Modelo de Diseño

"Sistema de lectura y comparación de archivos"

Sobera Sotomayor Iñaki
Alvarado Fragoso Miguel Ángel
Colín Reyes Tania Carolina

Ingeniería de Software
Universidad Tecnológica de México

Versión [1]

Tabla de Contenido

I. Introducción

Proporciona una funcionalidad básica para comparar archivos PDF y encontrar palabras comunes. Utiliza una GUI para la interacción del usuario y aprovecha bibliotecas externas para la manipulación de PDF y el procesamiento de texto. Lo que hace que algunos de los **objetivos** sean:

comparar dos archivos PDF: El objetivo principal del sistema es comparar dos archivos PDF y encontrar las palabras comunes entre ellos.

Facilitar la búsqueda de información: Al identificar las palabras comunes, el sistema facilita la búsqueda de información específica en dos documentos extensos.

Generar un documento con las coincidencias: El sistema puede generar un nuevo documento PDF que liste las palabras comunes, lo que permite a los usuarios tener una vista rápida de las coincidencias.

REQURIMIENTOS:

Capacidad para leer archivos PDF: El sistema debe ser capaz de leer y procesar archivos PDF utilizando bibliotecas como Apache PDFBox o iText.

Extracción de texto: El sistema debe extraer el texto de los archivos PDF y convertirlo a minúsculas para facilitar la comparación.

División de texto en palabras: El sistema debe dividir el texto en palabras individuales, ya sea utilizando espacios en blanco o puntuación como delimitadores.

Comparación de palabras: El sistema debe comparar las palabras de ambos documentos y encontrar las coincidencias.

Generación de un nuevo PDF: El sistema debe ser capaz de generar un nuevo documento PDF que liste las palabras comunes.

Interfaz gráfica de usuario (GUI): El sistema debe tener una GUI intuitiva para que los usuarios puedan seleccionar los archivos PDF y realizar la comparación.

II. Criterios

Arquitectura

La arquitectura está basada en componentes principales lo cuales son:

Interfaz gráfica de usuario (GUI): Desarrollada con la biblioteca Swing, permite al usuario seleccionar los archivos PDF a comparar y visualizar los resultados.

Módulo de lectura de PDF: Implementado utilizando las bibliotecas Apache PDFBox e iText, este módulo extrae el contenido textual de los archivos PDF seleccionados.

Módulo de comparación: Este módulo analiza los dos conjuntos de palabras extraídas y encuentra las palabras comunes.

Módulo de generación de informes: Crea un nuevo archivo PDF que contiene las palabras comunes encontradas.

Herramientas

Lectura de PDF: El código utiliza dos bibliotecas para manejar archivos PDF:

Apache PDFBox: Una biblioteca de código abierto que se utiliza para leer y manipular documentos PDF. Permite extraer contenido de texto, que es la funcionalidad utilizada en este programa.

iText: Otra biblioteca de código abierto para crear y manipular archivos PDF. Este fragmento de código utiliza iText para crear un nuevo documento PDF que contiene las palabras comunes encontradas entre los dos archivos PDF de entrada.

III. Inventario de Artefactos

Nombre	Tipo	Archivo	Descripción
Archivo de lectura	Archivo	Un archivo de lectura en PDF	Este archivo es un documento de lectura de un cierto tema de donde se van a sacar ciertas palabras o frases para el diccionario
Diccionario	Archivo	DICCIONARIO DE UN PDF	En el diccionario se alojarán un cierto tipo de palabras o feraces para que el programa busque las coincidencias en el archivo de lectura
Resultado	Archivo	Archivo PDF	Este es el archivo resultante de la lectura del archivo de lectura y el diccionario, ya que lo que contendrá este archivo será las palabras en común que tiene el archivo de lectura con el diccionario

IV. Restricciones de la Arquitectura

Seguridad: El programa no tiene medidas de seguridad para evitar el acceso no autorizado a los documentos PDF.

Portabilidad: El programa está escrito en Java, lo que lo hace portable a cualquier sistema operativo que tenga una máquina virtual Java (JVM).

Escalabilidad: El programa no está diseñado para manejar grandes cantidades de documentos PDF.

Disponibilidad: El programa no tiene mecanismos para garantizar la disponibilidad continua del servicio.

