



Método de la Ingeniería

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 el ingreso de libros y documentos físicos ya sea por donaciones o por compras de la misma universidad.
- 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, para que así se reduzca el espacio físico utilizado ayudando a la infraestructura.

Descripción del problema:

La universidad ICESI requiere de la implementación de un programa que permita mejorar el inconveniente de la sobrepoblación de libros físicos y documentos que se presenta en el espacio de su biblioteca.

El sistema debe estar en la capacidad de:

Nombre	R1- Agregar Libro Físico
Resumen	Este requerimiento implica la utilización de información de un libro, para así agregarlo a un registro de libros físicos en la biblioteca, esto se hará mientras aún quepa en el arreglo de libros físicos disponibles.
Entradas	
(String Nombre) Indicando el nombre del libro.	
(String Editorial) Indicando el nombre de la editorial del libro.	
(Integer Pag) indicando el número de páginas del libro.	

Resultados
Un bool indicando si se pudo agregar un libro, False en caso de el libro ya exista en el registro.

Nombre	R2- Agregar Libro Digital
Resumen	Este requerimiento implica la utilización de información de un libro digital, para así agregarlo a un registro de libros físicos en la biblioteca, Esto mientras aún se encuentre al margen de la restricción de memoria de libros que se puedan agregar.
Entradas	
(String Nombre) Indicando el nombre del libro. (String Editorial) Indicando el nombre de la editorial del libro. (Integer Pag) indicando el número de páginas del libro.	
Resultados	
Un bool indicando si se pudo agregar un libro, False en caso de el libro ya exista en el registro.	

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- Buscar libro
Resumen	Este requerimiento implica la búsqueda de un libro a partir de la búsqueda por nombre.
Entradas	
(String Nombre) Indicando el nombre del libro.	
Resultados	
Un objeto tipo libro si digo libro se encuentra guardado en la biblioteca.	

Nombre	R7- Actualizar Libro
Resumen	Este requerimiento implica la actualización de la información de un libro existente.
Entradas	

<p>(String Nombre) Indica el nombre del libro que se va actualizar.</p> <p>(bool prestado) Indicando la condición del libro.</p> <p>(bool digital) Indicando si se digitalizará el libro.</p>
Resultados
<p>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.</p>

Nombre	R8- Porcentaje
Resumen	<p>Este requerimiento implica la actualización de la información de la solución de la definición del problema , dando a conocer el porcentaje de libros físicos.</p>
Entradas	
<p>(String Nombre) Indica el nombre del libro que se va actualizar.</p> <p>(bool prestado) Indicando la condición del libro.</p> <p>(bool digital) Indicando si se digitalizará el libro.</p>	
Resultados	
<p>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.</p>	

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 fácil y rápidamente.

Paso 2. Recopilación de Información

- Consultar al encargado de la prestación de los libros de la biblioteca.
- Observar la cantidad de estudiantes que piden libros en la biblioteca y también aquellos que los piden prestados para la casa.
- Préstamo de los libros para los estudiantes becados en la universidad.

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 empezarán a comprar digitalizados 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, llevaremos una nueva metodología 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, ya que estas tablet cumplen con una condición es que sus pantallas son lo más parecido a un libro.

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.

Se procede a descartar principalmente la *Alternativa 4 debido* debido a que donar todos los libros físicos a otras bibliotecas supondría quitarle cierto estatus a la universidad y la

biblioteca perdería su esencia para la mayoría del personal asistente a esta, 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 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.
- Se presenta una inversión por parte de la universidad para poder abarcar una gran cantidad de derechos de autor.

Alternativa 3:

- Se presenta un alto costo de realización debido a la compra de grandes unidades de tablets, además de algunas manifestaciones por parte de estudiantes y docentes por el hecho de que no es cómodo para ellos el uso de estos dispositivos electrónicos.

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.* Utilización de la tecnología. La solución planteada requiere de la utilización de una:
 - [4] gran cantidad de dispositivos electrónicos.
 - [3] media cantidad de dispositivos electrónicos.
 - [2] poca cantidad de dispositivos electrónicos.
 - [1] muy mínima cantidad de dispositivos electrónicos
- *Criterio B.* Utilización del espacio físico. Se prefiere una solución que utilice la menor cantidad de espacio físico. La solución requiere de una:
 - [3] poca cantidad de espacio físico. (20 a 50 mts²)
 - [2] cantidad mediana de espacio físico (50-80 mts²)

- [1] cantidad grande de espacio físico (80 mts² en adelante)
- *Criterio 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. Costo monetario.* Se necesita una solución que no tenga un gran costo monetario (El presupuesto aún está en vista de planificación). La solución tiene un costo monetario:
 - [3] costo muy bajo.
 - [2] costo moderado.
 - [1] costo elevado .

