UNIVERSIDAD AUTONOMA DEL ESTADO DE MÉXICO

FACULTAD DE INGENIERIA

Sistemas Operativos:

Proyecto 1: Simulador de Memoria con reemplazo de paginas NRU

Alumno:

Victor Manuel Jiménez Rosas

**Introducción**

Para la realización del proyecto se utilizó el sistema operativo Windows y el lenguaje de programación Java debido a la facilidad para trabajar con interfaz de usuario.

**Instalación de Java**

Para la instalación de Java es necesario descargar el archivo desde https://www.java.com/es/download/ie\_manual.jsp y luego seguir la instalación recomendada por la aplicación.

**Configuración**

El programa usa dos archivos, uno de configuración llamado “**{nombre}**.conf” donde **{nombre}** hace referencia al nombre del archivo que tiene la siguiente estructura:

// memset virt page # physical page # R (read from) M (modified) inMemTime (ms) lastTouchTime (ms)

memset **{numero} {numero} {numero} {numero} {numero} {numero}**

//tamaño de pagina

tamanopag **{numero}**

// Paginas Virtuales

numpag **{numero}**

// Paginas Físicas

marcpag **{numero}**

Siendo **{numero}** un numero entero cualquiera y siendo que el máximo numero de marcos y paginas virtuales es 64, un numero mayor será establecido en 64.

Y un archivo txt cualquiera que tiene la siguiente estructura:

// Insertar el comando REA/WRITE en este archivo

// READ

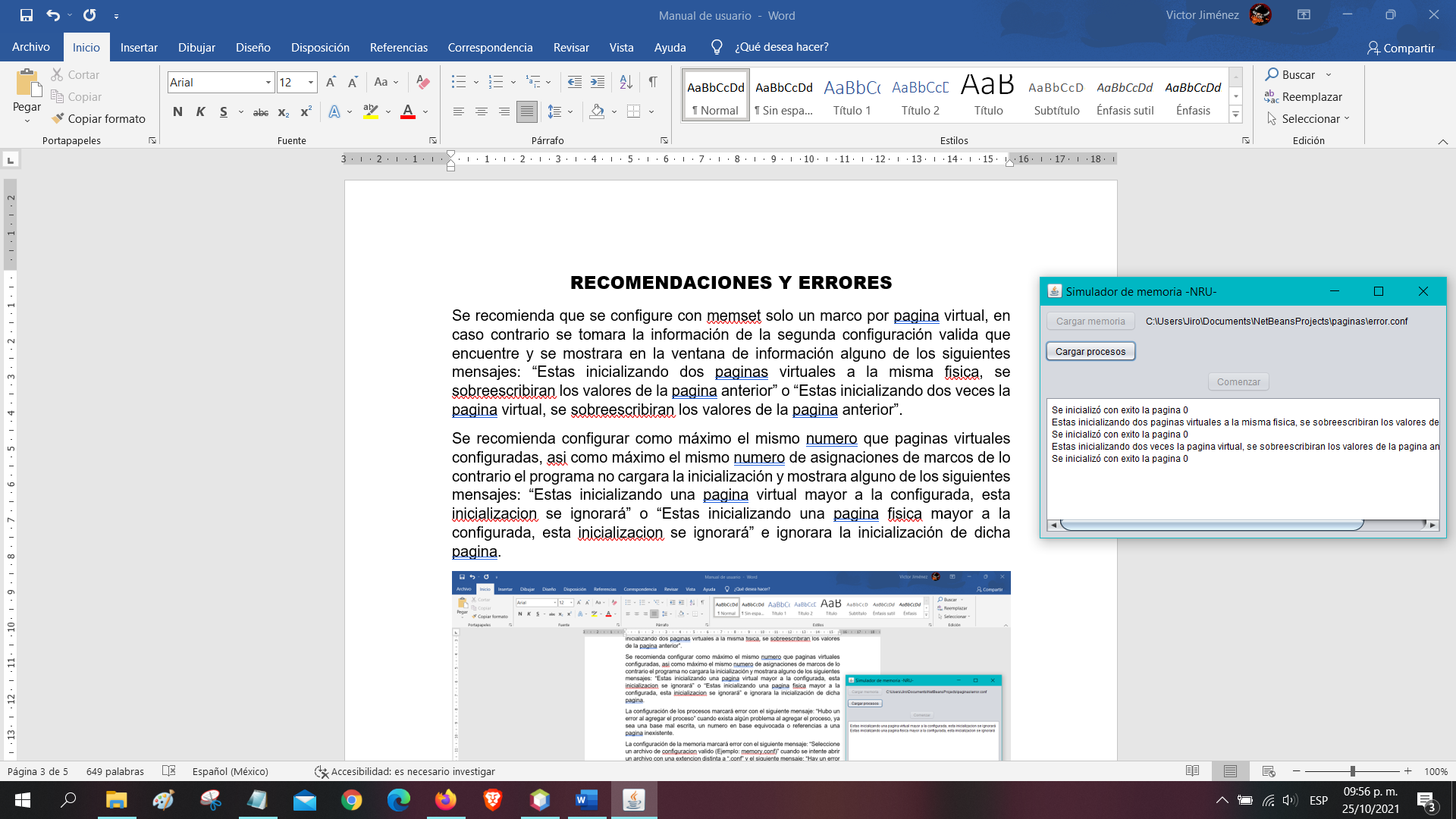
// WRITE

**{operacion}** **{base} {numero}**

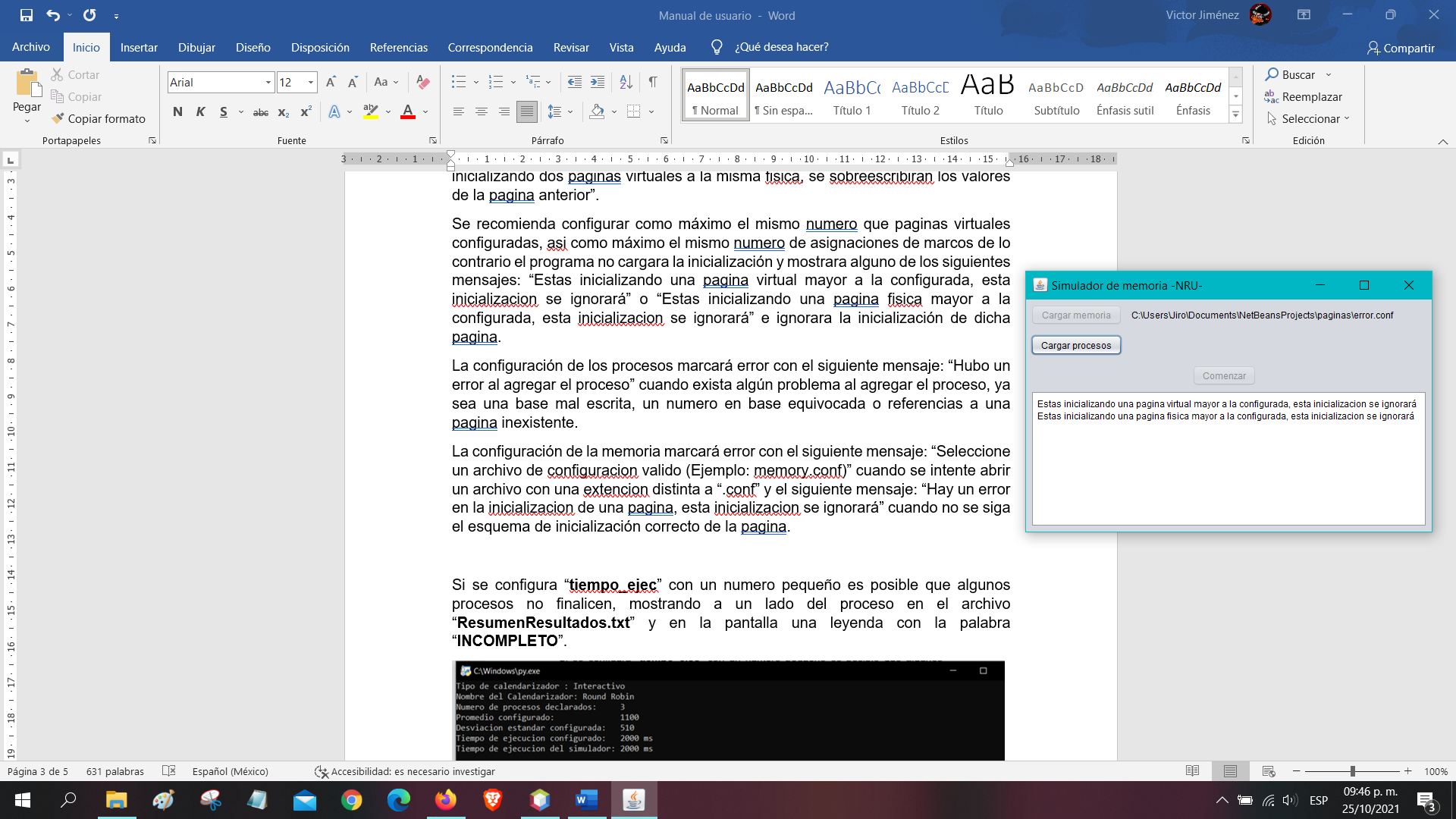
Siendo **{operacion}** “READ” o “WRITE”, **{base}** “BIN”, “HEX”, “OCT” o “DEC” siendo opcional y **{numero}** un numero en la base especificada o la palabra “aleatorio”.