V. Criterios de Orden de Construcción

Orden de Construcción en Base a la Prioridad

Prioridad	Criterios
1	Selección de archivo de lectura
1	Selección de Diccionario
2	Manejo de errores durante la selección de archivo seleccionado para lectura de PDF
3	Extracción de texto de los documentos PDF
4	Identificación de palabras comunes

Definición de Criterios Para el Orden de Construcción

Prioridad	Componente
1	Seleccionamos primero el archivo de lectura porque es necesario leerlo antes de compararlo con el archivo de diccionario
2	Seleccionamos después el diccionario para guardar las palabras que se compararan con el archivo de lectura
3	La función principal comparara las palabras del archivo de diccionario con las del archivo de lectura guardando esas palabras clave en un archivo externo
4	Crea el archivo con las palabras

VI. Ejecución de Casos de Uso

<Archivo de lectura >

Es el archivo de lectura en el que se va a basar el diccionario para sacar las palabras o frases clave el cual el archivo es un PDF con algún tema en especifico

Artefactos involucrados

Documento de lectura: es el primer archivo leído por el programa
Seleccionar archivos PDF utilizando JFileChooser.
Leer el contenido del PDF utilizando la biblioteca Apache PDFBox
Dividir el texto extraído en palabras individuales.

Bibliotecas:
pache PDFBox (org.apache.pdfbox): Esta biblioteca de terceros proporciona funcionalidades para leer y procesar documentos PDF.

Código fuente: el archivo contiene la funcionalidad principal del sistema.

Descripción

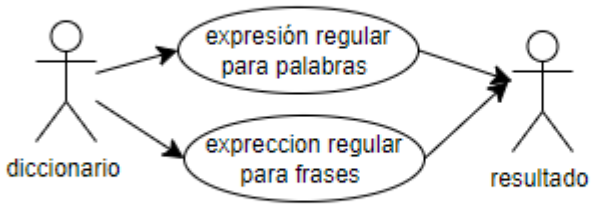
<Diccionario>

Identificar las similitudes y diferencias entre entre el documento de lectura a partir de las palabras clave que compone el diccionario.

Artefactos involucrados

Algunos artefactos utilizados fueron:
El documento de lectura: fue el primer documento leído por el programa
Diccionario: fue el segundo documento leído por el programa
Librería 1 **Apache PDFBox**: Extraer el contenido textual de los documentos PDF.
Librería 2 **Java Collections**: Almacenar y manipular las palabras clave.
Librería 3 **Apache Commons Lang:** Eliminar stopwords (palabras comunes irrelevantes).
Librería 4 **Lucene:** Implementar técnicas de extracción de palabras clave como TF-IDF.
Librería 5 **JSimilarity:** Implementar métodos de comparación de palabras clave como la distancia de Levenshtein o el Jaccard Index.

Descripción



El programa compara dos documentos de texto y crea un nuevo archivo que contiene las palabras que son comunes a ambos documentos.

<Resultado>

El sistema de comparación de archivos PDF, genera un nuevo archivo PDF que contiene las palabras comunes encontradas entre los dos documentos seleccionados.

El nuevo PDF permite a los usuarios comparar fácilmente las palabras que se encuentran en dos documentos diferentes.