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	Tota l
Alternativa 1	Poca cantidad de dispositivos electrónicos. 2	Poca cantidad de espacio físico. entre (20 a 50 mts ^2). 3	Espacio digital grande. 1	Poco alcance a la población. 2	Costo moderado 2	10
Alternativa 2	Gran cantidad de dispositivos electrónicos. 4	Cantidad mediana de espacio físico (50-80 mts^2) 2	Espacio digital pequeño. 2	Gran alcance a la población. 4	Costo muy bajo 3	15
Alternativa 3	Poca cantidad de dispositivos electrónicos. 2	Poca cantidad de espacio físico. entre (20 a 50 mts ^2). 3	Espacio digital grande. 1	Mediano alcance a la población. 3	Costo muy elevado 1	10

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: Sobrepoblación 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

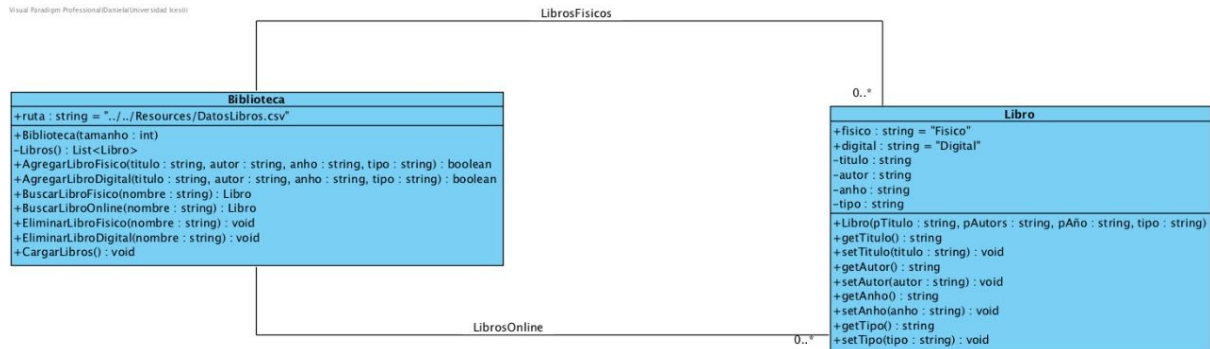
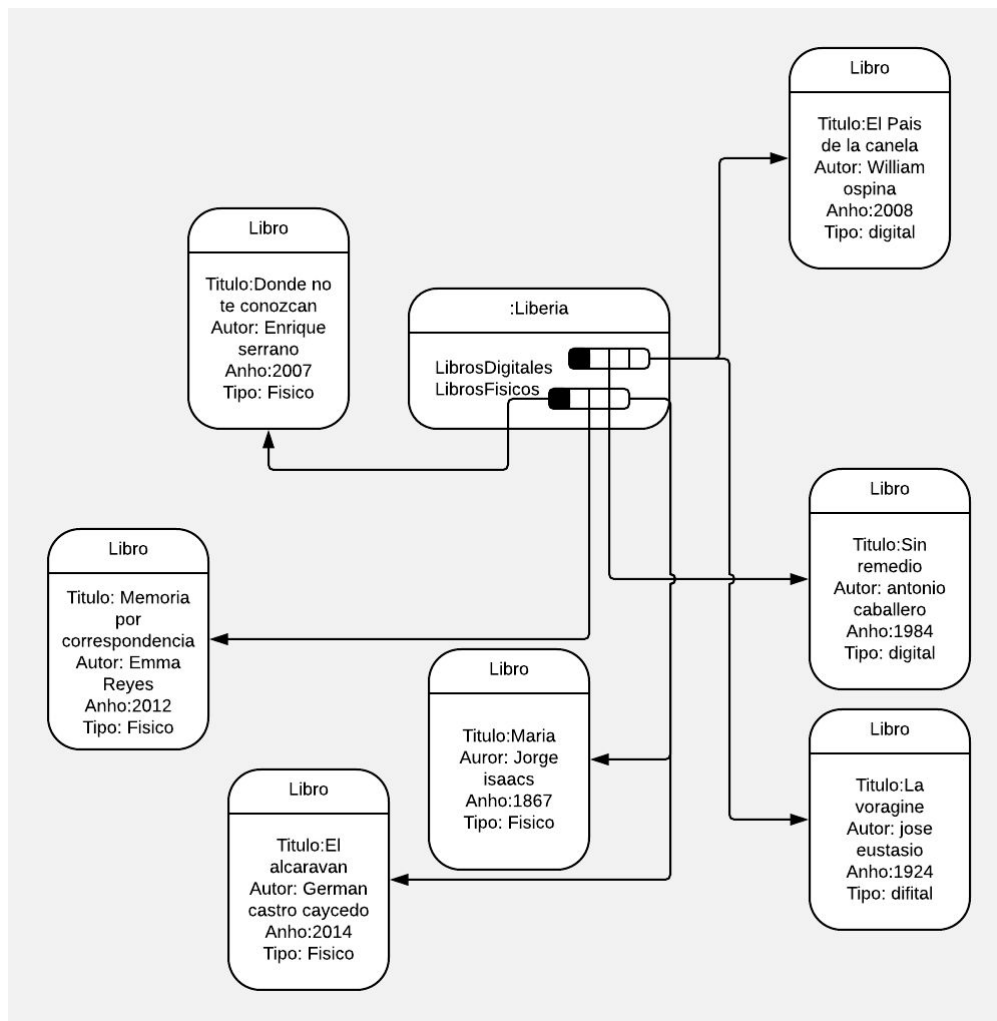


