

Paso 1. Identificación del problema

Se identifican de manera adecuada las necesidades que se presentan en el problema propuesto así como sus síntomas y condiciones por los que debe ser resuelto.

Identificación de sus síntomas y necesidades:

- Existe una conglomeración de libros en el espacio que proporciona la biblioteca
- Se presenta un gran ingreso de libros y documentos físicos ya sea por donaciones o por compras.
- Se requiere que el acceso a libros y documentos sea permitido al estudiante, pero que igualmente se reduzca la gran cantidad física que hay de estos.

Descripción del problema:

Una importante institución educativa como ICESI requiere de la implementación de un programa que permita mejorar el inconveniente de la sobrepoblación de libros y documentos que se presenta en su biblioteca.

El sistema debe estar en la capacidad de:

Nombre	R1- Agendar Libros Físico
Resumen	Este requerimiento implica la utilización de información de un libro físico para poder agendar el libro y así poder convertirlo digital
Entradas	
(String Nombre) Indicando el nombre del libro. (String Autor) Indicando el nombre de la editorial del libro. (String Año) indicando el año de creación del libro.	
Resultados	
Un bool indicando si se pudo agregar el libro, False en caso de que el libro ya exista en el registro.	

Nombre	R3- Buscar Libro
Resumen	Este requerimiento implica la búsqueda de un libro a partir de la búsqueda por nombre, esto con el objetivo de indicar si el libro está disponible físico o digital.
Entradas	
(String Nombre) Indicando el nombre del libro.	
Resultados	
Un elemento de tipo Libro	

Nombre	R4- Eliminar Libro Digital
Resumen	Este requerimiento implica la búsqueda de un libro a partir de la búsqueda por nombre, para luego eliminarlo de la lista de libros digitales..
Entradas	
(String Nombre) Indicando el nombre del libro.	
Resultados	
Un bool indicando si se eliminó el libro de la lista de libros digitales.	

Nombre	R5- Eliminar Libro Físico
Resumen	Este requerimiento implica la búsqueda de un libro a partir de la búsqueda por nombre, para luego eliminarlo de la lista de libros Físicos..
Entradas	

(String Nombre) Indicando el nombre del libro.
Resultados
Un bool indicando si se eliminó el libro de la lista de libros Físicos.

Nombre	R6- Actualizar Libro
Resumen	Este requerimiento implica la actualización de la información de un libro existente.
Entradas	
(String Nombre) Indica el nombre del libro que se va actualizar.	
Resultados	
Un bool indicando si se actualizo la información del libro, retorna false en caso contrario o en caso de que no encuentre el elemento solicitado.	

Nombre	RN1 - Mostrar en la interfaz la ruta deseada
Resumen	La aplicación debe ser amigable con el usuario, proporcionando una interfaz gráfica sencilla en la cual se pueda mover facil y rapidamente.

Paso 2. Recopilación de Información

Con el propósito de tener claros todas las definiciones involucradas se hace una búsqueda de los términos relacionados con el problema planteado. Esta búsqueda se realiza en fuentes reconocidas y confiables para reconocer que conceptos hacen parte del problema y no.

Fuentes:

<https://es.wikipedia.org>

http://www.alegsa.com.ar/Dic/texto_plano.php

<https://docs.microsoft.com/es-es/dotnet/framework/winforms/windows-forms-overview>

Archivos de texto plano:

Son aquellos formados exclusivamente por texto sin ningún formato, es decir, que no requieren ser interpretados para leerse. También son llamados archivos de texto llano, simple o sin formato.

Windows Forms:

Las características principales de la programación de Windows Forms y cómo puede usar Windows Forms para compilar smart clients que satisfagan las necesidades actuales de las empresas y usuarios.

C#

C# (pronunciado si sharp en inglés) es un lenguaje de programación orientado a objetos desarrollado y estandarizado por Microsoft como parte de su plataforma .NET, que después fue aprobado como un estándar por la ECMA (ECMA-334) e ISO (ISO/IEC 23270). C# es uno de los lenguajes de programación diseñados para la infraestructura de lenguaje común



Microsoft .NET



.NET es un framework de Microsoft que hace un énfasis en la transparencia de redes, con independencia de plataforma de hardware y que permite un rápido desarrollo de aplicaciones.

