

**UNIVERSIDAD RAFAEL LANDIVAR**

**FACULTAD DE INGENIERIA**

**INGENIERIA EN INFORMATICA Y SISTEMAS**

**MANEJO E IMPLEMENTACION DE ARCHIVOS**

**PABLO ISAAC GARCIA ORELLANA 15352-18**

**REPORTE DE PROYECTO**

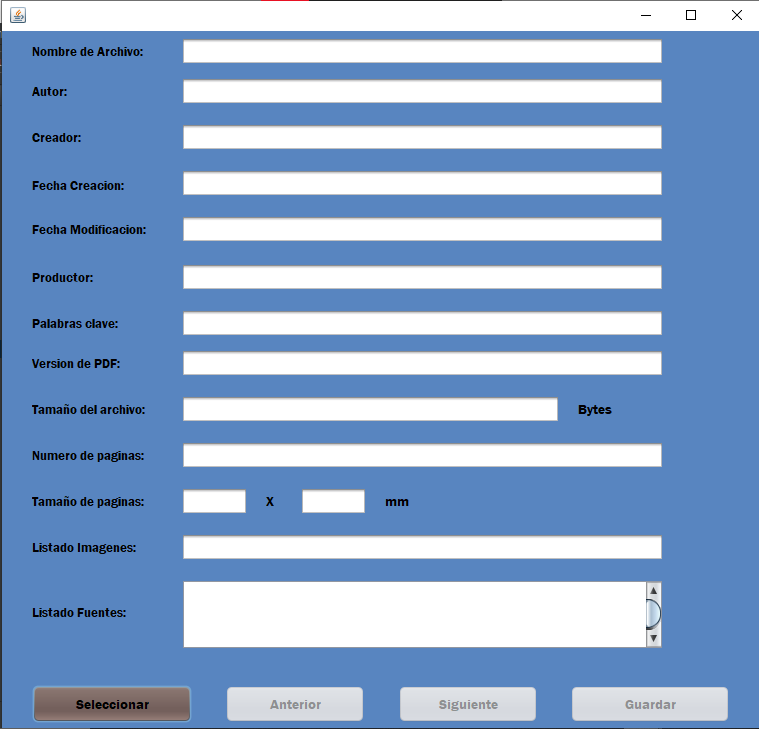
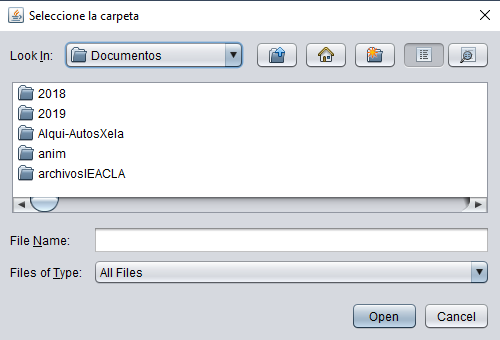
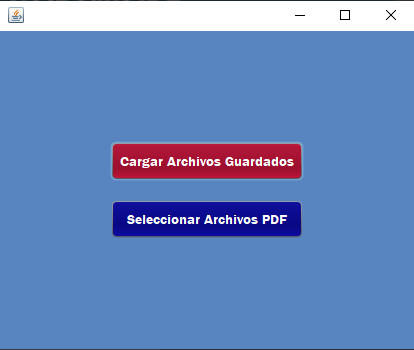
# **INTRODUCCION**

El formato PDF (Portable Document Format) es en la actualidad el estándar de facto en todo el mundo para el intercambio de documentos tanto a nivel empresarial como de usuario. Su éxito está dado principalmente por ser multiplataforma, puede visualizarse e imprimirse de forma indistinta en sistemas Macintosh, Microsoft Windows, Unix, Linux y otros dispositivos móviles.

Así, este proyecto pretende crear una herramienta de software que sea capaz de leer los metadatos de los documentos PDF, tamaño del archivo, número de página, título del documento, asunto, palabras clave, tipo de archivo, versión de PDF, aplicación con la que fue creada, lista de imágenes que pueda tener el documento y lista de fuentes que incluye el documento.