Artefactos involucrados

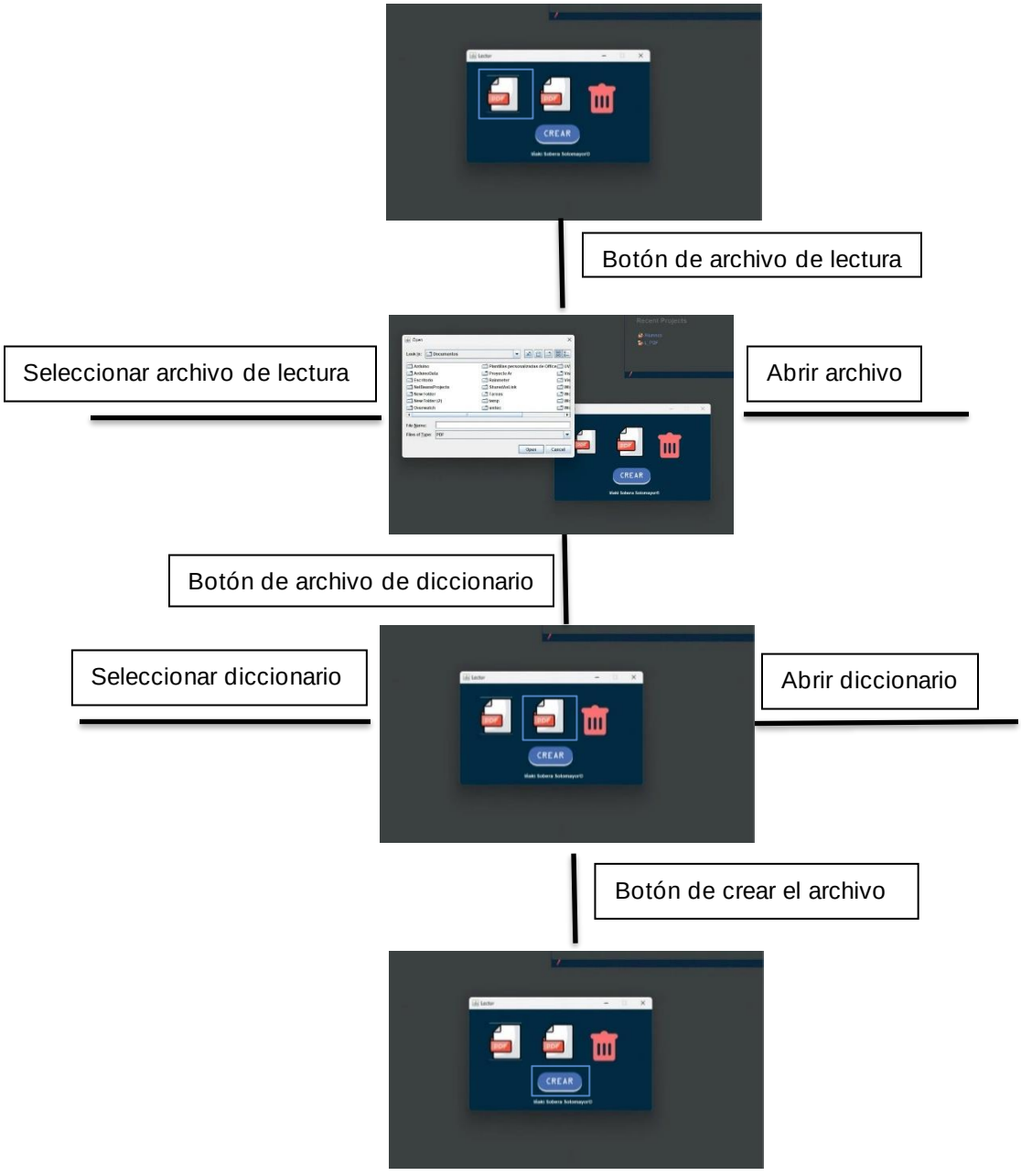
Nuevo Archivo PDF: Contiene las palabras comunes encontradas en los dos documentos originales.
Información de Salida:
Fecha de generación del nuevo PDF.
Rutas de los archivos comparados.
Artefactos Internos:
Lista de Palabras Comunes: Almacena las palabras que se encuentran en ambos documentos.
Formato del Nuevo PDF: Define la fuente, tamaño de letra, márgenes, etc. del archivo generado.

Descripción

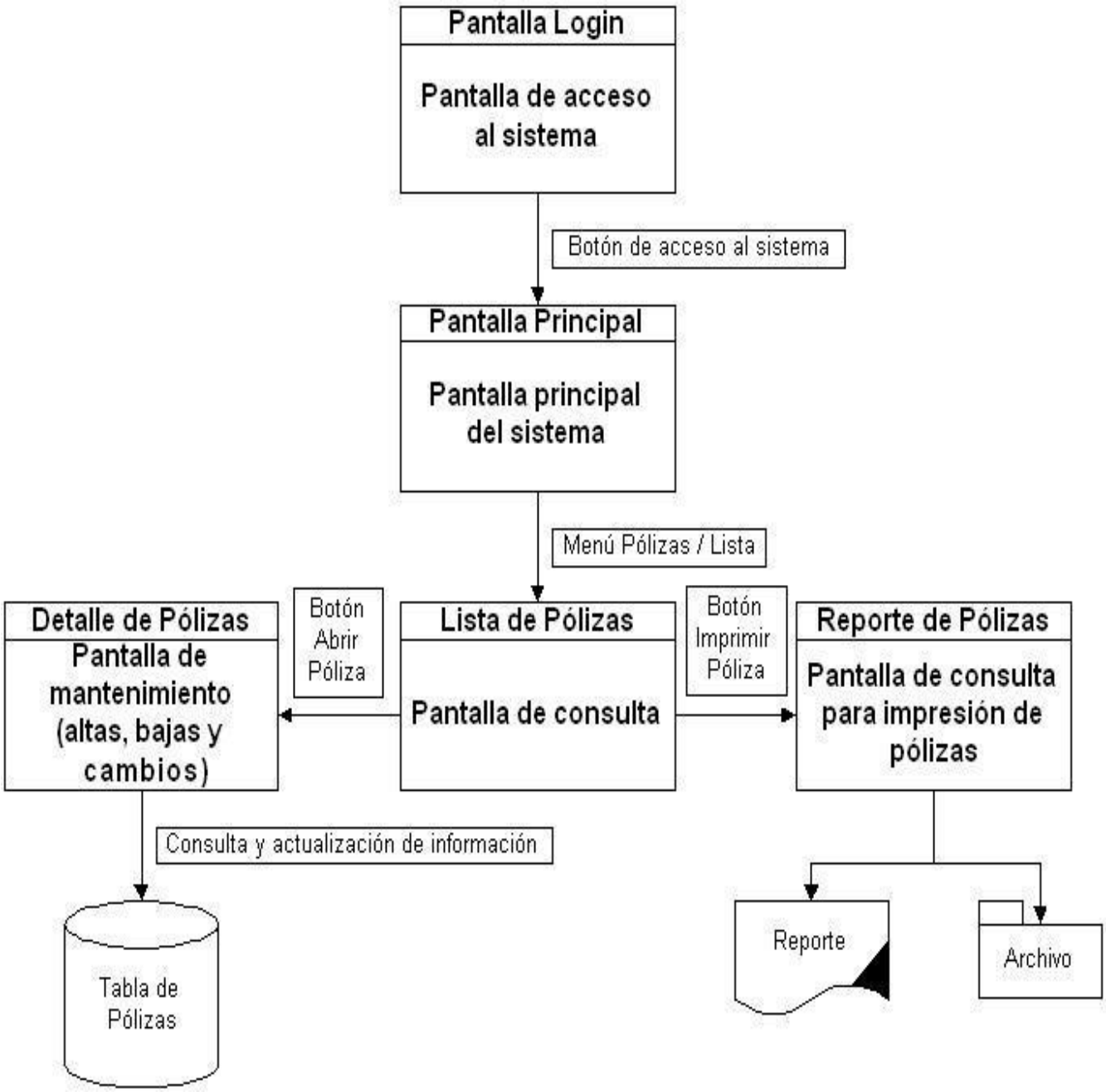
VII. Definición de las Pantallas del Sistema

