MANUAL TECNICO

GENERADOR DE DOCUMENTOS PARA EL DEPARTAMENTO DE METAL-MECANICA

ALUMNOS:

Rodriguez Vázquez Carlos 19221865

Carbino Ramírez Jesús 19221900

Versión 1.0

Fecha: 13/03/2023

Contenido

PRESENTACION 3
RESUMEN 3
OBJETIVO 3
FINALIDAD DEL MANUAL 3
1.- ASPECTOS TÉCNICOS 4
1.1.- HERRAMIENTAS UTILIZADAS PARA EL DESARROLLO 4
1.1.1 APACHE NETBEANS IDE 4
1.1.2 GITHUB 5
2.- CONFIGURACIÓN DE ARCHIVOS 5
3.- BLOQUES DE CODIGO 7
BLOQUES DE CODIGO MAIN 7
CONFIGURAR TAMAÑO DEL PANEL 7
CAMBIAR PANEL MEDIANTE EL MENU 8
BLOQUES DE CODIGO PARA DOCUMENTOS 9
SELECCIONAR DOCUMENTO BASE 9
MODIFICAR TABLAS 11
MENCIONES DE CODIGO EXTRA CON EXPLICACIÓN BREVE 12
LIBRERIAS UTILIZADAS EN LA VERSION 1.0 12
4.- REQUERIMIENTOS DEL SOFTWARE 13
4.1 REQUISITOS MINIMOS 13
BIBLIOGRAFIA 14

PRESENTACION

El siguiente manual se ha desarrollado con la finalidad de dar a conocer la información necesaria para realizar el mantenimiento, instalación y exploración de software 'Generador de Documentos' , el cual consta de diferentes actividades para mejorar y facilitar la generación de documentos necesarios para distintas actividades académicas dentro de nuestro Instituto Tecnológico de Puebla.

El manual ofrece la información necesaria de Como está realizado el software, para que un futuro desarrollador que quiera editar el software lo haga de una manera adecuada al desarrollo del aplicativo.

RESUMEN

El manual detalla los aspectos técnicos e informáticos del software 'Generador de Documentos' con la finalidad de explicar la estructura del aplicativo al personal que quiera editarlo o configurarlo. La siguiente guía se encuentra dividida en las herramientas que se utilizaron para la creación del software con una breve explicación paso a paso, la configuración que se debe darle a los archivos que se modifican, así como el funcionamiento de bloques de código para su futura modificación.

El aplicativo maneja diferentes funcionalidades el cual requieren de hardware y software el cual se explicará que funcionamiento realiza cada uno de ellos, dando sugerencias para el debido uso del sistema de información.

OBJETIVO

Dar a conocer el uso adecuado del software 'Generador de Documentos' en aspectos técnicos de manera descriptiva e ilustrada sobre los componentes y funcionalidades que conforman el buen funcionamiento del sistema de información.

FINALIDAD DEL MANUAL

La finalidad de este manual técnico es instruir a la persona que quiera administrar, editar o configurar el software 'Generador de Documentos' usando las debidas herramientas.

1.- ASPECTOS TÉCNICOS

El objetivo de este programa es facilitar la creación de documentos académicos para el usuario, permitiendo ingresar la información requerida de manera sencilla y generando el documento de forma rápida y eficiente. El programa busca mejorar la productividad del usuario al reducir el tiempo y esfuerzo necesario para la creación de documentos, a través de una interfaz intuitiva y amigable que simplifica el proceso de creación y edición. Al utilizar este programa, el usuario puede generar documentos de alta calidad en menos tiempo y con menos esfuerzo, lo que le permite concentrarse en otras tareas académicas importantes.

Se recomienda que el siguiente manual sea manipulado únicamente por la persona que quiera administrar, editar o configurar el software, para velar por la seguridad de los datos que se almacenan, ya que pueden ser usados para otros fines.

1.1.- HERRAMIENTAS UTILIZADAS PARA EL DESARROLLO

En esta sección se procede a explicar las herramientas informáticas empleadas para el desarrollo del aplicativo.

1.1.1 APACHE NETBEANS IDE

Apache NetBeans IDE es un entorno de desarrollo integrado (IDE) de código abierto utilizado para desarrollar aplicaciones en lenguajes como Java, C++, PHP y HTML5. Fue desarrollado inicialmente por Sun Microsystems en 1996 como NetBeans y posteriormente adquirido por Oracle. En 2016, Oracle donó NetBeans a la Apache Software Foundation, donde ahora se conoce como Apache NetBeans.