Se realizó previamente un estudio de métodos para conocer la estructura de los diferentes tipos de archivos PDF y poder obtener la información mencionada.

# **MANUAL DE USO**



Está será la segunda ventana que le aparecerá al usuario, Le daremos al botón “Seleccionar” y se abrirá una ventana emergente en donde podrá elegir los archivos PDF que desee leer.

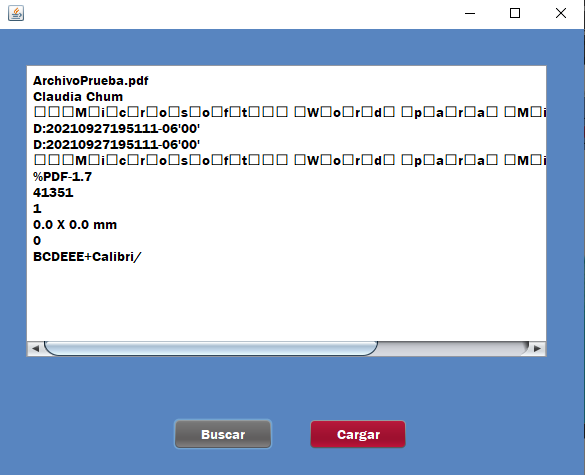
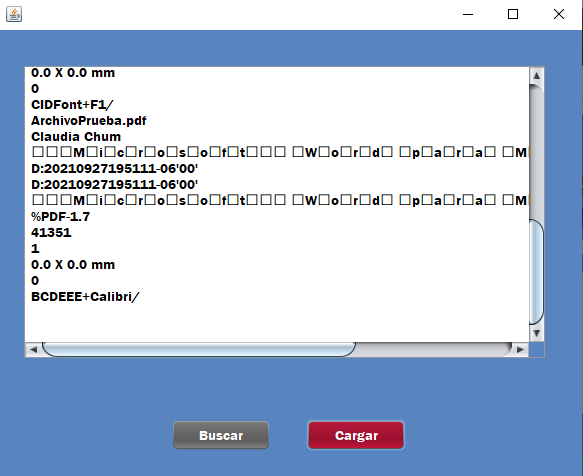
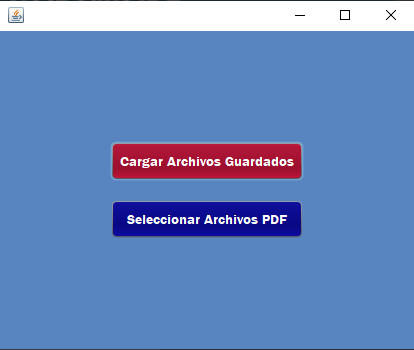
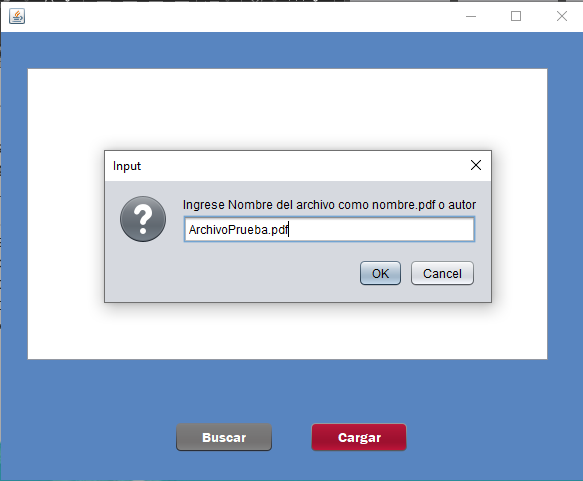
Tendrá la opción de elegir una carpeta y darle open. Si quiere abrir y leer una carpeta que esta adentro de otra darle doble click a la carpeta, al momento de leer la carpeta correcta le aparecerá lo siguiente al usuario.

Al momento de ejecutar el programa, aparecerá esta ventana.

