

Informe de prácticas PW

Autores: Ángel Quintero Montes, David Olmo Cejudo,
Diego Tabas Costa, Francisco Moreno Cano

Índice

Práctica 2	1
Ejercicio 1	1
Ejercicio 2	2
Extras	2
Fuentes	2
Credenciales Base de Datos	2
Corrección de errores de la Práctica 1 en detalle	3

Práctica 2

Ejercicio 1

Viendo cómo habíamos acabado la práctica anterior al comienzo de esta tuvimos que estar una semana sólo para corregir errores y hacer un poco de preparación para que en esta práctica empezáramos con una idea muchísimo más clara de lo que hace falta hacer en cada campo, ya sea programar, diseñar, o en el caso de esta memoria, mantener un registro de cómo estamos llevando los problemas y complicaciones que podrían ocurrir por el camino.

Para comenzar simplemente dibujamos un par de modelos iniciales para ir haciéndonos una idea de como formar la base de datos, tras un par de iteraciones simplemente teníamos que decidir qué campos harían de claves primarias. Esto, que en un principio se presentó como una tarea simple con mínima complejidad, se convirtió en uno de los primeros problemas de nuestra base de datos.

La clase Reserva, una de las clases que más problemas ha producido consistentemente, de nuevo produce un nuevo dilema, ¿Cómo se identifica una Reserva?, por el correo podría haber varias reservas, por la fecha podríamos tener muchísimas reservas, ¿Deberíamos poner varios campos?, eso haría más compleja la base de datos y aún así nos dejaría con el problema. Esto no se iba a resolver eligiendo un campo de los que ya teníamos, por esto sólo quedaba una opción, bastante obvia pero aún así se nos estaba escapando ante nuestra actitud más conservadora respecto a los campos de nuestras clases, simplemente había que añadir un id, un id identificaría varias reservas ante una misma fecha, un mismo correo e incluso misma pista, dificultad y precio. Con el problema más complicado teniendo una de las soluciones más obvias nuestra confianza bajó ligeramente aunque mientras seguíamos con la base de datos sólo aparecieron problemas más básicos y simples como las clases Kart y Pista que estaban bastante relacionadas así que tuvimos que añadir una foreign key para Kart ya que no podía ser completamente independiente de pista.

Ejercicio 2

Tras mirar un poco la presentación de la práctica vimos que los DTO realmente no varían muchísimo de cómo teníamos pensadas las clases básicas así que fue más bien un traslado de variables. Aún así teníamos que hacer un traslado completo de toda la práctica 1, eso junto a una organización de archivos nueva demostró ser un trabajo simple pero largo por lo que estuvimos unos días para ver cómo se quedaría finalmente.

Por culpa de todos los cambios de archivos de repente comenzaron a salir algunos problemas, muchos teniendo soluciones bastante simples. Por desgracia no todo podría ser tan fácil, nos vimos de repente con un fallo que cambia completamente como habíamos pensado la estructura de nuestro programa, al guardar las cosas en ficheros en la práctica 1 solamente necesitábamos acceder a ellos al ejecutar el programa y guardar al final, ahora al tener una base de datos podríamos tener varias instancias accediendo al mismo tiempo, lo cual hace que necesitemos guardar cada vez que terminemos de modificar cualquier cosa.

Extras

Estando en octubre ya empezaron a aparecer varias entregas importantes, algún parcial y con el repentino cambio de temperaturas un par de resfriados así que el trabajo fue complicado de llevar consistentemente, por ello tuvimos que separarlo de nuevo de una manera caótica, teniendo a personas trabajando muchísimo en la primera parte, de manera que cuando llegamos al final ya no tenían nada que hacer. Aunque haya sido una forma fácil de no chocar a la hora de modificar archivos deja a algunas personas con mucho trabajo al final, que si tuvieran algún trabajo más podría complicar el desarrollo.

Finalmente incluso con toda esta problemática, el sentimiento general fue más cómodo que en la práctica 1, ya nos conocíamos mejor y la base del trabajo estaba hecha, por lo que si siguiéramos esta curva la siguiente práctica sería mucho más manejable, pero eso se tendrá que ver.

Fuentes

- Recursos digitales en Moodle.com distribuidos por el encargado de la asignatura.
 - Autores: Dra. Aurora Ramírez Quesada y el Dr. Jose Raúl Romero Salguero.
 - Enlace: [Práctica 2 - Programación en JDBC, DAO, DTO](#)

Credenciales Base de Datos

- **Usuario:** i02mocaf
- **Contraseña:** pw2223

Corrección de errores de la Práctica 1 en detalle

Un anexo a nuestra memoria con los errores marcados en la práctica 1 junto a las soluciones para una visión a nuestro razonamiento. Al ser algo extra hemos decidido ponerla en una 3a página para que no le quite espacio al resumen completo de los errores de la práctica

En la bibliografía hubiera estado bien indicar los autores, fechas, URLs, etc.	Aun tener una bibliografía escasa hemos dado toda la información accesible del documento consultado
En la memoria os ha sobrado media página que podríais haber aprovechado para contar algo más de vuestro trabajo.	En esta memoria hemos contado todos los problemas de manera más precisa, consiguiendo incluso que el primer ejercicio se alargara para ocupar una página.
En el fichero de propiedades veo algunas propiedades extrañas (pista1, pista2, ...), probablemente derivado del hecho de cómo habéis trabajado con los ficheros. Este fichero está pensado más bien para propiedades estáticas, conocidas, etc., más que para almacén dinámico. Además, por el código veo referencias directas a los nombres de los ficheros de entrada/salida en lugar de leer su propiedad correspondiente.	Como en esta práctica no usamos el fichero propiedades no necesitamos corregir este fallo como tal, pero aún así lo tomamos en cuenta en caso de que necesitáramos hacer cosas similares en un futuro
La entrada/salida por teclado/pantalla es correcta pero quizá los gestores abusan de ella, en lugar de estar en el main o sus clases auxiliares. La navegabilidad la veo mejorable ya que, en lugar de mostrar todas las funcionalidades, podría haber un menú previo más pequeño que derive a cada uno de los grandes grupos de funcionalidades.	Hemos quitado la entrada por teclado de la mayoría de funciones del gestor pistas pero en otros como el gestor reservas o usuarios hemos decidido dejarlas
Los enumerados deben ir en ficheros aparte.	Fueron separados antes de hacer toda la preparación para la práctica 2 para que no se quedaran olvidados
Veo que habéis metido en la factoría de bono un array con reservas. Conceptualmente una factoría de reservas debería devolver una reserva. Para este problema, quizá la gestión de los bonos (asociar bono a reservas de dicho bono) se debería hacer en una capa superior fuera de la factoría, probablemente con una clase bono y apoyándonos en el gestor de reservas. El resto de la factoría tiene buena pinta.	En esta práctica no usamos el arrays de la factoría si no que usamos un DTO y un DAO de Bono para usar todas sus funciones
Quizá deberíais haber usado algún paquete más para organizar vuestras clases más claramente.	En esta práctica hemos dividido los archivos en 3 paquetes con nombres más representativos sobre la función que tienen en nuestro programa por lo que se hacía cómodo buscar los archivos
Algún comentario más hubiera sido bienvenido en el código, si bien destaco vuestro excelente Javadoc.	Al realizar un Javadoc tan completo y al intentar hacer uso de unas buenas prácticas de programación, el código se queda correctamente explicado sin la necesidad del uso de comentarios extra