No aplica en este sistema

Diagrama de Navegación de Pantallas



Ejemplo de un diagrama de interfaz de usuario:



Definición de Pantallas

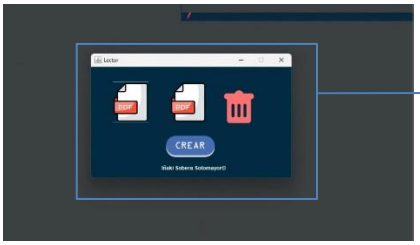
Pantalla: Inicio de Sesión

Objetivo de la Pantalla

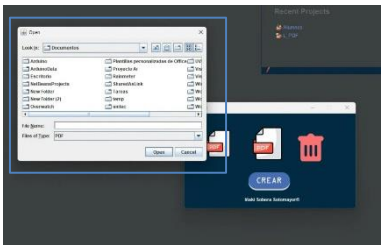
Esta pantalla permite el acceso al sistema solo a usuarios autorizados, validando los campos de usuario y contraseña.

Descripción de la Pantalla

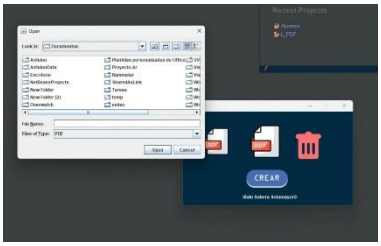
Campo	Tipo	Comentario
Usuario	Alfanumérico	Nombre del usuario.
Contraseña	Alfanumérico	Password del usuario.
Recordar Contraseña	Booleano	Indica si el usuario desea que su nombre y contraseña se encuentren presentes cada vez que acceda al sistema.

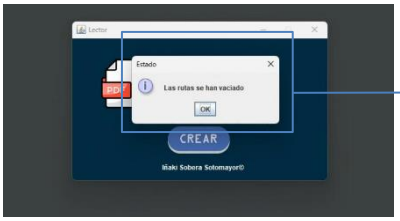


Pantalla de inicio: consta de 4 botones, el primero selecciona el documento de lectura, el segundo es el diccionario, el tercero borra las rutas de los documentos y el ultimo crea el archivo con las palabras en común entre ambos



Ventana de selección para el archivo de lectura, esta ventana fue desarrollada a partir de una libraría llamada JFilechooser
Esta librería crea la ventana que se muestra en la imagen, la misma fue modificada en el código para solo seleccionar archivos PDF, al hacerlo guarda la ruta del documento en una variable que se llama ruta 1





Esta ventana fue creada de la misma forma que la anterior pero esta guarda la ruta en una variable ruta 2 haciendo la diferente a la anterior

Esta ultima ventana seria la ventana de confirmación se acciona al darle darle clic al botón de crear este es el botón mas importante ya que guarda la funcionalidad del proyecto

VIII. Definición de interfaces del sistema

El sistema analizado no utiliza componentes de servicios comunes ni interactúa con otros subsistemas. Por lo tanto, no se requieren interfaces formales en este caso.