Primero necesitaremos Seleccionar Archivos

Regresaremos al formulario principal y ahora podremos cargar los archivos guardados.

En este formulario tendremos la opción de cargar los datos de todos los archivos pdf que están almacenados o de buscar el archivo por el nombre incluyendo “.pd” o por el autor.



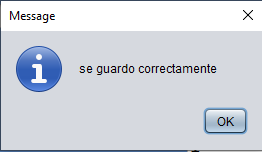
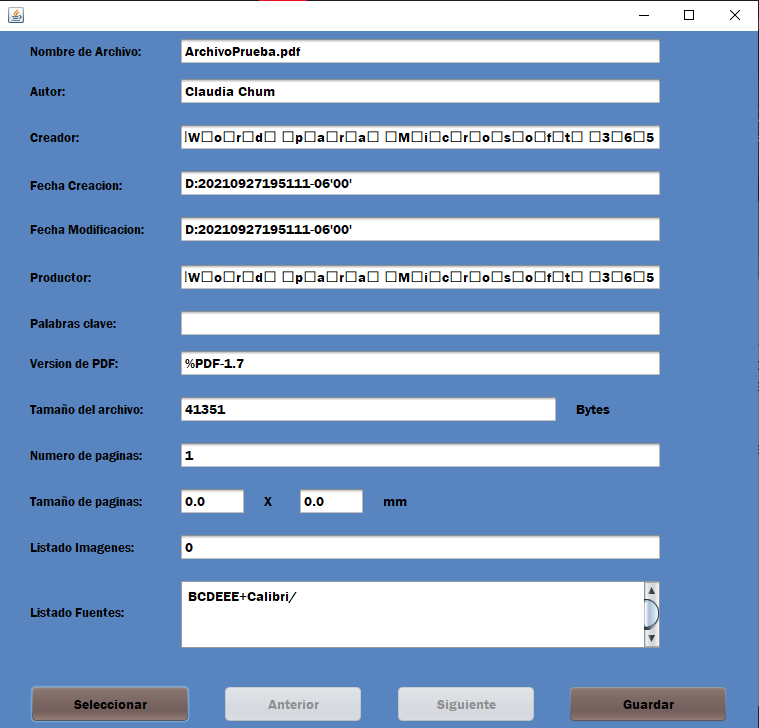
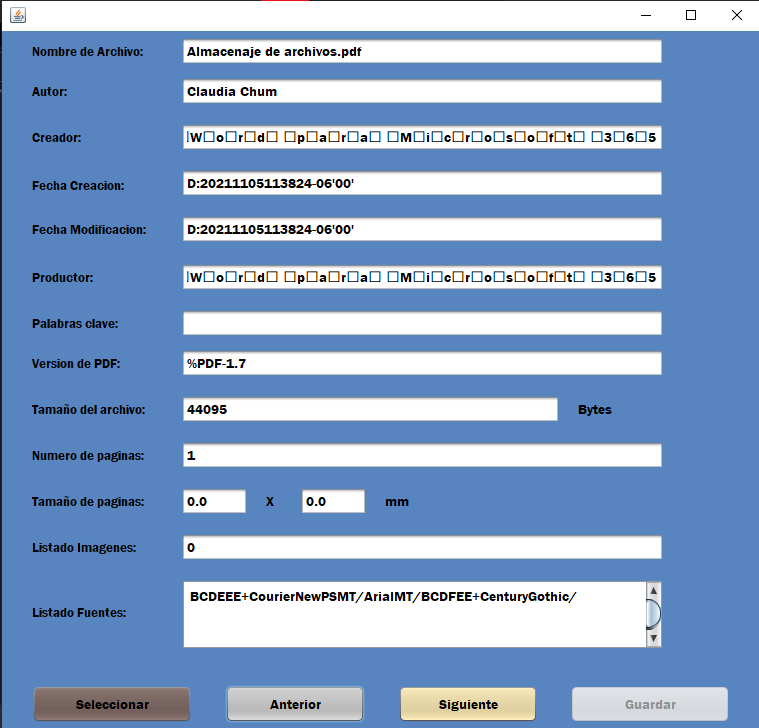
Un solo archivo

Si abre una carpeta podrá navegar sobre todos los archivos con los botones “Siguiente” y “anterior” y hasta llegar al último archivo se habilitará el botón de “Guardar”

Si abre un solo archivo, los botones de “Siguiente” y “Anterior” estarán bloqueados y solamente podrá guardar el documento.