Visual Studio

Microsoft Visual Studio es un entorno de desarrollo integrado (IDE, por sus siglas en inglés) para sistemas operativos Windows. Soporta múltiples lenguajes de programación, tales como C++, C#, Visual Basic .NET, F#, Java, Python, Ruby y PHP, al igual que entornos de desarrollo web, como ASP.NET MVC, Django, etc., a lo cual hay que sumarle las nuevas capacidades online bajo Windows Azure en forma del editor Monaco.



Paso 3. Búsqueda de Soluciones Creativas.

Para este paso, se decidió proponer soluciones propias utilizando como herramienta de trabajo grupal de lluvia de ideas. Las soluciones que se plantearon fueron:

Alternativa 1:

Todos los libros a medida que cumplen su ciclo, en vez de comprar nuevos para una nueva generación, se creará un programa para digitalizar todos los libros con el objetivo de ahorrar el mayor espacio posible.

Alternativa 2:

A medida que nos damos cuenta de cuáles son los libros más usados por los estudiantes, se implementará un nuevo programa para digitalizar los más importantes, y poco a poco se irán eliminando la versión física de estos.

Alternativa 3:

La existencia de libros Ebook en una tablet Kindle es la opción más cómoda para los renegados que aun quieren leer de manera física, entonces crearemos una app para convertir fácilmente los libros físicos, en digitales utilizando la tablet. .

Alternativa 4:

Se buscará la posibilidad de donar los libros de menor utilización a una universidad con la cual se tenga un convenio, dando así a los estudiantes la posibilidad de ir a estos libros y tenerlos presencialmente.

Paso 4. Transición de las Ideas a los Diseños Preliminares.

Lo primero que haremos en este paso es descartar las soluciones que se considere que no son factibles. En este caso descartamos la *Alternativa 4* debido a que donar todos los libros físicos a otras bibliotecas supondría quitarle cierto estatus a la universidad y convertiría la biblioteca en un espacio poco frecuentado e inútil, por esa razón se decide eliminar esta alternativa de las posibles soluciones.

A partir de una revisión cuidadosa de las otras alternativas, se saca lo siguiente:

Alternativa 1:

- Debido al alto costo de dinero y tiempo que conlleva digitalizar una gran cantidad de libros, esto lleva a que no sea posible hacerlo con todos los libros.

Alternativa 2:

- Se presentan problemas con los derechos de autor para algunos libros a la hora de digitalizarlos debido a su antigüedad o procedencia.
- Debido a la gran cantidad de libros que hay puede presentar mucho tiempo poder digitalizar los que sí se puedan.

Alternativa 3:

- Se presenta un alto costo de realización debido a la compra de grandes unidades de tablets.

Paso 5. Evaluación y selección de la mejor solución.

Criterios

Deben definirse los criterios que permitirán evaluar las alternativas de solución y con base en este resultado elegir la solución que mejor satisface las necesidades del problema planteado. Los criterios que escogimos en este caso son los que enumeramos a continuación. Al lado de cada uno se ha establecido un valor numérico con el objetivo de establecer un peso que indique cuáles de los valores posibles de cada criterio tienen más peso, es decir cuánto más puntaje tenga será el adecuado para realizar la solución.

- *Criterio A. Completitud.* Se prefiere una solución que encuentre todas las soluciones. Cuántas soluciones entrega:
 - [1] Sólo una o ninguna
 - [2] Más de una si las hay, aunque no todas
 - [3] Todas
- *Criterio B. Precisión de la solución.* El método entrega una solución:
 - [1] Aproximada
 - [2] Exacta

- *Cerio C.* Utilización del espacio digital. Se prefiere un método que no requiera de un gran espacio digital debido al costo que puede tener. La solución requiere de un:
 - [2]espacio digital pequeño.
 - [1]espacio digital grande.

- *Criterio D.* Alcance de la población. Se necesita una solución que sea aceptada por la mayoría de los estudiantes, debido a que son ellos que se verán beneficiados.La solución tiene:
 - [4] gran alcance a la población
 - [3] mediano alcance a la población
 - [2] poco alcance a la población
 - [1] muy poco alcance a la población

- *Criterio E.* Facilidad en implementación algorítmica:
 - [1] Se requiere de demasiados recursos para implementar la solución.
 - [2] Utiliza pocos recursos para implementar la solución.