NetBeans IDE proporciona un conjunto completo de herramientas para el desarrollo de software, incluyendo un editor de código, depurador, diseñador de interfaces gráficas de usuario (GUI), administrador de proyectos y herramientas para refactorización y pruebas. Además, es altamente personalizable y compatible con múltiples plataformas, lo que lo convierte en una herramienta popular entre los desarrolladores de software.

Una de las características más útiles de NetBeans IDE es su capacidad para integrarse con varios sistemas de control de versiones, como Git y Subversion, lo que permite a los desarrolladores trabajar en equipo y colaborar en proyectos de software.

1.1.2 GITHUB

GitHub es una plataforma en línea que permite a los usuarios alojar y compartir proyectos de software utilizando el sistema de control de versiones Git. En otras palabras, es una plataforma que permite a los desarrolladores y programadores trabajar en conjunto en un mismo proyecto, seguir y controlar los cambios realizados en el código fuente, y colaborar en la resolución de problemas y mejoras.

Los proyectos alojados en GitHub son llamados "repositorios" y cada uno tiene su propio espacio de trabajo en línea donde los usuarios pueden ver, editar y descargar el código fuente. GitHub también proporciona herramientas de colaboración, como la posibilidad de hacer comentarios en el código y de hacer "pull requests" (solicitudes de incorporación de cambios) para que otros usuarios revisen y aprueben los cambios.

Además de proyectos de software, GitHub también se utiliza para alojar proyectos de documentación y de diseño, así como proyectos relacionados con ciencia de datos, inteligencia artificial y otros campos técnicos.

2.- CONFIGURACIÓN DE ARCHIVOS

El programa genera archivos Word mediante plantillas preestablecidas que han sido configuradas previamente para la inserción de información específica en campos del documento, los cuales son ingresados por el usuario dentro del programa. Las plantillas se ubican en una carpeta oculta dentro de la carpeta raíz del programa llamada "documentos_base", la cual alberga todos los archivos necesarios.

Para realizar esta tarea, se utiliza la librería Apache Poi para la lectura y desglose de la información contenida en los archivos Word. El programa recorre el documento palabra por palabra, detectando las palabras clave que indican dónde se desea insertar la información ingresada por el usuario. Estas palabras clave son delimitadas por signos de porcentaje, como por ejemplo %FECHA%, que señala el lugar en el que se insertará la fecha generada por el programa mediante código.

Es importante destacar que estas palabras clave deben ser legibles y comprensibles, ya que se referenciarán mediante código y resultará más fácil su lectura y entendimiento en el futuro.

Ejemplo de la configuración de un documento:

AT'N.: PAOLA CRISTAL SERRANO ORTIZ
COORDINADORA DE APOYO A TITULACIÓN

Por este conducto me permito solicitar a usted por motivos de entrega de oficios de su área en periodos diferentes, el cambio de **JURADO** mediante la opción de Titulación Integral de maneara INDIVIDUAL del alumno: %NOMBRE% quedando de la siguiente manera:

Presidente:	**%PRESIDENTE%**
Secretario:	**%SECRETARIO%**
Vocal Propietario:	**%VOCALPROPIETARIO%**

Agradeciendo de antemano su valioso apoyo, quedo de usted.

En el caso de que el documento contenga tablas y se requiera modificar la información dentro de ellas, es importante destacar que la técnica de recorrer el texto palabra por palabra no es capaz de leer la información que se encuentra en las tablas. Por lo tanto, se lleva a cabo una extracción de las tablas y se modifican directamente en su interior utilizando una estructura de matriz. Para lograr esto, es necesario que el campo que requiere modificaciones también tenga su palabra clave delimitada por signos de porcentaje, con el fin de normalizar las palabras clave y saber que esta sección debe ser modificada. En secciones posteriores del documento, se detallará la técnica para modificar tablas, incluyendo ejemplos de código.

Es importante destacar que en el caso de que algún campo dentro de la tabla se encuentre vacío, la ejecución del código no será posible y se producirá un error lógico debido al funcionamiento de la técnica de programación utilizada. Para evitar esta situación, se requiere que cada campo dentro de la tabla cuente con al menos un carácter válido.

