

(P) Programación orientada a objetos con acceso a base de datos

Caso práctico

El centro excursionista Senderos y Montañas se dedica a organizar excursiones para sus socios y necesita implementar una aplicación para gestionar sus actividades.

Nuestra empresa ha sido seleccionada para desarrollar una aplicación de escritorio basada en Java para gestionar los socios, así como las inscripciones a las excursiones que se inscriben los socios.

Los datos que se gestionaran en la aplicación se dividen en excursiones, socios, seguros, federaciones e inscripciones.

Excursiones

las excursiones pueden ser de uno o varios días y cada excursión tiene su precio de inscripción, así pues, los datos básicos de las excursiones son un código alfanumérico, descripción, fecha, número de días y el precio de la inscripción.

Socios

Los socios pueden ser federados, estándares o infantil. Los datos que se almacenarán de cada socio serán el número de socio y el nombre. Existen tres tipos de socios:

- Estándar. Se debe que guardar también el Nif y el seguro contratado para las excursiones, el cual es obligatorio.
- Federado. Los socios federados tienen su propio seguro de la Federación y por lo tanto no será necesario que lo paguen aparte para cada excursión. Además, los socios federados tendrán un descuento del 5% en la cuota mensual del centro excursionista y el 10% en el precio de las excursiones. Así pues, para los socios federados se debe guardar también el Nif y la federación.
- Infantil. Los socios infantiles tendrán un descuento del 50% de la cuota mensual i no tendrán seguro ya que están cubiertos por el seguro del padre o de la madre, por lo tanto, se deberá guardar el número de socio de alguno de ellos

Seguros

Los datos básicos de los seguros son un tipo y un precio. Solo hay dos tipos de seguro Básico y Completo.

Federaciones

Los datos básicos de las federaciones son un código alfanumérico y un nombre.

Inscripciones

Para simplificar la aplicación se considerará que un socio debe realizar una inscripción para cada excursión que quiera realizar.

Los datos necesarios para cada inscripción son el número de inscripción, el socio y la excursión.

Una inscripción solo se puede cancelar si se realiza en una fecha estrictamente anterior de la que se realiza la excursión.

Requisitos funcionales:

Se pretende crear un sistema de información que modele el centro excursionista que recoja las especificaciones que se han indicado en el apartado anterior siguiendo un paradigma de programación orientada a objetos.

Para simplificar la aplicación, los datos de los tipos de seguros y de las federaciones no los deberán introducir los usuarios por teclado, sino que se precargarán dentro de la aplicación.

Se contemplará que la cuota base mensual como socio del centro excursionista es de 10€, las excursiones pueden ser de uno o varios días y cada excursión tiene su precio de inscripción.

La aplicación debe implementar un menú de opciones con las funcionalidades siguientes:

1. Gestión Excursiones:
 - 1.1. Añadir Excursión
 - 1.2. Mostrar Excursiones con filtro entre fechas
2. Gestión de Socios
 - 2.1. Añadir Socio Estándar
 - 2.2. Modificar tipo de seguro de un socio estándar
 - 2.3. Añadir Socio Federado
 - 2.4. Añadir Socio Infantil
 - 2.5. Eliminar socio.

Un socio solo se puede eliminar si no se ha inscrito nunca a una excursión.
 - 2.6. Mostrar Socios. (Todos o por tipo de socio).
 - 2.7. Mostrar Factura mensual por socios.

El total a pagar en la factura mensual de un determinado socio se calculará como la cuota base mensual más las excursiones que haya realizado.

Para el cálculo de la cuota mensual, en el caso de los socios estándares será la cuota íntegra, los socios infantiles tendrán un 50% de descuento de la cuota y los federados un 5% de descuento.

Para el cálculo del precio de las excursiones, en el caso de los socios estándares será el precio fijo de las excursiones más el precio del seguro contratado, en el caso de los socios federados al precio de las excursiones no hay que sumarle ningún seguro y se debe aplicar el 10%

de descuento y en el caso de los socios infantiles el precio es el precio de la excursión sin cargos ni descuentos.

3. Gestión de Inscripciones

3.1. Añadir Inscripción. En caso de que, al pedir los datos de la inscripción a una excursión se tratase de un nuevo socio, se deben pedir los datos del nuevo socio y se añadirá a la lista de socios. La excursión debe existir en el momento de realizar la inscripción.

3.2. Eliminar Inscripción. Una excursión puede ser borrada únicamente si la fecha es **anterior** a la fecha de la excursión.

3.3. Mostrar inscripciones con las opciones de filtrar por socio y/o fechas.

De cada inscripción se mostrará, el número de socio, el nombre, fecha de la excursión, la descripción y el importe con los cargos o descuentos aplicados.

FASE 1: Diseño del modelo estático. Es en esta fase se realizarán los diagramas de clases y de casos de uso a partir de los requisitos funcionales del caso práctico.

FASE 2: Implementación del modelo estático de clases en lenguaje Java. En esta fase se deberá implementar la mayor parte de la lógica de la aplicación aplicando el patrón MVC. La interacción con la aplicación será en modo consola almacenando los datos en estructuras dinámicas, además se debe tener en cuenta la gestión correcta de excepciones.

FASE 3: En esta fase se utilizará JDBC para realizar todas las operaciones necesarias con la BD. Se utiliza el SGBD MySQL para realizar la persistencia. Se tiene que aplicar el patrón DAO, transacciones y procedimientos almacenados.

FASE 4: En esta fase se cambiará el tipo de la programación de la persistencia, implementando la aplicación con mapeo ORM.

FASE 5: En esta fase se implementará la aplicación usando una interfaz gráfica mediante JavaFX.