Evaluación

Según la evaluación con los criterios anteriores en las alternativas se obtiene la siguiente tabla:

	Criterio A	Criterio B	Criterio C	Criterio D	Criterio E	Total
Alternativa 1	Todas 3	Exacta 2	Espacio digital grande. 1	Poco alcance a la población. 2	Se requiere de demasiados recursos para implementar la solución. 1	9
Alternativa 2	Más de una si las hay, aunque no todas 2	Aproximada 1	Espacio digital pequeño. 2	Gran alcance a la población. 4	Utiliza pocos recursos para implementar la solución. 2	11
Alternativa 3	Más de una si las hay, aunque no todas 2	Aproximada 1	Espacio digital grande. 1	Mediano alcance a la población. 3	Se requiere de demasiados recursos para implementar la solución.	8

Selección

De acuerdo con la evaluación anterior se selecciona la Alternativa 2, ya que obtuvo la mayor puntuación de acuerdo con los criterios definidos y porque principalmente es un método que no elimina los libros físicos más importantes que no se puedan conseguir digitalmente, debido a los derechos de autor.

Paso 6. Preparación de informes y especificaciones.

Especificación del Problema (en términos de entrada/salida)

Problema: Sobrepopulación de libros y documentos físicos en una biblioteca.

Entrada: Libros que pertenecen a la biblioteca de forma Física y Digital

Salida: Registro de los libros escogidos para Digitalizar próximamente.