Sin embargo, se recomienda encarecidamente continuar con la normalización de palabras clave en todos los campos, incluyendo los que se encuentran dentro de las tablas. Esto permitirá una gestión más eficiente y coherente de la información, y reducirá la probabilidad de errores en la generación de documentos. En caso de que surja alguna duda o dificultad en este proceso, se brindará orientación y apoyo a través de la documentación y el soporte técnico correspondiente.

3.- BLOQUES DE CODIGO

Una vez que el usuario ha cargado el proyecto en el entorno de desarrollo integrado Apache NetBeans, se encontrará con dos paquetes denominados "Package" e "ImagePackage". El paquete ImagePackage alberga todos los recursos de imágenes que se utilizan dentro del proyecto, mientras que el paquete "Package" es el paquete principal que contiene una amplia variedad de JPanels, cada uno representando un documento diferente, y conteniendo su propio código específico para dicho documento.

En cuanto al diseño de la interfaz gráfica de usuario (GUI), solo se dispone de un JFrame Form, denominado Main, que sirve como menú en blanco en el cual el usuario puede seleccionar qué JPanel desea mostrar, y en consecuencia, este será desplegado en pantalla tras ser pintado y revalidado.

BLOQUES DE CODIGO MAIN

CONFIGURAR TAMAÑO DEL PANEL

Incluir el nuevo Panel en el array de Paneles es crucial para una gestión de código efectiva y evitar su repetición. De esta forma, se pueden configurar todos los Paneles de manera uniforme y eficiente, sin perder la consistencia del código. Además, esto permite una fácil identificación y localización de los Paneles en el proyecto.

```
        desCheckTodo();

        JPanel[] pages = {page1, page2, page3, page4, page5,
                          page6, page7,page8,page9,page10};
        int x = 0;
        int y = 0;
        int width = 899;
        int height = 577;

        for (JPanel page : pages) {
            page.setSize(width, height);
            page.setLocation(x, y);
        }

        content.removeAll();

        this.setLocationRelativeTo(null);
        this.setVisible(true);
        this.setResizable(false);
```

CAMBIAR PANEL MEDIANTE EL MENU

Es crucial garantizar que, al agregar un nuevo documento, se incluya su código correspondiente para que se pueda mostrar en pantalla al hacer clic en él. Esto asegura una buena organización y gestión del código, evitando la repetición de código y mejorando la eficiencia del programa. Además, la inclusión de un nuevo documento sin su código correspondiente podría llevar a errores y fallos en el funcionamiento del programa.

```
private void cambiarPanel(JCheckBoxMenuItem checkBox, JPanel panel) {
desCheckTodo();
checkBox.setSelected(true);
content.removeAll();
content.add(panel);
content.repaint();
content.revalidate();
}

private void jCheckBoxMenuAsignacionDeJuradoActionPerformed(java.awt.event.ActionEvent evt) {
    cambiarPanel(jCheckBoxMenuAsignacionDeJurado, page1);
}

private void jCheckBoxMenuAsignacionDeRevisoresActionPerformed(java.awt.event.ActionEvent evt) {
    cambiarPanel(jCheckBoxMenuAsignacionDeRevisores, page2);
}

private void jCheckBoxMenuAutorizacionDeImpresionTesisActionPerformed(java.awt.event.ActionEvent evt) {
    cambiarPanel(jCheckBoxMenuAutorizacionDeImpresionTesis, page3);
}

private void jCheckBoxMenuAsesorInternoActionPerformed(java.awt.event.ActionEvent evt) {
    cambiarPanel(jCheckBoxMenuAsesorInterno, page4);
}
```

BLOQUES DE CODIGO PARA DOCUMENTOS

SELECCIONAR DOCUMENTO BASE

Es fundamental destacar que en cada Panel, se hace referencia a una ubicación específica en la computadora, tanto en la carpeta "documentos_base" (dos veces) como en la carpeta "documentos_generados" (una vez). Es esencial prestar especial atención a las referencias de la carpeta "documentos_base", ya que es la ubicación donde el programa obtendrá el archivo Word previamente configurado con sus palabras clave para insertar la información.

```
try {
    // Abrir el archivo de Word
    String fileName =  "documentos_base/LIBERACION_DE_ACTIVIDADES_ACADEMICAS.docx";
    File file = new File(fileName);
    String absolutePath = file.getAbsolutePath();
    XWPFDocument doc = new XWPFDocument(new FileInputStream(absolutePath));
```

