Universidad Mayor de San Andres
Facultad de Ciencias Puras y Naturales
Carrera de Informática
lab-121
Lic. Cecilia E. Tarquino P.

PROYECTO: Seguimiento de prestamos de material

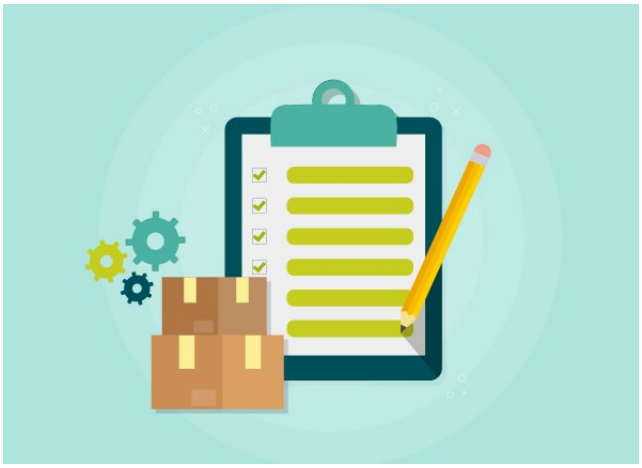
Marco Antonio Vino Chipana
CI 9111299 L.P.
29 de noviembre de 2108

1 Introducción

2 Análisis del ambiente y el problema

2.1 Estado del arte

Como primer recurso se utilizará un inventario, el cual es una relación detallada, ordenada y valorada de los elementos que componen el patrimonio de una empresa o persona en un momento determinado.



El segundo elemento serán los usuarios, en informática un usuario es una persona que utiliza una aplicación o sistema a través del uso de una computadora, la cual para su utilización requiere cierto nivel de experticia.

Cada uno de los recursos que se podrán utilizar serán considerados ítems, Consideraremos como un ítem a cada uno de los recursos físicamente separables que puedan ser prestados individualmente, como ser, un tubo de ensayo, un mortero, etc.

2.2 Problemática

Debemos llevar un registro del historial de los préstamos a los usuarios para poder saber su confiabilidad y si es que tienen o no sanciones por incumplimiento de reglas
Alertar al administrador cuando los materiales no han sido devueltos en los plazos establecidos para poder hacer in seguimiento.
Saber en tiempo real la disponibilidad de recursos para poder solicitarlos en préstamo.
Tener un estimado de cuando debería estar disponible el recurso que necesitamos para poder hacer una planificación.
Averiguar la rotación de los elementos para ver la demanda de recursos y utilizar esta información para futuras adquisiciones, generando preferencias por aquellos elementos que más se hayan encontrado agotados en varias situaciones.

3 Objetivos

3.1 Objetivo general

Realizar seguimiento al movimiento de los recursos susceptibles a préstamo dentro de una institución.

3.2 Objetivos específicos

- Registrar todos los préstamos de material
- Identificar a todos los usuarios que pueden prestarse material
- Establecer plazos de entrega del material prestado
- Mostrar en tiempo real la disponibilidad de materiales
- Analizar el comportamiento de los materiales más utilizados, más buscados, los encontrados agotados con mayor frecuencia.

4 Diagrama de clases

5 Descripción de las clases

5.1 Clases más importantes del desarrollo

En esta sección se describirán las clases propias del desarrollo más allá del entorno gráfico que fueron utilizadas durante el desarrollo para cumplir nuestros objetivos.

5.1.1 Estudiante

Estudiante
- CI : int - nombre : string - np : int - Prestados: int[]
+ Estudiante() : void + Estudiante(int,string) : void + writeArch(BinaryWriter) : void + readArch(BinaryReader) : void + prestarX(int) : void + devolverX(int cod) : bool + getCI() : int + getNombre() : string