Diagrama de Clases

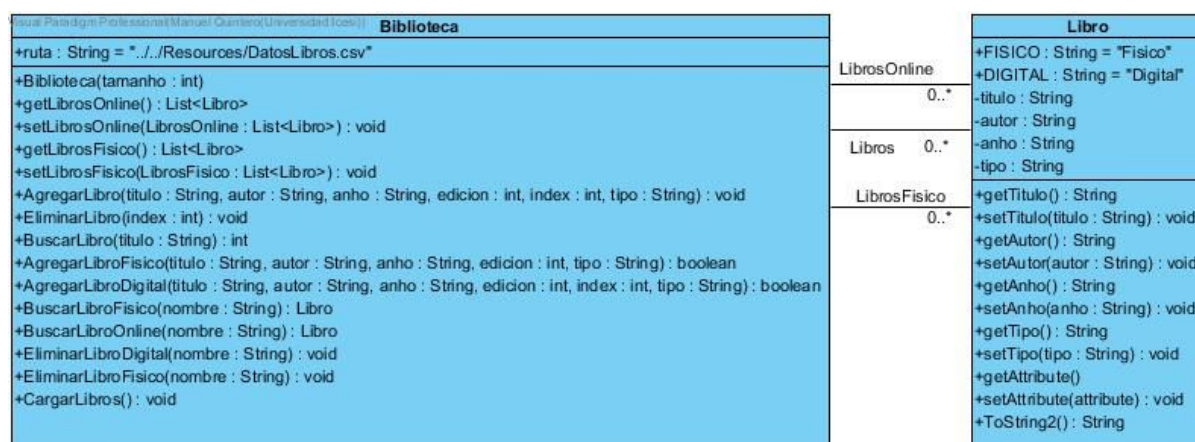
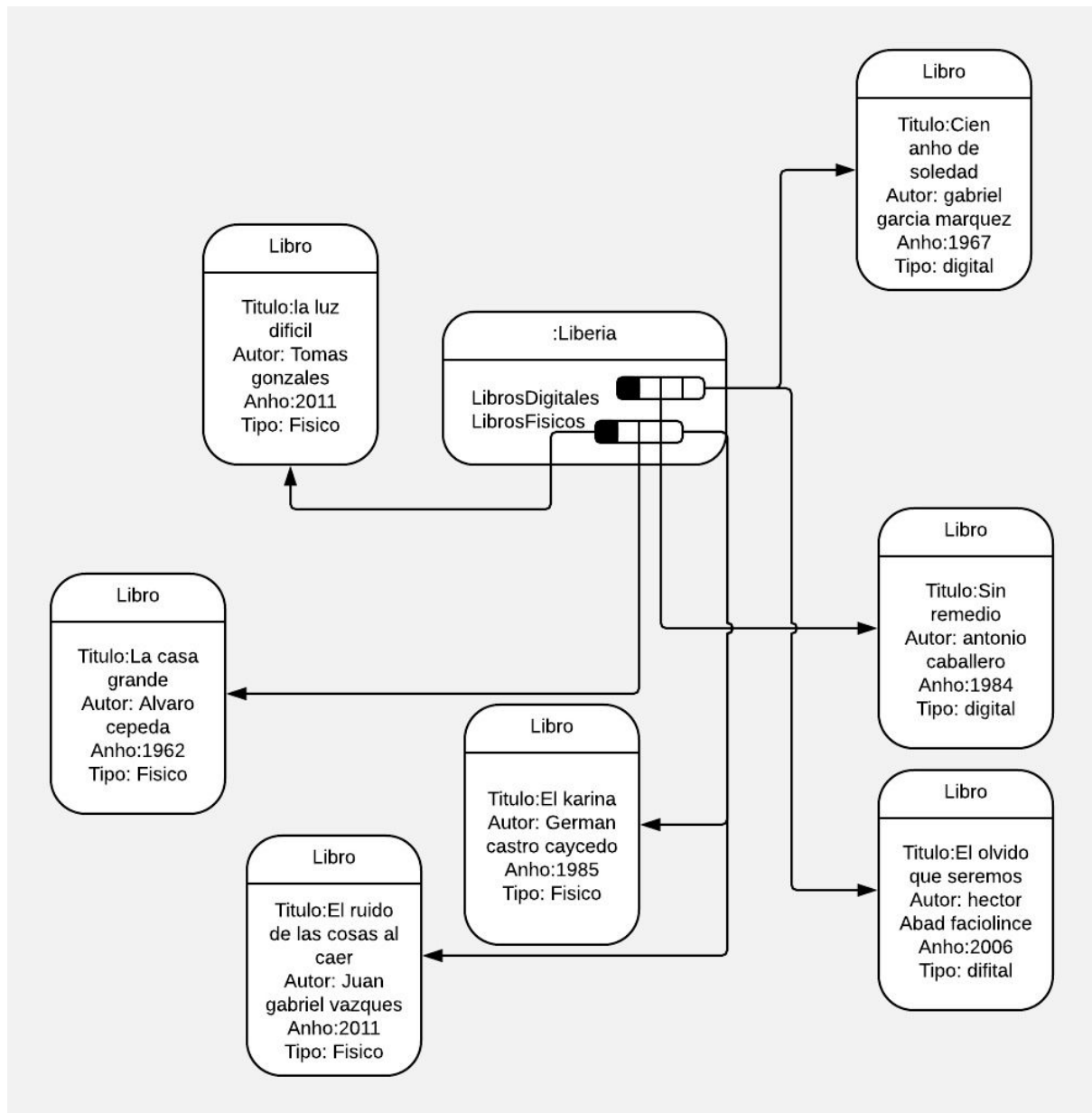


Diagrama de Objetos



Paso 7.Implementación del diseño

Implementación en el lenguaje de programación C# bajo la plataforma Visual Studio 2017

Lista de tareas a implementar:

- Implementación de registro de libros
- Implementación de la consulta de libros desde un archivo de texto plano
- Realizar el guardado por medio de archivos de texto plano
- Implementación de eliminación de libros en la aplicación