Es crucial asegurarse de que la ubicación dentro de la carpeta "documentos_generados" sea precisa, ya que esta es la ubicación donde se guardará el archivo resultante. Se recomienda crear una nueva carpeta dentro de la carpeta "documentos_generados" con un nombre descriptivo para el nuevo archivo que se generará, con el fin de mantener un orden y evitar la confusión que podría surgir de la combinación de distintos archivos en una misma ubicación. Esta práctica contribuirá a simplificar la búsqueda de archivos en el futuro.

Además, es importante tener en cuenta que el nombre del archivo generado debe ser único y no existir previamente en la carpeta de documentos generados. De lo contrario, se podría sobrescribir información importante y generar conflictos en el futuro. Es recomendable incluir algún tipo de identificador único en el nombre del archivo, como una fecha o un número de versión, para evitar posibles confusiones. Sin embargo, una función que detecte si ya se ha generado un documento previamente con el mismo nombre no está integrada en la versión 1.0. Por lo cual en esta versión será trabajo del usuario el no colocar un mismo nombre al documento.

```
// Guardar los cambios en el archivo

String nombreGenerado = jTextNombreArchivo.getText();
String outputPath = "documentos_generados/LIBERACION DE ACTIVIDADES ACADEMICAS/"+nombreGenerado+".docx";
File outputFile = new File (outputPath);
doc.write(new FileOutputStream(outputFile));
doc.close();

Desktop.getDesktop().open(new File(outputPath));
limpiarCampos();
```

RECORRER EL DOCUMENTO E INTERCAMBIAR PALABRAS CLAVE

El código utilizado para recorrer el documento utiliza la biblioteca Apache POI para acceder al contenido de un documento de Word y reemplazar ciertos textos con información ingresada por el usuario. El código recorre todos los párrafos en el documento y, dentro de cada párrafo, recorre cada run (trozo de texto que no contiene saltos de línea) y comprueba si el texto contiene alguna de las palabras clave que se deben reemplazar. Si el texto contiene una de las palabras clave, se reemplaza por el valor correspondiente ingresado por el usuario y se actualiza el texto en el documento.

```
for (XWPFParagraph p : doc.getParagraphs()) {
    for (XWPFRun r : p.getRuns()) {
        String text = r.getText(0);
        System.out.println("Texto actual: " + text);
        try{
            if (text.contains("%DOCENTE%")) {
                text = text.replace("%DOCENTE%", jComboBoxDocente.getSelectedItem().toString());
                r.setText(text, 0);
                System.out.println("Palabra reemplazada con éxito");
            }
            if (text.contains("%PERIODO%")) {
                text = text.replace("%PERIODO%", jComboBoxPeriodo.getSelectedItem().toString()+" "+añoActual);
                r.setText(text, 0);
                System.out.println("Palabra reemplazada con éxito");
            }
            if (text.contains("%FECHA%")) {
                text = text.replace("%FECHA%", jTextFecha.getText().toUpperCase());
                r.setText(text, 0);
                System.out.println("Palabra reemplazada con éxito");
            }
        }catch(Exception e){
        }

    }
}
```

En este ejemplo en particular, el código reemplaza "%DOCENTE%" con el valor seleccionado en un JComboBox llamado "jComboBoxDocente", "%PERIODO%" con el valor seleccionado en otro JComboBox llamado "jComboBoxPeriodo" concatenado con el año actual, y "%FECHA%" con el valor ingresado en un JTextfield llamado "jTextFecha". Si se produce alguna excepción durante el proceso de reemplazo, el código simplemente la captura y la omite.

El presente código se duplica en cada JPanel correspondiente a un caso de documento diferente, lo que resulta en una cantidad variable de condicionales para cada palabra clave. En la versión 1.0 del programa, se utiliza una cantidad significativa de instrucciones "if", sin embargo, en futuras actualizaciones, se considera la posibilidad de sustituir estas estructuras condicionales por un mapa hash con el fin de optimizar su rendimiento. Cabe destacar que, a pesar de la simplicidad del programa y de no manejar grandes volúmenes de información, se mantiene un rendimiento satisfactorio, lo que lo convierte en una herramienta eficaz para aquellos usuarios principiantes en programación en Java o nuevos en la librería empleada.

