

Programación Orientada a Objetos

Proyecto Primer Parcial

I PAO 2024

Aplicación para publicación de artículos científicos en una revista

Descripción General:

El objetivo de este proyecto es desarrollar una aplicación para gestionar la publicación de artículos en una revista de investigación. La aplicación permitirá a los autores someter sus escritos para ser revisados y publicados. Además, se gestionará el proceso de revisión, asignación de revisores, respuesta de revisores y el resultado final de la publicación, incluyendo el envío de correos electrónicos a los involucrados.

Requisitos Específicos:

El sistema puede ser usado por autores (quienes escriben artículos científicos), revisores (quienes validan los artículos científicos y su redacción), y el editor de la revista (de él depende si el artículo será publicado en la revista o no).

Autor:

Un autor puede subir (someter) un artículo para revisión. Asumir que un artículo tiene un solo autor.

De cada autor se conocerá su código de identificación único (que genera el sistema), nombre, apellido, correo electrónico, institución y campo de investigación.

Los artículos sometidos deben incluir un código único que genera el sistema, y los datos que ingresa el autor sobre título, resumen, contenido, y palabras clave.

Los datos propios de los autores y sus artículos deben almacenarse en un archivo investigadores.txt y articulos.txt.

Revisor:

Un revisor es asignado para revisar los artículos sometidos. De cada revisor se conocerá, nombre, apellido, correo electrónico, especialidad, y número de artículos que ha revisado durante su etapa como revisor.

Un revisor puede aceptar o rechazar la asignación de un artículo. Los datos de los revisores deben almacenarse en un archivo revisores.txt.

Editor:

Basado en los comentarios de los revisores y sus decisiones, el editor toma la decisión final sobre la publicación de los manuscritos. De él depende que se publique el artículo o no. De un editor se conoce su nombre, apellido, correo electrónico, y nombre del Journal para el que trabaja.

Puede ocurrir que de los dos revisores asignados para una revisión, uno acepte el manuscrito (artículo) y otro no, En estos casos es fundamental la decisión del editor.

Acceso al sistema:

Para este proyecto, los datos de los usuarios del sistema ya estarán registrados en el archivo usuarios.txt. **NO SE CREARÁN NUEVOS USUARIOS (usted agregue datos manualmente en el archivo para que pueda probar la aplicación).** Los usuarios pueden ser autores, revisores o editor. **En el caso de editores y revisores, deben iniciar sesión en la aplicación, mientras que los autores no lo necesitan.** Cada usuario tiene un user de acceso, una contraseña, un rol (E: editor, R: revisor, y puede agregar otros campos que necesite)

Al ejecutar la aplicación, se mostrarán las opciones de Someter artículo e Iniciar sesión.

Cuando un usuario revisor o editor inicia sesión se mostrará su tarea a realizar:

Revisor: Revisión de artículo

Editor: Registro de decisión final sobre artículo

Gestión de someter artículo:

Cuando un autor quiere someter un artículo primero debe registrar sus datos en la aplicación (nombre, apellido, etc.) y los datos deben guardarse en el archivo correspondiente.

Posteriormente, debe ingresar los datos de su artículo y los datos deben guardarse en el archivo correspondiente.

A continuación puede enviar el artículo a revisión. Al aceptar esa opción, inicia la gestión de revisión.

Gestión de Revisión:

Cuando un autor somete un artículo, se debe asignar automáticamente a dos revisores de la lista de revisores. La aplicación debe enviar un correo a los revisores indicando que se les ha asignado tal artículo (en el contenido del mail incluir los datos del artículo).

Asumir que se adjunta el pdf también para la revisión (esto no será parte de la aplicación, solo asumir que está ocurriendo).

Luego de revisar el paper, los revisores ingresar a la aplicación con su usuario y contraseña, para proporcionar comentarios y una decisión (aceptar o rechazar).

Los comentarios y la decisión de los revisores deben almacenarse en el archivo correspondiente.

Resultado Final:

El editor puede acceder a la aplicación con su usuario y contraseña para ingresar su decisión final sobre el artículo.

Para ello, el editor debe ingresar el código del artículo y podrá conocer los comentarios ingresados por los revisores.

Basado en las decisiones de los revisores, el editor tomará una decisión final sobre la publicación del artículo.

El investigador será notificado del resultado final por correo electrónico y se incluirán los comentarios de los revisores.

Los datos del proceso de revisión deben almacenarse en un archivo revisiones.txt.

Requerimientos de desarrollo de la aplicación:

Cada clase debe encapsular sus atributos utilizando modificadores de acceso apropiados.