En esta ventana es en donde se podrá modificar el nombre o el autor del archivo. Una vez qu e se guarde ya no se podrá modificar nuevamente.

Al guardarlo mostrará un mensaje emergente en donde confirmará que se guardó correctamente.



|  |  |  |  |
| --- | --- | --- | --- |
|  | **Lectura** | **Archivo** |  |
| **VARIABLE** | **No. Bytes** | **TIPO** | **JUSTIFICACIÓN** |
| Nombre | El # de bytes varían | Byte[] | Se usa de tipo array, porque varia la cantidad de  caracteres. |
| Autor | El # de bytes varían | Byte[] | Se usa de tipo array, porque varia  la cantidad de caracteres. |
| Creador | El # de bytes varían | Byte[] | Se usa de tipo array, porque varia  la cantidad de caracteres. |
| Fecha Creación | 23 bytes | Byte[] | Se usa de tipo array, y se le asigna un valor por que son los que nosotros  creemos que utilizará. |
| Fecha Modificación | 23 bytes | Byte[] | Se usa de tipo array, y se le asigna un valor porque son los que nosotros creemos que  utilizará. |
| Productor | El # de bytes varían | Byte[] | Se usa de tipo array, porque varia la cantidad de  caracteres. |
| Versión | 8 bytes | Byte[] | Se usa de tipo array, y se le asigna un valor por que son los que nosotros creemos que son  los que utilizará. |

# **ESTRUCTURA DE ALMACENAMIENTO**

|  |  |  |  |
| --- | --- | --- | --- |
| Tamaño | 4 bytes | Int | Aquí se utiliza de |
|  |  |  | tipo int, ya que son |
|  |  |  | suficientes los 4 |
|  |  |  | bytes para el |
|  |  |  | tamaño. |
| Paginas | 4 bytes | Int | Se utiliza tipo int, ya que los 4 bytes son |
|  |  |  | suficientes para su |
|  |  |  | uso. |
| Tamaño X | 4 bytes | Float | Se utiliza float por |
|  |  |  | que se da en |
|  |  |  | decimales. Y son |
|  |  |  | suficientes para el |
|  |  |  | uso del tamaño. |
| Tamaño Y | 4 bytes | Float | Se utiliza float por que se da en |
|  |  |  | decimales. Y son |
|  |  |  | suficientes para el |
|  |  |  | uso del tamaño. |
| Imágenes | 4 bytes | Int | Aquí es de tipo int, |
|  |  |  | ya que es suficiente |
|  |  |  | para contar la |
|  |  |  | cantidad de |
|  |  |  | imágenes. |
| Fuentes | El # de bytes | Byte[] | Aquí es un array |
|  | varían |  | depende de los caracteres que |
|  |  |  | puede tomar en |
|  |  |  | total. |

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Escritor** | **Archivo** |  |
| **VARIABLE** | **No. Bytes** | **TIPO** | **JUSTIFICACIÓN** |
| Nombre | Variable | String | Variable String que guarda el nombre  del archivo. |
| Autor | Variable | String | Variable String que  guarda el autor del archivo. |
| Creador | Variable | String | Variable de tipo String que guarda el creador del  archivo. |