RECOMENDACIONES Y ERRORES

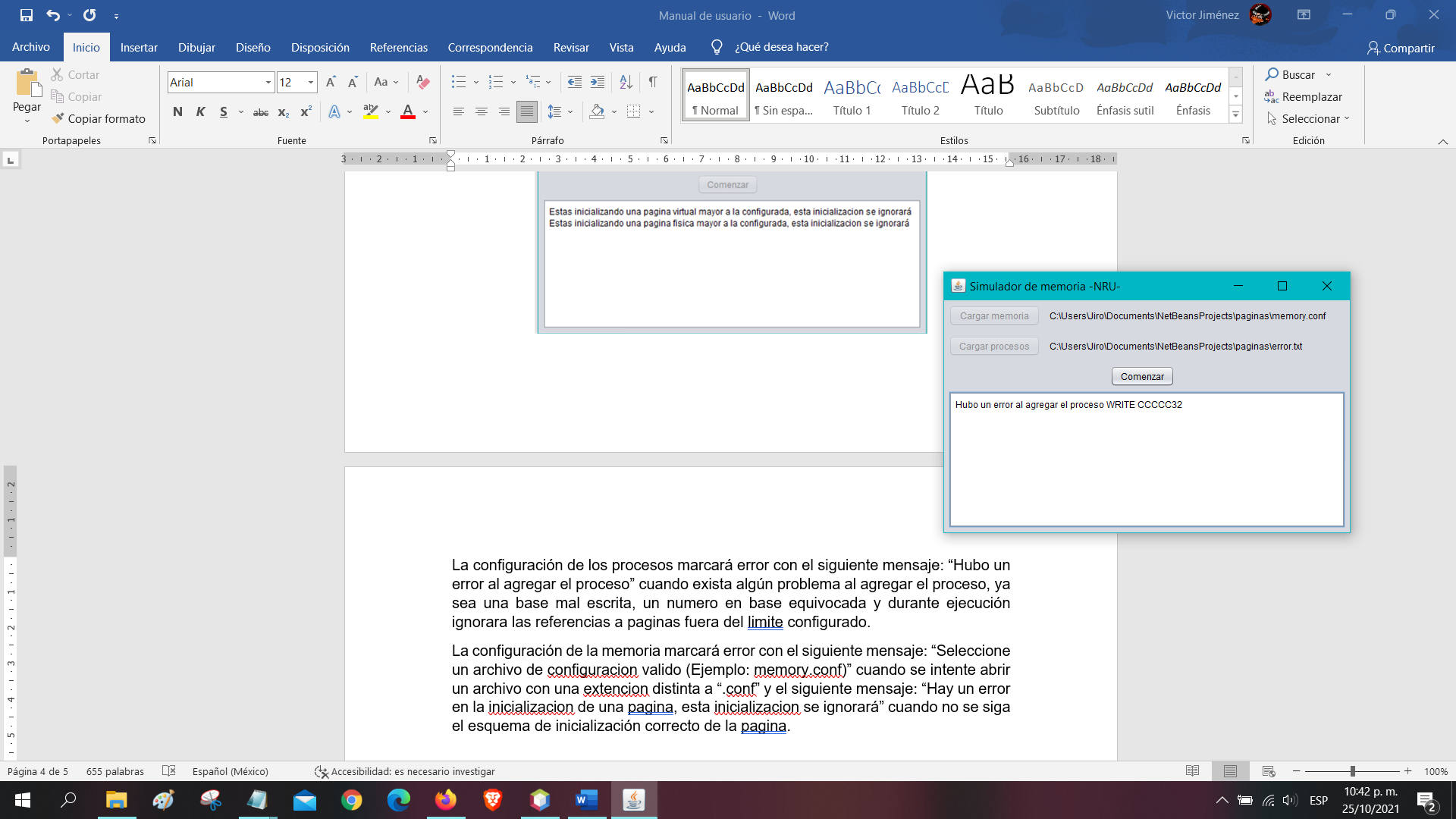
Se recomienda que se configure con memset solo un marco por pagina virtual, en caso contrario se tomara la información de la segunda configuración valida que encuentre y se mostrara en la ventana de información alguno de los siguientes mensajes: “Estas inicializando dos paginas virtuales a la misma fisica, se sobreescribiran los valores de la pagina anterior” o “Estas inicializando dos veces la pagina virtual, se sobreescribiran los valores de la pagina anterior”.