MODIFICAR TABLAS

Cuando se trata de modificar tablas en un documento, el proceso difiere del método de recorrer la información del mismo. En este caso, todas las tablas del documento son extraídas en orden de aparición y almacenadas en una Lista. A continuación, se puede modificar una tabla específica mediante la extracción de esta de la Lista y su posterior modificación utilizando el siguiente código:

```
// Primer tabla
List<XWPFTable> tables = doc.getTables();
XWPFTable table = tables.get(1); // obtener la primer tabla del documento
XWPFTableCell cell = table.getRow(0).getCell(0); // obtener la celda en la fila 0 y columna 0 de la tabla
// Obtener el párrafo y el objeto XWPFRun que contiene el texto que deseas modificar
XWPFParagraph paragraph = cell.getParagraphArray(0);
XWPFRun run = paragraph.getRuns().get(0);
String newText = "PUEBLA PUE. a "+jTextFecha.getText().toUpperCase();
run.setText(newText, 0);
```

Es fundamental que el campo de la tabla en cuestión contenga al menos un carácter o la palabra clave necesaria para la que hemos estado trabajando desde el inicio del documento, a fin de que pueda ser modificada y el programa se ejecute. En caso contrario, el programa no se ejecutará y si se incluye dentro de un bloque try-catch, no se modificará la información.

MENCIONES DE CODIGO EXTRA CON EXPLICACIÓN BREVE

estaLleno()

Esta función verifica que todos los campos dentro del programa han sido llenados por el usuario, caso contrario saldrá un aviso para invitar al usuario a terminar de llenar los campos.

limpiarCampos()

Esta función vaciar todos los campos después de haber generado un documento el usuario, insertando texto vacío a todos los campos. Esto con la finalidad de evitar que el usuario tenga que borrar manualmente los campos y así mejorar la eficiencia.

buttonFecha:

Es importante verificar que el Sistema Operativo (SO) tenga configurada la fecha y hora correctas, ya que la función del botón extrae la fecha actual del SO para su uso en el documento. Si la fecha del SO es incorrecta, la fecha en el documento generado también será incorrecta.

UIManager.setLookAndFeel(new FlatLightLaf());

FlatLightLaf.setup();

Estas dos líneas de código son para configurar el diseño FlatLightLaf en el programa. Al incluir estas líneas de código y sus correspondientes importaciones de librerías, es posible que se muestren algunos errores de código, los cuales el IDE NetBeans puede solucionar rápidamente mediante su sistema de ayuda (destacados en amarillo).

LIBRERIAS UTILIZADAS EN LA VERSION 1.0

```
import java.awt.Desktop;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.Calendar;
import java.util.List;
import java.util.Locale;
import javax.swing.DefaultComboBoxModel;
import javax.swing.JOptionPane;
import org.apache.poi.xwpf.usermodel.XWPFDocument;
import org.apache.poi.xwpf.usermodel.XWPFParagraph;
import org.apache.poi.xwpf.usermodel.XWPFRun;
import org.apache.poi.xwpf.usermodel.XWPFTable;
import org.apache.poi.xwpf.usermodel.XWPFTableCell;
import com.formdev.flatlaf.FlatLightLaf;
import javax.swing.UIManager;
import javax.swing.UnsupportedLookAndFeelException;
```

4.- REQUERIMIENTOS DEL SOFTWARE

En esta sección se detallará los requisitos mínimos del sistema para poder ejecutar el aplicativo usado para modificar el software “Generador de Documentos”.

4.1 REQUISITOS MINIMOS

*** Microsoft Windows 7 / 8 / 10 / 11 o Ubuntu 15.04:**

* Procesador Intel Core i3 o superior
* Memoria 2GB (32 bits), 4 GB(64 bits)
* Espacio en disco: 1.5GB de espacio libre en el disco

*** OS X 10.10 Intel:**

* Procesador: Intel de doble núcleo o superior
* Memoria: 4GB
* Espacio en disco: 1.5GB de espacio libre en el disco

BIBLIOGRAFIA

Apache NetBeans: https://netbeans.apache.org/download/index.html

Java: https://www.java.com/en/download/

GitHub: https://github.com/