Al iniciar la aplicación, enseguida de forma automática debe cargar la lista de usuarios registrados en el sistema (editor y revisores en la misma lista) para que cuando algún usuario inicie sesión en la aplicación se pueda comprobar sus credenciales (user y contraseña) y se muestra la opción de tarea adecuada de acuerdo al usuario que inició sesión.

Encapsulamiento:

Utilizar getters y setters para el acceso a los atributos privados de las clases. Implementar diferentes tipos de restricción con modificadores de acceso que permitan encapsular los datos de acuerdo a su naturaleza.

Abstracción:

Definir la clase Usuario como abstracta con un método abstracto para generar correo electrónico que funciona diferente dependiendo del tipo de usuario que la implemente. También el método de decidir sobre artículo que funciona diferente dependiendo del usuario.

Herencia:

La clase Usuario será la clase padre de la cual heredarán Editor y Revisor. El Autor no necesita heredar de usuario porque no tendrá nombre de usuario y contraseña.

Asegurarse de que las subclases implementen y/o sobrescriban métodos necesarios.

Polimorfismo:

Utilizar polimorfismo para manejar la lista de Usuarios que puedan contener tanto editor como revisores.

Cuando se inicie sesión debe verificar si el tipo de usuario que usando instance of para mostrar las opciones de tarea adecuada)

Sobrescritura y Sobrecarga:

Sobrescribir métodos como toString para personalizar la representación de los objetos.

También debe sobrescribir los métodos para generar correo electrónico y para ingresar la decisión sobre el artículo.

Incluya sobrecarga en su proyecto que no sea la sobrecarga de constructores. Queda a disposición de cada grupo de trabajo de estudiantes, donde y cómo incluirá sobrecarga.

Persistencia de Datos:

Almacenar y recuperar datos desde archivos de texto (usuarios.txt, autores.txt, articulos.txt, revisores.txt, revisiones.txt, editores.txt). En este primer parcial no se enseña manejo de archivos. Sin embargo, los estudiantes tienen conocimientos adquiridos sobre la persistencia de datos en archivos. Por lo tanto, se les proporcionará los métodos necesarios para leer y escribir en archivos. Los estudiantes solo deben entender cómo usar estos métodos (abstracción).

Correo Electrónico:

El ayudante de la materia hará ejemplos en sus clases sobre implementar esta funcionalidad. Debe asistir a ayudantías o investigar sobre cómo hacerlo.

Criterios de evaluación:

Funcionalidad (40 %)

- Se valorará la apariencia del proyecto, sin que esto implique que el estudiante deba utilizar recursos adicionales a los aprendidos en clase. (El proyecto es en consola)
- Se verificará el cumplimiento de toda la funcionalidad requerida, así como las correctas validaciones de los datos.
- El proyecto debe ser debidamente probado antes de presentarlo al profesor. Por lo tanto se espera que el programa no se caiga al ejecutarlo, ni tenga un comportamiento no esperado (Si se cae en la presentación, son 5 puntos menos por cada caída)

Documentación (10 %)

- El programa fuente deberá presentar un código documentado internamente (un correcto uso de comentarios).
- Diagrama de Clases (relaciones, herencias, no multiplicidad, modificadores de acceso y estático).
- Se valorará las facilidades para mantenimiento del programa (comentario de encabezado, propiedades y métodos con nombre correcto, métodos con comentarios, uso de sangrías, **Generación de Javadoc**).
- Reporte del proyecto. Se entregará un formato de reporte que los estudiantes deben completar.

Abstracción y Uso de Objetos (15 %)

- Creación de Clases pertinentes con sus propiedades y métodos.
- Correcta interacción de objetos. Que los objetos se comuniquen entre si y no solo sean llamados todos en el main.
- Usar recursos de herencia y sobrescritura para la solución.

Modularidad y Encapsulamiento (15 %)

- Dividir el problema usando los métodos correspondientes. No escribir bloques inmensos de código.
- Encapsular correctamente el proyecto. Crear paquetes donde se agrupen clases que se relacionen.
- No olvidar que para acceder a las propiedades de las clases se debe proveer los métodos get y set.

Herencia y polimorfismo (20%)

- Identificar y emplear cuando sea necesario los principios de herencia y polimorfismo. Revisar los puntos obligatorios que van relacionados con esto.

Penalidades

Se disminuirá un 10% de la nota en caso de no seguir buenas prácticas de programación (convención para nombre de clases, nombre de atributos y métodos, uso de constantes cuando es necesario, o algún fallo grave que se detecte en la forma de programar).

Modo de Entrega:

- Subir un zip de su proyecto al link de la tarea en el aula virtual con el siguiente formato: POO_P#_PY1P_Apellido1_Apellido2_Apellido3
- Solo un estudiante debe entregar el proyecto. Aquel de los tres, cuyo apellido se registre primero en la lista de asistencia.