Universidad de San Carlos de Guatemala Facultad de Ingeniería

Curso: Introducción a la Programación y Computación 2



Erik Vladimir Girón Márquez Carnet # 200313492 Sección C

Guatemala, 26 de Noviembre de 2005

Introducción:

LIBRA es un programa de administración de biblioteca, compilado para la plataforma Windows bajo MS-DOS en modo protegido 32-bit.

Este programa permite el ingreso y control de categorias, editoras, ejemplares de libros, autores, y sus respectivas asociaciones entre ellas, todo esto para que los libros puedan se clasificados estructuradamente, permitiendo un sistema de préstamos y devoluciones eficaz, y un sistema de consulta completo.

La interfaz del programa es muy simple, y contiene un menú principal que permite especificar lo que se desea hacer.

Este programa esta hecho en el lenguaje ANSI C/ISO C++, compilado bajo Borland Turbo C++ 3.1 en Windows, activándole #define bc, y compilado bajo GCC en Linux desactivándole la constante anterior.

Indice:

INTRODUCCIÓN:	2
INDICE:	3
DOCUMENTO DE ANÁLISIS, PROYECTO LIBRA	4
DEFINICIÓN DEL PROBLEMA:	4
Visión:	
OBJETI VOS:	4
REQUERIMIENTOS:	5
ALCANCES:	5
LIMITACIONES:	5
TIEMPO FIJADO:	5
DOCUMENTO DE DISEÑO, PROYECTO LIBRA	_
DOCUMENTO DE DISENO, PROYECTO LIBRA	<u> 6</u>
Arquitectura:	6
DEFINICIÓN DE CLASES:	
LÓGICA:	
LOGICA:	······································
<u>CODIFICACIÓN</u>	<u>9</u>
PRUEBAS:	10

Documento de Análisis, Proyecto LIBRA

Definición del Problema:

- El proyecto consiste en desarrollar un paquete de software para control de biblioteca, implementando las siguientes funciones:
 - Ingreso de Datos(Categorias, Editoras, Autores, Libros, Ejemplares de libros) utilizando estructuras de datos dinámicas.
 - Asociacion de Datos (Categoria-Libro, Auto-Categoria, Autor-Editora, Libro-Editora, Libro-Autor,...)
 - Control de Préstamos y devoluciones (Solicitud de Libros, Prestamos de Ejemplares, Devoluciones de Ejemplares)
 - Generacion de Consultas(Listar en archivo de salida las consultas establecidas en el enunciado)

Visión:

- Se espera tener un sistema de control de libros completamente estable, amigable y funcional para la fecha establecida, simplificando la implementación al utilizar el paradigma de Programación Orientada a Objetos, aprovechando las construcciones que proporciona el lenguaje para implementar estructuras de datos dinámicas que permitan una sencilla manipulacion de datos en memoria.

Objetivos:

- Automatizar el sistema de clasificación, asociación, préstamos y consultas de una biblioteca estándar, aprovechando los recursos computacionales que se ofrece en la actualidad, utilizando el lenguaje de programación C++, aprovechando el paradigma de Programación orientada a objetos y el sistema de *template* (plantillas) al 100%; e implementando con esto, tipos de dato abstracto que permitan la construcción de estructuras de datos dinámicas más intuitivas y a la vez eficientes.
- Proporcionar una interfaz de usuario intuitiva y fácil de utilizar. Aprovechando la el muestreo de texto ASCII en pantalla (stdout).
- Generar un código estable y maduro para la fecha planificada, implantando toda la funcionalidad requerida con el menor número de errores posibles y optimizando el uso de memoria (en el *heap*).
- Escribir un código fuente claro, con formato legible y bien documentado.
- Hacer que éste código generado corra tanto en sistemas Windows, como en la plataforma libre GNU/Linux.
- Establecer como formato de salida un documento TXT, hacia donde se escribirá el reporte.

Requerimientos:

- Software: Sistema operativo basado Windows o Linux que soporte operaciones con archivos, compilador de C++ compatible con estándar ISO 9899 de preferencia.
- Hardware: Ordenador con microprocesador compatible con Intel i386, sistema de almacenamiento secundario con capacidad de soportar modo de escritura.
- Archivo de entrada correctamente escrito, desde donde se darán las órdenes del programa.

