

Instituto Tecnológico de Costa Rica

Ingeniería en Computación

Programación Orientada a Objetos

Profesor: Adriana Álvarez

Proyecto: Control de préstamos : Biblioteca Alejandrina

Descripción del proyecto



Usted es amante de los libros y tiene una biblioteca de libros físicos (no electrónicos) equivalente a la biblioteca de Alejandría.

Leer es su pasión y los libros uno de sus mayores tesoros.

Sin embargo, a lo largo de los años usted ha notado una disminución significativa de sus libros debido a préstamos que ha hecho a otras personas y que nunca ha recuperado, lo cual le provoca una molestia enorme al punto tal que incluso le ha afectado sus horas de sueño.

Por lo que ha decidido crear una aplicación para llevar el control del préstamo de sus libros.

El sistema debe cumplir con los siguientes requisitos:

1. Registro de personas
 - a. En este registro usted guardará a las personas a las cuales les presta un libro.
 - b. El registro de personas debe incluir al menos los siguientes datos:

1. Nombre
 2. Primer apellido
 3. Segundo apellido
 4. Teléfono
 5. Correo electrónico
 - c. Usted quiere categorizar el tipo de persona a la cual le presta un libro en:
 1. Estudiante
 2. Colega
 3. Familiar
 - d. El sistema debe permitir cargar personas desde un archivo.
2. Registro de libros
- a. En este registro se guardarán al menos los siguientes datos de los libros:
 1. Título
 2. Autor o autores
 3. Editorial
 4. Edición
 5. Imagen de la portada del libro
 6. Calificación. Este valor corresponde a la calificación del libro representado con estrellas según el dueño del libro.
 - b. El sistema debe permitir cargar un archivo con la información de los libros.
3. Control de libros prestados.
4. Edición: El sistema permite editar y actualizar los datos de personas y de libros.
5. Consultas
- a. De libros. La consulta debe mostrar todos los datos del libro (incluyendo la imagen) por orden alfabético según el título con filtros de título, autor y editorial.
 - i. Debe incluir el dato del total de libros consultados.
 - b. De libros prestados. La consulta debe mostrar todos los datos del libro (incluyendo la imagen) por orden alfabético según el título con el filtro del nombre de la persona a la cual se le prestó el libro y un indicador de préstamo que se comporte de la siguiente manera:
 - i. Verde: Libro prestado dentro de los días de préstamo d. Los días de préstamo debe ser un valor parametrizable.
 - ii. Amarillo: Libro prestado con n días de tolerancia, donde n corresponde a la cantidad de días de tolerancia adicionales a los días de préstamo d y está definido por un valor parametrizable.
 - iii. Rojo: Libro prestado con m días de alerta donde m corresponde a una cantidad de días de tolerancia adicionales a los días de préstamo d, se respeta $d < n < m$ y está definido por un valor parametrizable.
 - iv. Debe incluir el dato del total de libros prestados.
 - c. Listado de libros que no se encuentran prestados.
 - i. Debe incluir el dato del total de libros que no se encuentran prestados.
 - d. Top 10 de libros más prestados.
 - i. El valor de 10 debe ser un número parametrizable.
 - e. Libros que se han prestado más de 3 veces en los últimos 6 meses.
 - i. El valor de 3 y 6 deben ser números parametrizables.

6. Procesos:
 - a. Cada vez que se abre el sistema, se debe ejecutar automáticamente un reporte de todos los libros que se encuentran prestados y que superan los días de tolerancia y mostrarlos agrupados en la pantalla principal.
 - b. El sistema debe enviar correos electrónicos a la persona a la cual se le cedió el préstamo del libro con un mensaje amistoso recordando la devolución del libro al dueño. Proponga un mensaje creativo y amistoso que invite a la persona a devolver el libro.
 - c. El sistema debe enviar mensajes de texto a la persona a la cual se le cedió el préstamo del libro con un mensaje amistoso recordando la devolución del libro al dueño. Proponga un mensaje creativo y amistoso que invite a la persona a devolver el libro. Hint: utilice el servicio de www.navegalo.com que es gratis aunque tiene publicidad (extra +1).
7. Ahora que usted lo piensa mejor, le gustaría que la aplicación también le sirviera para el control de su colección de películas e incluso revistas (a futuro puede que otras posesiones también), por lo que debe pensar en el diseño y los ajustes necesarios para que el sistema también funcione con este adicional.
8. Aplicación en Android. Desarrolle la aplicación en Android (extra +2).

Otras consideraciones

1. La aplicación debe ser gráfica.
2. La aplicación puede ser web (extra +1).
3. La aplicación debe tener un logo creado por el equipo. No puede ser un logo copiado de internet o de otra persona y debe ser original. Puede solicitar ayuda a un estudiante de diseño.
4. El código debe trabajarse desde un programa de control de versiones y se deberá demostrar en la defensa la cantidad de commits realizados por cada miembro del equipo.
5. Para aplicar a los puntos extras, la aplicación debe tener el resto de la funcionalidad completa.
6. El equipo puede estar conformado por grupos de un número máximo según le indique el profesor.
7. La entrega oficial se debe hacer en un zip en el Tec Digital antes de la fecha y hora ahí indicada. No se aceptarán trabajos posteriores a esa fecha y hora. En caso de problemas de conexión, se debe aplicar el plan B de la presentación de Reglas del Juego.
8. La entrega debe contener los archivos fuente y el ejecutable de la aplicación (si aplica).
9. Cualquier sospecha de copia anulará el trabajo y se procederá con el trámite administrativo.

Documentación

1. Documento de matriz de casos de prueba (en Excel) con al menos 50 casos de prueba con los cuales se probará el programa. Debe contener un id como enumerador del caso de prueba, la descripción del caso de prueba, el resultado esperado y el estado (Certificado, Con Error, Pendiente). Además de una tabla pivote con la cantidad de casos de prueba en cada estado y el total.
2. Manual de usuario con la descripción del propósito y uso del sistema que incluya imágenes significativas de la aplicación.
3. Diagrama de clases.
4. El código debe estar documentado internamente según Clean Code. Las clases o métodos importantes debe ir documentados con descripción, autor, fecha de creación.

Referencias:

<http://www.booklending.com/index.htm>

<http://www.imdb.com/>

<http://www.navegalo.com/>

http://es.wikipedia.org/wiki/Biblioteca_de_Alejandro_C3%ADa