TRABAJO PRÁCTICO INTEGRADOR

Materia: Programación 2

Alumno: Andrés Sánchez Albertti

Profesor: Carlos Arroyo

INFORME TÉCNICO

Estructura del proyecto  
Paquetes:

org.universidad

org.universidad.gestores

org.universidad.interfaces

org.universidad.recursos

Explicación general:  
El sistema está estructurado en 4 paquetes: recursos, gestores, excepciones, interfases. Está diseñado con el objetivo de gestionar recursos académicos de Como funcionalidades principales tiene:

* Listar registros
* Evaluar Registros
* Ordenar Registros
* Filtrar Registros
* Emitir reportes en archivos .txt

Como funcionalidades secundarios tiene:

* Añadir Recursos
* Eliminar Recursos

El flujo del programa consiste en realizar la carga de recursos (En este caso por hardcodeo, pero podría conectarse a una base de datos)

En principio se inicializa un evaluador el cual contiene una lista de recursos (vacía)

luego cada instancia de evaluador agrega los recursos correspondientes a su lista. Luego a su vez cada vez que se inicializa un evaluador, dentro de la instancia de evaluador se inicializa una instancia de la clase GestorRecursos la cual contiene los métodos para procesar los registros.

CLASES PRINCIPALES

Main

Ubicación: org.universidad

Función: Punto de entrada del programa. Crea instancias de Evaluador, Articulo, Libro, y TrabajoInvestigacion. Asigna categorías y recursos a los evaluadores, y muestra y genera reportes de estadísticas.

Evaluador

Ubicación: org.universidad.gestores

Función: Representa a un evaluador que puede tener una lista de recursos académicos. Contiene métodos para agregar y eliminar recursos, y obtener el gestor de recursos asociado. En este caso tomé la decisión de instanciar la clase GestorRecursos dentro de la clase Evaluador ya que entiendo que existe una relación 1 a 1 entre el Evaluador y el Gestor de recursos que procesará la lista de registros de Evaluador. Esto fue disñado de esta forma dado que facilita y ordena mejor el proceso de organizar los datos, a diferencia de la elección de aislar la clase GestorRecursos que fue una de las opciones que tomé en consideración

GestorRecursos

Ubicación: org.universidad.gestores

Función: Gestiona una lista de RecursoAcademico. Proporciona métodos para ordenar, filtrar, mostrar recursos, calcular porcentajes por categoría y generar reportes de estadísticas. En el caso de esta clase, tenía la opción de instanciarla en la clase recursos ya que cada recurso podría haber tenido su instancia de GestorRecursos pero dado que los métodos que requería esta clase necesitaban procesar datos de colecciones, me pareció conveniente tomar la decisión de instanciarla dentro de la clase Evaluador.

RecursoAcademico

Ubicación: org.universidad.recursos

Función: Clase abstracta que representa un recurso académico. Contiene atributos comunes como id, titulo, fechaCreacion, autor, categoria, cantidadCitas, y evaluador. Define métodos abstractos que deben ser implementados por las subclases.

Esta clase está declarada de forma abstracta. Esto significa que no puede instanciarse de forma directa sino que sus atributos estarán presentes en la instanciacion de las clases hijas. En este caso Realicé algunos agregados a la clase como la cantidad de Citas que ha obtenido el recurso con el objetivo de usarlo para calcular el puntaje y la relevancia del articulo. Me pareció conveniente agregarlo a esta clase para mantener la objetividad dentro del calculo de la relevancia

Articulo

Ubicación: org.universidad.recursos

Función: Subclase de RecursoAcademico que representa un artículo académico.

Implementa métodos para calcular relevancia, mostrar detalles, y realizar evaluaciones.

Libro

Ubicación: org.universidad.recursos

Función: Subclase de RecursoAcademico que representa un libro. Implementa métodos para calcular relevancia, mostrar detalles, y realizar evaluaciones. Además, contiene atributos específicos como numPaginas, editorial, y libroDigital.

TrabajoInvestigacion

Ubicación: org.universidad.recursos

Función: Subclase de RecursoAcademico que representa un trabajo de investigación. Implementa métodos para calcular relevancia, mostrar detalles, y realizar evaluaciones. Contiene atributos específicos como autores y lineaInvestigacion.

INTERFACES

Clasificable

Ubicación: org.universidad.interfaces

Función: Define métodos para asignar y obtener categorías de clasificación.

Evaluable

Ubicación: org.universidad.interfaces

Función: Define métodos para obtener puntaje y realizar evaluaciones.

FiltroRecurso

Ubicación: org.universidad.interfaces

Función: Interfaz funcional que define un método para evaluar si un recurso cumple con un criterio específico.

Resumen de Funcionalidades

Creación de Evaluadores y Recursos: Se crean instancias de Evaluador, Articulo, Libro, y TrabajoInvestigacion en la clase Main.

Asignación de Categorías: Los recursos pueden ser clasificados en diferentes categorías.

Gestión de Recursos: GestorRecursos proporciona métodos para ordenar, filtrar, mostrar recursos, calcular porcentajes por categoría y generar reportes de estadísticas.

Evaluación de Recursos: Los recursos pueden ser evaluados y se puede calcular su relevancia y puntaje.

Esta estructura permite una gestión organizada y extensible de los recursos académicos y sus evaluaciones.