Alcances:

- Se manejará únicamente la memoria RAM dentro de las operaciones realizadas por el programa.
- La entrada de datos será únicamente por un archivo de entrada en una sintaxis específica.

Limitaciones:

- La interfaz se usuario será únicamente en modo texto,
- El archivo de datos deberá seguir la sintaxis estricta para que se puedan hacer las operaciones correspondientes, de otro modo se ignorará la operación actual y se escribirá el error en el reporte.
- El reporte generado será únicamente en TXT, por lo que el usuario deberá verlo con su editor de texto preferido.
- Al cerrar la sesión de la aplicación, todos los datos serán eliminados de la memoria principal.

Tiempo Fijado:

31 de Octubre – 05 de Noviembre: Análisis y Diseño

6 – 20 de Noviembre: Implementación de Código, Revisiones al diseño.

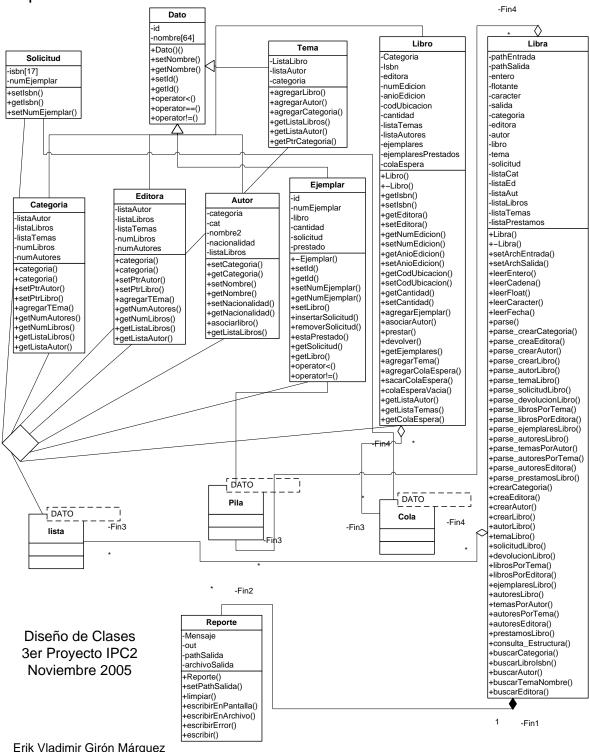
21 – 23 de Noviembre: Pruebas, Mejoras al Código.

24 – 26 de Noviembre: Documentación y Pruebas finales.

Documento de Diseño, Proyecto LIBRA

Arquitectura:

C# 200313492



Definición de Clases:

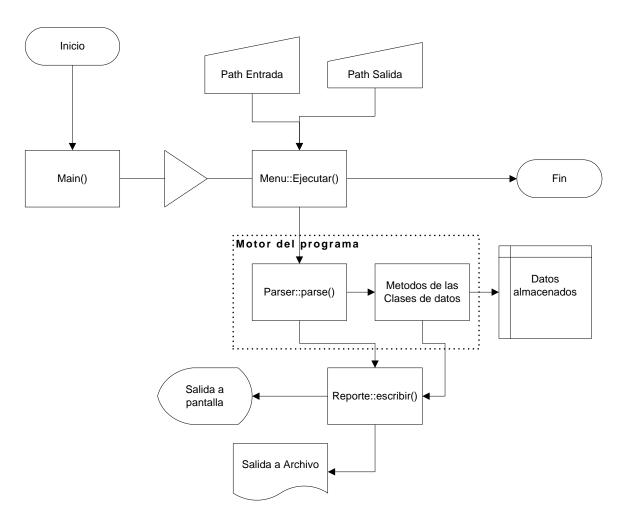
A continuación se presentan la definición formal de cada clase.

```
Lista<Tema> listaTemas;
                                                     Lista<Ejemplar* > listaPrestamos;
class Libro : public Dato{
  long categoria;
                                                    public:
  char isbn[17];
                                                     Libra();
  long editora;
                                                     ~Libra();
  unsigned int numEdicion;
                                                     int validar();
                                                     int setArchEntrada(char *);
  unsigned int anioEdicion;
  char codUbicacion[16];
                                                     int setArchSalida(char *);
                                                     int leerEntero();
  unsigned int cantidad;
  Lista<Tema*> listaTemas;
                                                     char * leerCadena(int&);
  Lista<Autor*> listaAutores;
                                                    float leerFloat();
  pila<Ejemplar> ejemplares;
                                                     char leerCaracter();
  Lista<Ejemplar> ejemplaresPrestados;
                                                    int parse();
  cola<Solicitud> colaEspera;
                                                     int parse_crearCategoria();
 public:
                                                     int parse_crearEditora();
  Libro();
                                                     int parse crearAutor();
  ~Libro();
                                                     int parse_crearLibro();
  long getCategoria();
                                                     int parse_AutorLibro();
  void setCategoria(long);
                                                     int parse_TemaLibro();
  char* getIsbn();
                                                     int parse_SolicitudLibro();
  void setIsbn(char[]);
                                                     int parse_DevolucionLibro();
  long getEditora();
                                                     int parse_LibrosPorTema();
  void setEditora(long);
                                                     int parse_LibrosPorEditora();
  unsigned int getNumEdicion();
                                                     int parse_ejemplaresLibros();
  void setNumEdicion(int);
                                                     int parse_autoresLibro();
  unsigned int getAnioEdicion();
                                                     int parse_temasPorAutor();
  void setAnioEdicion( int);
                                                     int parse_autoresPorTema();
  char* getCodUbicacion();
                                                     int parse_autoresEditora();
  void setCodUbicacion(char[]);
                                                     int parse_prestamosLibro();
  unsigned int getCantidad();
                                                     int crearCategoria();
  void setCantidad(int);
                                                    int crearEditora();
  void agregarEjemplar(Ejemplar ej);
                                                    int crearAutor();
  void agregarEjemplares();
                                                     int crearLibro();
  void asociarAutor(Autor*);
                                                    int asociarAutorLibro();
  Ejemplar* prestar(Solicitud);
                                                     int asociarTemaLibro();
  Ejemplar* devolver(Solicitud);
                                                     int obtenerUltimoIDTema();
  pila<Ejemplar>* getEjemplares();
                                                    int solicitarLibro();
  void agregarTema(Tema*);
                                                     char * devolucionLibro();
  void agregarColaEspera(Solicitud);
                                                     void escribirAutoresLibro(Libro *);
  Solicitud sacarColaEspera();
                                                     void
                                                   escribirLibrosPorCategoria(Categoria *);
  bool colaEsperaVacia();
  Lista<Autor*>* getListaAutor();
                                                     void escribirLibrosPorEditora(Editora*);
  Lista<Tema*>* getListaTemas();
                                                    void escribirTemasPorLibro(Libro*);
  cola<Solicitud>* getColaEspera();
                                                    void escribirLibrosPorTema(Tema *);
};
                                                     void consulta_Estructura();
                                                     int LibrosPorTema(char*);
                                                     int LibrosPorEditora(long);
class Libra{
                                                     int ejemplaresLibros(long,char[]);
 private:
                                                     int autoresLibro(char[]);
  char pathEntrada[255];
                                                     int temasPorAutor(long);
  char pathSalida[255];
                                                     int autoresPorTema(long, char*);
  int entero;
                                                     int autoresEditora(long);
  float flotante;
                                                     int prestamosLibro(long,char*);
  char caracter;
                                                     int solicitudesLibro(long,char*);