|  |  |  |  |
| --- | --- | --- | --- |
| Fecha Creación | 23 bytes | Char[] | Array con un valor asignado que creemos son lo que utilizará para  almacenar la fecha de creación. |
| Fecha Modificación | 23 bytes | Char[] | Array con un valor asignado que creemos son lo que utilizará para almacenar la  fecha de  modificación. |
| Productor | Variable | String | Variable de tipo String que guarda  el productor. |
| Versión | 8 bytes | Char[] | Array con un valor asignado que creemos son lo que utilizará para almacenar la  versión del archivo. |
| Tamaño | 4 bytes | Int | Variable tipo int, suficiente para  guardar el tamaño. |
| Paginas | 4 bytes | Int | Variable tipo int, suficiente para guardar el número  de páginas. |
| Tamaño X | 4 bytes | Float | Variable tipo int, suficiente para  guardar el tamaño. |
| Tamaño Y | 4 bytes | Int | Variable tipo int, suficiente para  guardar el tamaño. |
| Imágenes | 4 bytes | Int | Variable tipo int, suficiente para  guardar las imágenes. |
| Fuentes | Variable | String | Variable de tipo String que guarda  las fuentes. |

# **ALGORITMOS UTILIZADOS**

* Lectura Metadatos: este algoritmo consiste en varios ciclos while, condiciones, ciclos for, y variables que leen partes específicas del archivo PDF para poder obtener la información, se tuvo que navegar a través de la estructura del archivo para llegar a los metadatos, para leer ordenadamente los archivos se utilizo una estructura para separar e ir encontrando los datos.
* Lectura de Versión: este algoritmo consiste en leer la primera línea del PDF, ya que todos los PDF tiene versión en la primera línea.
* Lectura de Tamaño: este algoritmo esta estructurado de comandos directos, obtener el tamaño del archivo en bytes.
* Lectura Numero de Paginas: este algoritmo consiste en varios ciclos while, condiciones, ciclos for, y variables estan estructurados para navegar en el archivo PDF y asi solo leer el número de páginas.
* Lectura Tamaño de la Pagina: este algoritmo consiste en varios ciclos while, condiciones, ciclos for, y variables estan estructuradas para navegar en el archivo PDF y leer solamente el tamaño de pagina que estaba dada en point y el algoritmo lo convierte en milímetros.
* Lectura de Imágenes: este algoritmo consiste en varios ciclos while, condiciones, ciclos for y variables estan estructuradas para navegar en el archivo PDF e ir leyendo página por página las imágenes que contiene cada una.
* Lectura de Fuentes: este algoritmo consiste en varios ciclos while, condiciones, ciclos for, y variables estan estructuradas para navegar en el archivo PDF e ir leyendo las fuentes que contiene cada pagina diferente.

* Escritura Aleatoria: este algoritmo consiste en condiciones, variables que estan estructuradas para crear un archivo de tipo .bin que almacena registros de los archivos PDF y un ciclo while para recorrer la lista del índice del mismo.
* Leer Índice: este algoritmo consiste en almacenar en una lista todos los índices del archivo .bin para no perder la información de los mismos y sobre escribir al final del archivo
* Búsqueda de Archivo: este algoritmo consiste en varios ciclos while, condiciones, ciclos for y variables estan estructuradas para navegar dentro del archivo creado con registros de PDF, esto para buscar tanto como el nombre o autor.
* Lector Aleatorio: este algoritmo consiste en varios ciclos while, condiciones, ciclos for y variables estan estructuradas para navergar en el archivo, almacena la información del PDF en un .bin, de tal forma, al momento de buscarlo aparece la informacion en una ventana
* Lista, InsertarM, NodoM, MostrarM: este algoritmo consiste en una lista doblemente enlazada en donde se puede recorrer los diferentes registros de archivos PDF, con un botón de siguiente y anterior que son lo que nos ayudan a recorrer dentro de una carpeta con varios archivos PDF.

# **COCLUSIONES**

La realización del estudio de métodos en el proyecto Lector, Escritura, Buscador y Modificación del pdf es importante debido a que, a partir de éste, se puede elaborar un sistema de mejoras a los procesos estudiados, haciendo que sean óptimos y que a su vez permita aumentar los niveles de productividad de la generación.

Durante y posterior a la construcción del proyecto se realizaron pruebas en el sistema para garantizar la calidad del funcionamiento de los mismos respecto a los criterios solicitados

Con la realización de este proyecto se han cumplido los objetivos que se fijaron al comienzo del mismo, obteniendo resultados que los validan.