Construcción en lenguaje C#

```
class Biblioteca
{
    public const String ruta = "../../Resources/DatosLibros.csv";

    public List<Libro> LibrosFisicos { get; set; }
    public List<Libro> LibrosOnline { get; set; }

    public Biblioteca(int tamanho)
    {
        LibrosFisicos = new List<Libro>(tamanho);
        LibrosOnline = new List<Libro>(tamanho);

    }
    private List<Libro> Libros(){

        List<Libro> union = LibrosFisicos.Union(LibrosOnline).ToList<Libro>();
        return union;
    }

    //Desde aqui
    //

    public bool AgregarLibroFisico(String titulo, String autor, String anho, int edicion, String
tipo)
    {
        bool retorno = true;
        foreach (Libro b in LibrosFisicos)
            if (titulo.Equals(b.Titulo)) throw new Exception("Ya existe dicho libro fisico");
        if (LibrosFisicos.Capacity == LibrosFisicos.Count) throw new Exception("Lista llena");
        Libro a = new Libro(titulo, autor, anho, edicion, 0, tipo);
        LibrosFisicos.Add(a);
        Libros();
        return retorno;
    }
    public bool AgregarLibroDigital(String titulo, String autor, String anho, int edicion, String
tipo)
    {
        bool retorno = true;
```

```

        foreach (Libro b in LibrosOnline)
            if (titulo.Equals(b.Titulo)) throw new Exception("Ya existe dicho libro fisico");
        if (LibrosOnline.Capacity == LibrosOnline.Count) throw new Exception("Lista llena");
        Libro a = new Libro(titulo, autor, anho, edicion, 0, tipo);
        LibrosOnline.Add(a);
        Libros();
        return retorno;
    }

    public Libro BuscarLibroFisico(String nombre)
    {
        Libro buscar = null;
        foreach (Libro b in LibrosFisicos)
            if (nombre.Equals(b.Titulo)) buscar = b;
        if (buscar == null) throw new Exception("No existe el libro fisico");
        return buscar;
    }

    public Libro BuscarLibroOnline(String nombre)
    {
        Libro buscar = null;
        foreach (Libro b in LibrosOnline)
            if (nombre.Equals(b.Titulo)) buscar = b;
        if (buscar == null) throw new Exception("No existe el libro online");
        return buscar;
    }

    public void EliminarLibroDigital (String nombre){
        BuscarLibroOnline(nombre);
        Libros();
    }

    public void EliminarLibroFisico (String nombre){
        BuscarLibroFisico(nombre);
        Libros();
    }

    public void CargarLibros()
    {
        String line;
        try
        {
            StreamReader sr = new StreamReader(ruta);
            line = sr.ReadLine();
            while ((line = sr.ReadLine()) != null)
            {
                String[] prueba = line.Split(',');
                int num;
                String nombre;

```

```

String autor;
String anho;
Random rnd = new Random();
if (prueba.Length > 4)
{
    int contador = 0;
    string nueva = "";
    foreach(String a in prueba)
    {
        if(contador !=0 || contador< prueba.Length -2)
        {
            nueva += a;
        }
        contador++;
    }
    nombre = nueva;
}
else
{
    nombre = prueba[1];
}
num = int.Parse(prueba[0]);
autor = prueba[2];
anho = prueba[3];
String tipo= (rnd.Next(0, 6) < 4 ? "Fisico" : "Digital");

if (tipo.Equals("Fisico") == true) AgregarLibroFisico(nombre, autor, anho, 0,
tipo);

else AgregarLibroDigital(nombre, autor, anho, 0, tipo);
}

foreach(Libro a in Libros())
{
    Console.WriteLine(a.Titulo);
}
Console.ReadLine();
sr.Close();
}

catch( Exception e)
{
    Console.WriteLine(e.Message);
}
Console.WriteLine("a");
Console.ReadLine();
}

```

```

    }

}

public class Libro
{

    public const string fisico = "Fisico";
    public const string digital = "Digital";
    private String titulo;
    private String autor;
    private String anho;
    private int edicion;
    private int index;
    private String tipo;

    public Libro(String pTitulo, String pAutor, String pAño, int pEdicion, int pIndex, String
tipo)
    {
        titulo = pTitulo;
        autor = pAutor;
        anho = pAño;
        edicion = pEdicion;
        index = pIndex;
        this.Tipo = tipo;
    }

    public String Titulo
    {
        get
        {
            return titulo;
        }
        set
        {
            titulo = value;
        }
    }
    public String Autor
    {
        get
        {
            return autor;
        }
        set

```

```

        {
            autor = value;
        }
    }
    public String Anho
    {
        get
        {
            return anho;
        }
        set
        {
            anho = value;
        }
    }
    public int Edicion
    {
        get
        {
            return edicion;
        }
        set
        {
            edicion = value;
        }
    }
    public int Index
    {
        get
        {
            return index;
        }
        set
        {
            index = value;
        }
    }
}

public string Tipo { get => tipo; set => tipo = value; }

public String toString()
{
    return "[" + titulo + ", " + autor + ", " + anho + ", " + edicion + " ]";
}

}

```