  char cadena[512];
                                                     Categoria* buscarCategoria(long );
  Reporte salida;
                                                     Libro* buscarLibroISBN(char *);
  Categoria categoria;
                                                     Autor* buscarAutor(long );
                                                    Tema* buscarTemaNombre(char *);
  Editora editora;
  Autor autor;
                                                     Editora* buscarEditora(long);
  Libro libro;
                                                   };
  Tema tema;
  Solicitud solicitud;
  Lista<Categoria> listaCat;
  Lista<Editora> listaEd;
  Lista<Autor> listaAut;
  Lista<Libro> listaLibros;
```

```
class Tema : public Dato{
  Lista<Libro*> listaLibro;
                                                   #define Editora Categoria
 Lista<Autor*> *listaAutor;
                                                   class Autor : public Dato{
 Categoria* categoria;
 public:
                                                   private:
  void agregarLibro(Libro*);
                                                    long categoria;
  void agregarAutor(Lista<Autor*>);
                                                    Categoria * cat;
  void agregarCategoria(Categoria*);
                                                    char nombre2[50];
  Lista<Libro*>* getListaLibros();
                                                     char nacionalidad[25];
  Lista<Autor*>* getListaAutor();
                                                    Lista<Libro*> listaLibros;
  Categoria* getPtrCategoria();
                                                    public:
                                                     void setCategoria(long);
                                                     long getCategoria();
class Dato{
                                                     void setNombre2(char[]);
private:
                                                     char* getNombre2();
  long id;
                                                     void setNacionalidad(char[]);
                                                    char* getNacionalidad();
 char nombre[64];
 public:
                                                     void asociarLibro(Libro*);
 Dato();
                                                    Lista<Libro*>* getListaLibros();
  void setNombre(char[]);
 char* getNombre();
 void setId(long);
                                                   class Ejemplar{
  long getId();
                                                     int id;
 virtual bool operator<(Dato);</pre>
                                                     int numEjemplar;
  virtual bool operator==(Dato);
                                                     Libro * libro;
 virtual bool operator!=(Dato);
                                                     unsigned int cantidad;
                                                     Solicitud * solicitud;
};
                                                     bool prestado;
class Categoria : public Dato{
                                                    public:
                                                     Ejemplar();
private:
 Lista<Autor*> listaAutor;
                                                     ~Ejemplar();
 Lista<Libro*> listaLibros;
                                                     void setId(int);
 Lista<Tema*> listaTemas;
                                                    int getId();
  int numLibros;
                                                     void setNumEjemplar(int);
 int numAutores;
                                                     int getNumEjemplar();
 public:
                                                     void setLibro(Libro *);
                                                     void insertarSolicitud(Solicitud);
  Categoria();
 Categoria(long);
                                                    void insertarSolicitud(Solicitud*);
 void setPtrAutor(Autor*);
                                                     void removerSolicitud();
 void setPtrLibro(Libro*);
                                                     bool estaPrestado();
  void agregarTema(Tema*);
                                                    void estaPrestado(bool);
 int getNumAutores();
                                                    Solicitud * getSolicitud();
                                                    Libro* getLibro();
  int getNumLibros();
  Lista<Libro*>* getListaLibros();
                                                    bool operator<(Ejemplar);</pre>
Lista<Autor*>* getListaAutor();
                                                    bool operator!=(Ejemplar);
class Solicitud : public Dato{
  //id actua como num de solicitud
  //Carne actua como nombre
  char isbn[17];
 int numEjemplar;
 public:
  void setIsbn(char *);
  char * getIsbn();
 void setNumEjemplar(int);
};
```

Lógica:

A continuación se presenta el modelo entrada y salida de datos del proyecto:

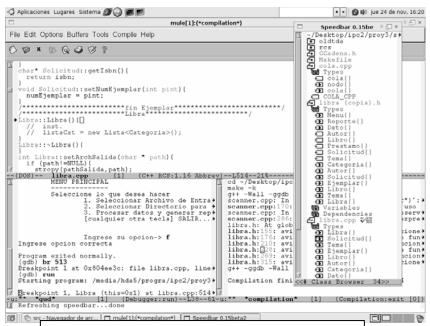


Codificación

Los archivos fuentes son:

- -libra.h:
 - Definicion de clases
- -libra.cpp:
 - Implementacion de las clases y funcion Main
- -lista.cpp, pila.cpp, cola.cpp:
 - Definicion e implementacion de TDAs, usando plantillas.
- -scanner.cpp
 - Modulo de autómatas finitos para lectura de elementos lexicográficos desde el archivo de entrada.

Pruebas:



Salida desde el entorno de GNU Emacs (Linux kernel 2.6.12)

Salida desde el entorno de Borland C++ 3.1 (Linea de comandos de WinXP SP2)