

PRÁCTICA 2: INTRODUCCIÓN A JDBC, DAO Y DTO

Informe práctica 2 de la asignatura de
Programación Web

Miembros del grupo:

Sergio Alcántara Avilés: i12alavs

Manuel Antonio Muriel: i32anmum

Carla Bravo Maestre: i12brmac

Alejandro Dueñas Ruiz: i12durua

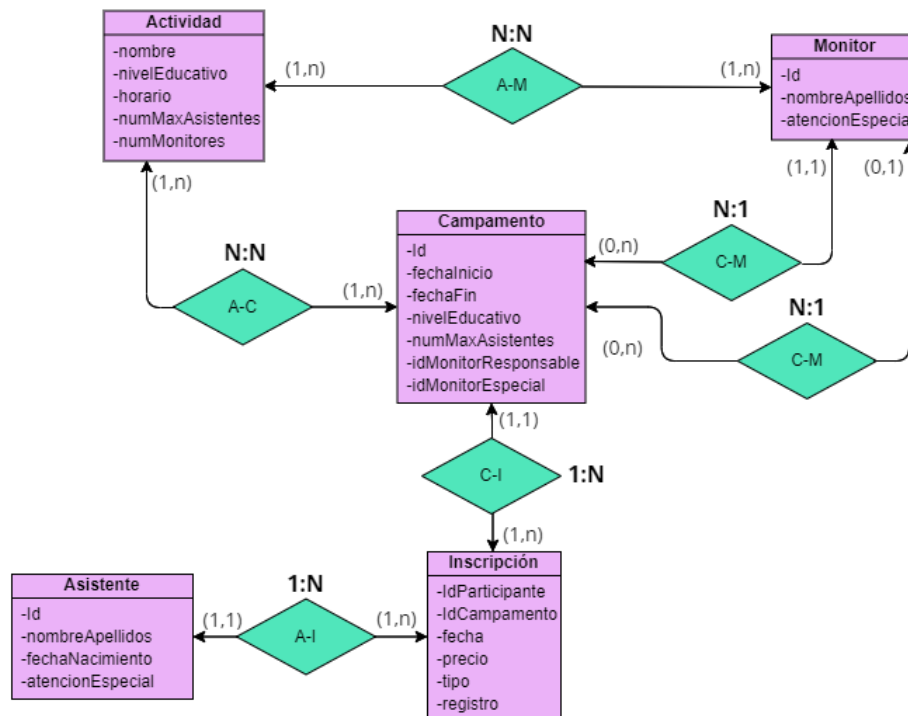


INDICE

Base de Datos:	2
Diseño en capas:	2
• Capa de datos:	2
• Capa de Negocio:	3
• Capa de visualización:	3
Webgrafía:	4

Base de Datos:

Para la creación de la base de datos hemos seguido el siguiente modelo:



Inicialmente entre campamento y monitor teníamos una única relación y no sabíamos exactamente como representar los monitores responsable y especial, finalmente los representamos como un atributo de campamento y las relaciones correspondientes.

Respecto a las relaciones N:N, hemos necesitado usar tablas auxiliares con claves primarias compuestas.

Diseño en capas:

El proyecto esta diseñado siguiendo una división en tres capas, datos, negocio y visualización:

- **Capa de datos:**

En esta capa hemos usado múltiples objetos DAO, uno por cada tabla, excepto para las tablas de las relaciones N:N, las funciones que necesitamos para acceder a estas están separadas entre campamentoDAO, monitorDAO y actividadDAO, pensamos crear objetos DAO específicos para estas tablas, pero al final por tiempo, hemos decidido dejarlo así.

Además de las sentencias para la inserción y modificación de datos, en los diferentes DAO tenemos funciones con consultas que nos ayudan a cumplir los requisitos del problema. Como pueden ser, conocer el numero de monitores asignados a una actividad, el numero de asistentes que hay inscritos en un campamento, o algo tan simple como saber si ya hay un asistente con el mismo id en la tabla.

Si queremos saber un atributo concreto de una tupla de alguna tabla, el DAO tiene una función que nos devuelve todos los atributos dado un identificador y los pasa al DTO que será recibido por el gestor, este ya se encargará de manejar la información que necesite de ese objeto en concreto.

- Capa de Negocio:

En esta capa tenemos los diferentes DTO, que serán usados simplemente para transferir datos entre las capas. Estos, no cambian mucho su estructura respecto a la de la anterior práctica, ahora solo tienen las variables private y los métodos get y set, junto a un método toString.

Para las inscripciones hemos decidido eliminar las factorías y dejar un único DTO de inscripción que ahora tendrá los atributos tipo (con valores Completa o Parcial) y registro (con valores Temprano o Tardío), para estos tipos de atributos, hemos usado enumerados, ya que son muy sencillos de utilizar con los métodos name() y valueOf().

Los gestores, no varían mucho de los que ya teníamos implementados, ahora simplemente llaman a las funciones de los DAO y con la información de estas, realizan las comprobaciones pertinentes en base a los requisitos del problema.

- Capa de visualización:

Desde esta capa realizamos todas las entradas y salidas por pantalla, a excepción de los mensajes de error de los controles de errores try-catch que se encuentran en las otras capas.

Hemos separada esta capa en 4 archivos, un programa principal y 3 interfaces, una para cada gestor. El principal mostrará un menú, y llamará a los otros tres programas, estos tienen sus propios menús, a excepción de la interfaz de campamento que además tiene un menú extra para gestionar las actividades y otro para los campamentos, estas separaciones son meramente estéticas y de organización.

Webgrafía:

<https://docs.oracle.com/javase/8/docs/api/java/sql/ResultSet.html>

<https://www.java67.com/2012/08/string-to-enum-in-java-conversion.html>

<https://online.visual-paradigm.com/es/login.jsp?t=diagrams>

<https://es.stackoverflow.com/questions/229152/scanner-nextline-error-salta-una-linea>

<https://www.tutorialesprogramacionya.com/postgresqlya/temarios/descripcion.php?cod=193&punto=35&inicio=>

<https://www.java67.com/2012/08/string-to-enum-in-java-conversion.html>