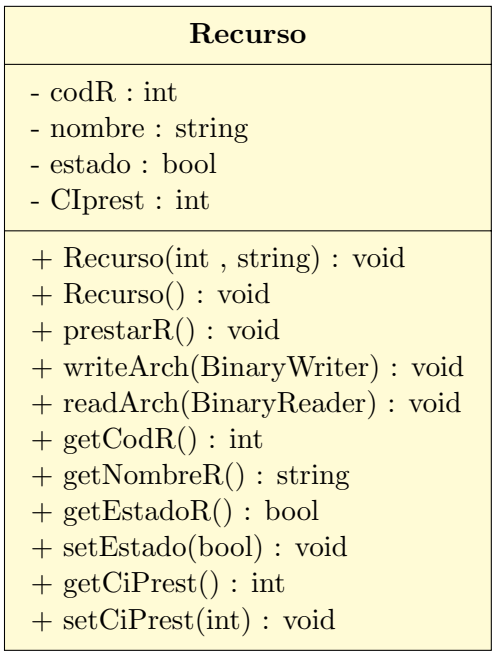
Esta clase guarda la información de cada estudiante, con su *nombre* , su *C.I.* además de todos los recursos que se ha prestado.
El método **prestarX(int)** agrega un recurso a la lista de presados del estudiante. El método **devolverX(int)** elimina de la lista de recursos prestados al estudiante, aquel que tenga el código que se pasa como argumento, retorna verdadero si se pudo realizar la devolución y falso si hubo algún fallo.

5.1.2 Archivo Estudiante

ArchEst
- nombre : string - route : string
+ ArchEst(string) : void + crearNuevo() : void + addE(Estudiante) : void + listar() : LinkedList¡Estudiante¿ + getEstudiante(int): Estudiante + eliminar(int) : bool + getRoute() : string

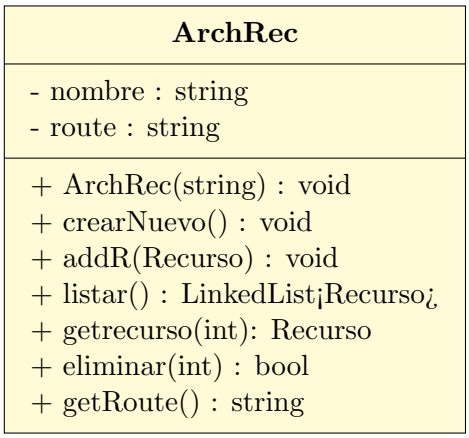
Es la clase que realiza el manejo del archivo donde se guardan todos los estudiantes. Como atributos tiene el *nombre* del archivo, además de su *ruta*. El método **addE(Estudiante)**, agrega al estudiante que pasamos como argumento al final del archivo. La función **listar()** nos permite extraer una lista que contiene a todos los estudiantes guardados en el archivo. la función **getEstudiente(int)** extrae al estudiante cuyo carnet se pasó como argumento. El método **eliminar(int)** borra del archivo al estudiante cuyo carnet se haya ingresado como argumento.

5.1.3 Recurso



La clase recurso se utiliza para guardar la información de un elemento suceptible a prestamo dentro de nuestro sistema, tiene como atributos un identificador único denotado por *codR*, además de un *nombre*, un *estado* (que es verdadero si está en préstamo y falso si no), además de un *CiPrest*, para saber a quien se le prestó el recurso.
El método **prestar()** cambia el estado de un recurso a verdadero, el método **setCiPrest(int)** pone como persona a quien se prestó el recurso, aquella que tenga el carnet ingresado. De la misma forma **setEstado(bool)** nos sirve para cambiar el estado del recurso.

5.1.4 Archivo Recurso



Es la clase que realiza el manejo del archivo donde se guardan todos los Recursos. Como atributos tiene el *nombre* del archivo, además de su *ruta*. El método **addR(Recurso)**, agrega un nuevo Recurso, el que pasamos como argumento al final del archivo, por defecto setea el estado como disponible. La función **listar()** nos permite extraer una lista que contiene a todos los recursos guardados en el archivo. la función **getrecurso(int)** extrae al recurso con el código que se pasó como argumento. El método **eliminar(int)** borra del archivo al recurso cuyo código se haya ingresado como argumento.

5.2 Clases de los formularios del entorno gráfico

Se describirán como funcionan los diferentes formularios desarrollados dentro del entorno gráfico del proyecto.

6 Herramientas y librerias utilizadas

7 Código fuente

8 Ejecución del programa

9 Conclusiones

10 Bibliografía