Diagrama de Objetos



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

Clase biblioteca

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Libreria

{

class Biblioteca

{

public const String ruta = "../../../Resources/DatosLibros.csv";

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

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

public double porcentajeFisicos{ get; set; }

public int numLibros { get; set; }

public Biblioteca(int tamanho)

{

LibrosFisicos = new List<Libro>(tamanho);

LibrosOnline = new List<Libro>(tamanho);

porcentajeFisicos = 0;

numLibros = 0;

}

private List<Libro> Libros(){

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

return union;

}

public bool AgregarLibroFisico(String titulo, String autor, String anho, 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, tipo);
        LibrosFisicos.Add(a);
        Libros();
        return retorno;
    }
    public bool AgregarLibroDigital(String titulo, String autor, String anho, 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, 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;
        return buscar;
    }
    public Libro BuscarLibroOnline(String nombre)
    {
        Libro buscar = null;
        foreach (Libro b in LibrosOnline)
            if (nombre.Equals(b.Titulo)) buscar = b;
        return buscar;
    }
    public void EliminarLibroDigital (String nombre){
        foreach(Libro b in LibrosOnline)
            if(nombre.Equals(b.Titulo))LibrosOnline.Remove(b);
        Libros();
    }

    public void EliminarLibroFisico (String nombre){
        foreach(Libro b in LibrosFisicos)
            if(nombre.Equals(b.Titulo))LibrosFisicos.Remove(b);
        Libros();
    }
    public void CargarLibros()
    {

```

```

String line;
try
{
    StreamReader sr = new StreamReader(ruta);
    line = sr.ReadLine();
    while ((line = sr.ReadLine()) != null)
    {
        String[] prueba = line.Split(',');
        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
        {
            if( 1 <prueba.Length) nombre = prueba[1];

            if( 2<prueba.Length ) autor = prueba[2];
            if (3<prueba.Length) anho = prueba[3];
            String tipo= (rnd.Next(0, 6) < 3 ? "Fisico" : "Digital");
            if (!nombre.Equals(""))
            {
                if (tipo.Equals("Fisico") == true) AgregarLibroFisico(nombre, autor, anho, tipo);
                else AgregarLibroDigital(nombre, autor, anho, tipo);
            }
        }
        sr.Close();
    }

}

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

```

```

    }
    Console.ReadLine();
}
public void actualizarPorcentaje()
{
    numLibros = LibrosOnline.Count + LibrosFisicos.Count;
    porcentajeFisicos = ((double)LibrosFisicos.Count / numLibros) * 100;
}
}
}

```

Clase libro

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;

```

namespace Libreria

```

{
    public class Libro
    {
        public const string fisico = "Fisico";
        public const string digital = "Digital";

        private String titulo;
        private String autor;
        private String anho;
        private String tipo;

        public Libro(String pTitulo, String pAutor, String pAño, String tipo)
        {
            Titulo = pTitulo;
            Autor = pAutor;
            Anho = pAño;
            this.Tipo = tipo;
        }
        public string Titulo { get => titulo; set => titulo = value; }
        public string Autor { get => autor; set => autor = value; }
        public string Anho { get => anho; set => anho = value; }
        public string Tipo { get => tipo; set => tipo = value; }
    }
}
}

```

