciones que os pido, y los ficheros de informes. **El nombre del PDF tarea15NombreYApellidos.pdf.**

El nombre del **zip tarea15NombreYApellidos.zip**.

Tarea. Descripción.

Usaremos **lo ya construido hasta ahora en la Tarea 14. La idea es aprovechar todo lo construido hasta ahora y que nuestro objeto Proxy acceda a base de datos en lugar de a ficheros usando un conector jdbc como habéis visto en los apuntes. Llegados a este punto los alumnos deben ser**



Nombre y Apellidos:

autónomos para elaborar sus programas y **encontrar algunos detalles de implementación**. Damos recomendaciones, pero el alumno será el encargado de implementar los detalles y llegar a sus propias conclusiones.

Recomendaciones:

Se **recomienda crear una clase proxy proxyConexion** para la **conexión de base de datos**, y que **ese objeto use el patrón Singleton** para que la **conexión se habrá una sólo vez**. Mejor con **métodos estáticos**

Se **recomienda crear dos clases para la consulta y modificación de nuestros datos**. Estas clases **usaran la conexión a base de datos proporcionada por proxyConexion**. Al contrario que en **ficheros ahora podemos acceder a tablas y modificar una empresa o un trabajador individualmente**. Debéis crear:

1. Modificaremos **nuestra clase Proxy** para que **haga la conexión hacia la base de datos**. Para realizar la conexión a base de datos usar el **patrón Singleton** y reutilizar siempre la misma conexión.
2. Objeto **daoEmpresas**, que **modificará o consultará empresas o subsidiarias**. Con seis **operaciones**, **listaEmpresas**, **listaSubsidiarias**, **insertarEmpresa**, **borrarEmpresa**, **modificarEmpresa**, **leerEmpresa**.
3. Objeto **daoTrabajadores**, que **modificará Trabajadores**. Con **cinco operaciones**, **listaTrabajadores**, **insertarTrabajador**, **borrarTrabajador**, **modificarTrabajador**, **leerTrabajador**.

Elegid **los parámetros para cada método** teniendo en cuenta el **modelo de datos**. Por ejemplo, para **leerTrabajador** debéis proporcionar el **id del trabajador**. Para la **listaSubsidiarias** debes proporcionar el **id de la empresa matriz**. Estos son **los métodos base** si necesitáis más métodos añadidlos.

Parte I. Base de datos relacional (7 puntos)

Indicador 89, 90, 91, 92

Modelo de base de datos

Aplicad los conocimientos aprendidos en la asignatura de base de datos para generar un **modelo de base de datos**, y **crear una base de datos**



Nombre y Apellidos:

basándoos en el modelo de datos que usamos en el programa. Por ejemplo, si una subsidiaria tiene una lista de trabajadores, es una relación uno a muchos entre la tabla Empresa, y Trabajador. Construimos el modelo a partir de nuestras colecciones de datos en nuestro programa.

Bases de datos que podéis elegir:

1. **SQLite como en los apuntes**
2. **Oracle** que es con la que trabajáis en la asignatura de base de datos.
3. **MySQL mariadb de Xampp**

Consejos para manejo de datos en la aplicación

Opción A

Hasta ahora cargabamos todos los datos desde fichero a **ControladorPrincipal**. Usando base de datos no será necesario. Por ejemplo, en la ventana principal sólo necesitáis extraer las Empresas matriz de la base de datos. Pero en la ventana **VentanaEmpresa**, podéis volver a acceder a base de datos para extraer las subsidiarias de la empresa elegida.

En la ventana **VentanaEmpresaSubsidiaria**, para la empresa subsidiaria elegida podéis volver a acceder a la base de datos para obtener la lista de trabajadores.

Pero es más, con el fin de tener siempre los datos más recientes, se recomienda, que cuando accedo a **VentanaEmpresa**, volver a acceder a base de datos para volver a recuperar la Empresa matriz además de sus subsidiarias.

Otra opción más plausible incluso es, que cargéis todos los datos desde el principio en nuestro modelo es la **Opción B**.

Opción B (La que os recomiendo)



Nombre y Apellidos:

Tras el acceso a **ControladorPrincipal** cargar todos los datos como antes. Recuperar de **base de datos** todas las **empresas matrices**, para cada una de ellas sus **subsidiarias**, y para cada subsidiaria su **listado de trabajadores**. Que la recuperación de los datos se haga igual que en la tarea 10, completa.

Pero cada vez que accedáis a un nivel inferior, por ejemplo, a **VentanaEmpresa**, volved a recuperar la información completa de esa empresa y la volvéis a colocar en **mapEmpresas**, sustituyendo la versión anterior. Igual cuando accedemos a **trabajadores**. De esta manera siempre trabajamos con los datos más actualizados.

Para que las **sustituciones de los datos nuevos con los antiguos** sean más sencillas. Mi recomendación es que añadáis un método equal a **Empresa** y **Trabajador**. Ese método equals debe **comparar objetos por el id**, porque el id nunca cambia.

Tarea Parte II. Migrar a una Neodatis (3 puntos)

Indicador 82, 83, 84, 85

Partiendo de la base de datos relacional migrar a la base de datos neodatis.

Cread las clases siguientes clases para manejo de base de datos para con los siguientes métodos

1. **NeoDatisProxy**, para abrir y cerra la base de datos y mantener una instancia de la misma.
2. Clase **daoEmpresasNeoDatis**, que modificará o consultará **empresas o subsidiarias**. Con seis **operaciones**, **listaEmpresas**, **listaSubsidiarias**, **insertarEmpresa**, **borrarEmpresa**, **modificarEmpresa**, **leerEmpresa**.
3. Clase **daoTrabajadoresNeoDatis**, que modificará **Trabajadores**. Con cinco **operaciones**, **listaTrabajadores**, **insertarTrabajador**, **borrarTrabajador**, **modificarTrabajador**, **leerTrabajador**.