

# Plataforma para búsquedas avanzadas en entornos de datos desestructurados

TRABAJO DE FIN DE GRADO

GRADO EN INGENIERÍA INFORMÁTICA



**VNiVERSiDAD  
D SALAMANCA**

**ANEXO 3: ESPECIFICACIÓN DE DISEÑO**

SEPTIEMBRE 2023

Autor

JUAN JOSÉ SALVO MATEOS

Tutoras

ANA DE LUIS REBOREDO

MARÍA BELÉN PÉREZ LANCHO



## 1. Contenido

1. Introducción .....	4
1.1. Ámbito del software.....	5
2. Diseño de datos .....	6
3. Diseño arquitectónico .....	7
4. Diseño de la interfaz .....	8
5. Entorno tecnológico del sistema .....	9
6. Plan de desarrollo e implementación .....	10

## 1. Introducción

El presente documento refiere a la especificación técnica del diseño, se tratará el diseño de los datos con sus patrones de diseño, el diseño arquitectónico con sus patrones arquitectónicos y el diseño de interfaces. Además se abordarán temas como el entorno tecnológico necesario y el plan de desarrollo.

## 1.1. **Ámbito del software**

En este proyecto se pretende desarrollar un programa en un entorno WPF .NET, el cual tendrá el objetivo de ser un sistema funcional con una interfaz bien definida, en el cual podremos encontrar un sistema de búsquedas tanto para expresiones regulares como para expresiones literales, en este proyecto se han tomado los siguientes requisitos:

1. Poder realizar búsquedas en sistemas de archivos desestructurados.
2. Poder acceder a una función de búsquedas avanzadas.
3. Poder generar informes sobre las búsquedas.
4. Poder utilizar la aplicación de forma intuitiva y simple.

## 2. Diseño de datos

Los datos en este sistema se recogerán de dos tipos de datos, datos persistentes y datos no persistentes.

Los datos no persistentes se almacenarán en tiempo de ejecución en la aplicación y serán eliminados al cerrar la aplicación. Para almacenar este tipo de datos se ha utilizado un patrón singleton, este patrón nos da la capacidad de almacenar los datos en una única instancia accesible desde cualquier punto del programa. Los datos regidos de este tipo serán:

1. Ruta de directorio actual.
2. Listado de archivos del directorio actual.

Los datos persistentes en memoria se almacenarán en formatos TXT, estos datos consistirán en dos conjuntos principales:

1. Un sistema de archivos desestructurados
2. Un directorio que almacene expresiones regulares personalizadas

Sobre estos datos trabajará el grueso del programa, ya que deberemos leerlos, analizarlos y mostrar resultados.

### 3. Diseño arquitectónico

La arquitectura de este programa se basará en un patrón MVC, en el cual encontraremos tres capas, una de modelos, una de vistas y una capa de controladores como bien indica el nombre del patrón.

En la capa del modelo se almacenarán los datos no persistentes, consistirá en un patrón singleton donde crearemos una única instancia en tiempo de ejecución y trataremos todos los datos desde el controlador.

En la capa del controlador nos encontraremos dos controladores, uno para las funcionalidades de búsqueda y otro para la generación de informes.

En la capa de la vista nos encontraremos dos ventanas, la primera será la ventana principal de la aplicación, que contendrá dos páginas, una para la carga de directorio y otra para el grosor de la aplicación. La otra ventana servirá para almacenar expresiones regulares.

## 4. Diseño de la interfaz

Este apartado se corresponde con la capa de la vista, en ella encontraremos dos ventanas creadas para funcionar con el framework WPF de .NET, por lo que cada ventana que creemos estará formada por mínimo un archivo XAML y un archivo C#, aún así en esta práctica se ha considerado oportuno aplicar una de las funcionalidades de WPF que son las pages, esta funcionalidad nos aportará una mayor versatilidad para cambiar entre interfaces en la ventana principal.

Las interfaces en esta práctica serán las siguientes:

1. Interfaz de carga de directorio, esta interfaz será utilizada para que el usuario indique el directorio que desea cargar a la aplicación.
2. Interfaz principal, en esta interfaz encontraremos las funcionalidades principales de la aplicación, el usuario podrá introducir expresiones, ya sean literales o regulares; ver listado de resultados, estos resultados estarán formados por bloques generados en código; visualizar listado de archivos, el usuario podrá activar o desactivar los archivos en los que desea realizar búsquedas; Acceder al menu, donde podrá acceder a funcionalidades tales como cargar expresiones regulares, almacenar expresiones regulares, generar informes y cargar un nuevo directorio.
3. Interfaz de almacenamiento de expresiones regulares, en esta interfaz el usuario podrá añadir descripciones y títulos a las expresiones regulares a almacenar para su posterior consulta en los directorios.

## **5. Entorno tecnológico del sistema**

El entorno del sistema constará de un equipo equipado con Windows, este equipo no es necesario que este compuesto por componentes muy potentes ya que la aplicación cuenta con un bajo consumo.

## 6. Plan de desarrollo e implementación

El desarrollo se ha organizado de la siguiente forma.

1. Creación de una interfáz básica con la que trabajar para cargar directorios.
2. Creación de una interfaz básica a la que agregar las funcionalidades principales.
3. Creación de la funcionalidad para cargar directorios
4. Creación de listados en la interfaz principal y capacitarlos para que el usuario pueda activarlos y desactivarlos.
5. Creación de la funcionalidad para realizar búsquedas.
6. Creación de la funcionalidad de mostrar resultados.
7. Creación de la funcionalidad de cargar y almacenar expresiones regulares
8. Modificar la forma de mostrar resultados para que sean compuestos por bloques de códigos y posteriormente mostrados en la interfaz principal.
9. Creación de la funcionalidad para almacenar informes.
10. Modificación de toda la interfaz para hacerla más intuitiva y estética.