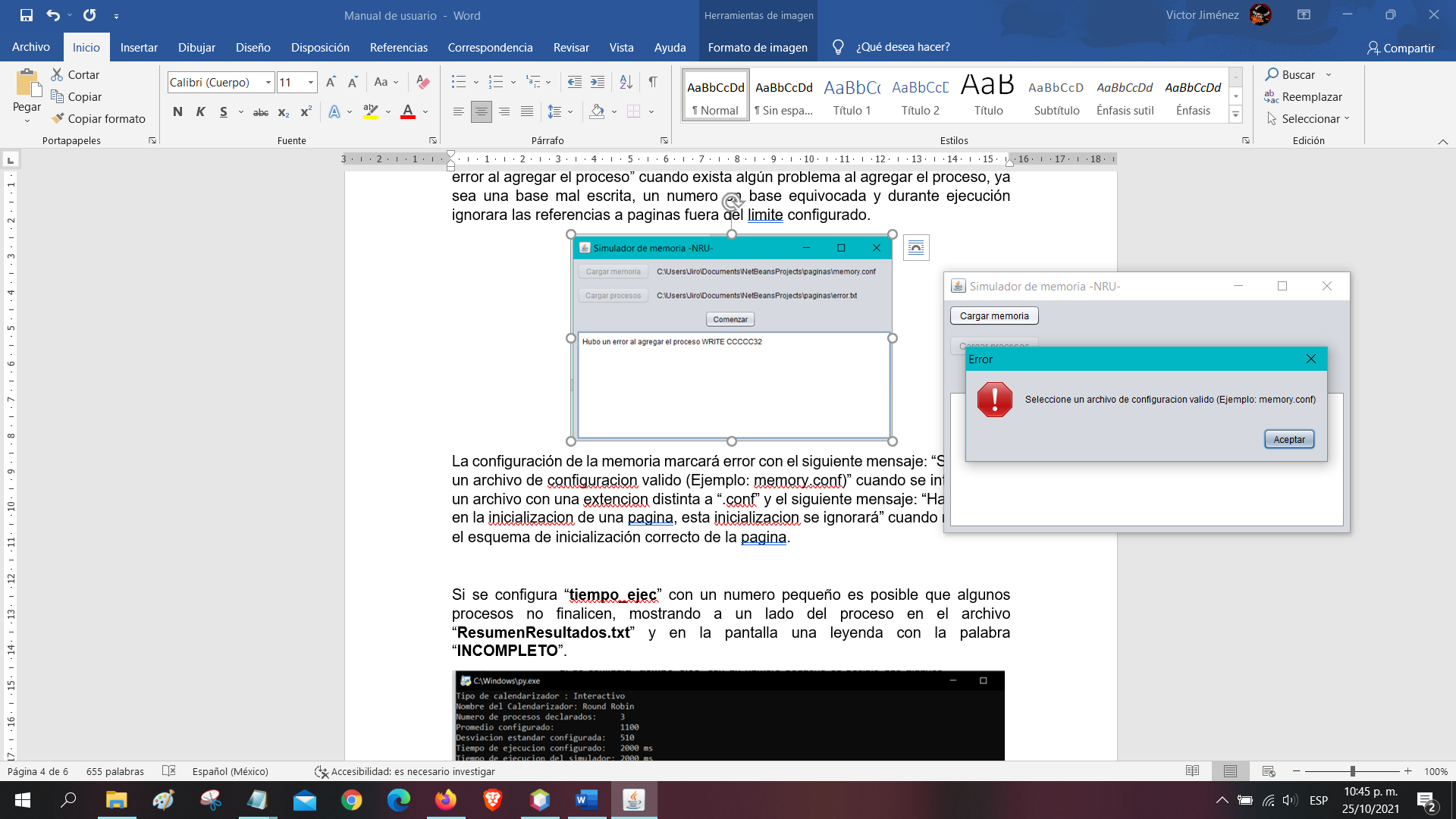
Se recomienda configurar como máximo el mismo numero que paginas virtuales configuradas, asi como máximo el mismo numero de asignaciones de marcos de lo contrario el programa no cargara la inicialización y mostrara alguno de los siguientes mensajes: “Estas inicializando una pagina virtual mayor a la configurada, esta inicializacion se ignorará” o “Estas inicializando una pagina fisica mayor a la configurada, esta inicializacion se ignorará” e ignorara la inicialización de dicha pagina.



La configuración de los procesos marcará error con el siguiente mensaje: “Hubo un error al agregar el proceso” cuando exista algún problema al agregar el proceso, ya sea una base mal escrita, un numero en base equivocada y durante ejecución ignorara las referencias a paginas fuera del limite configurado.

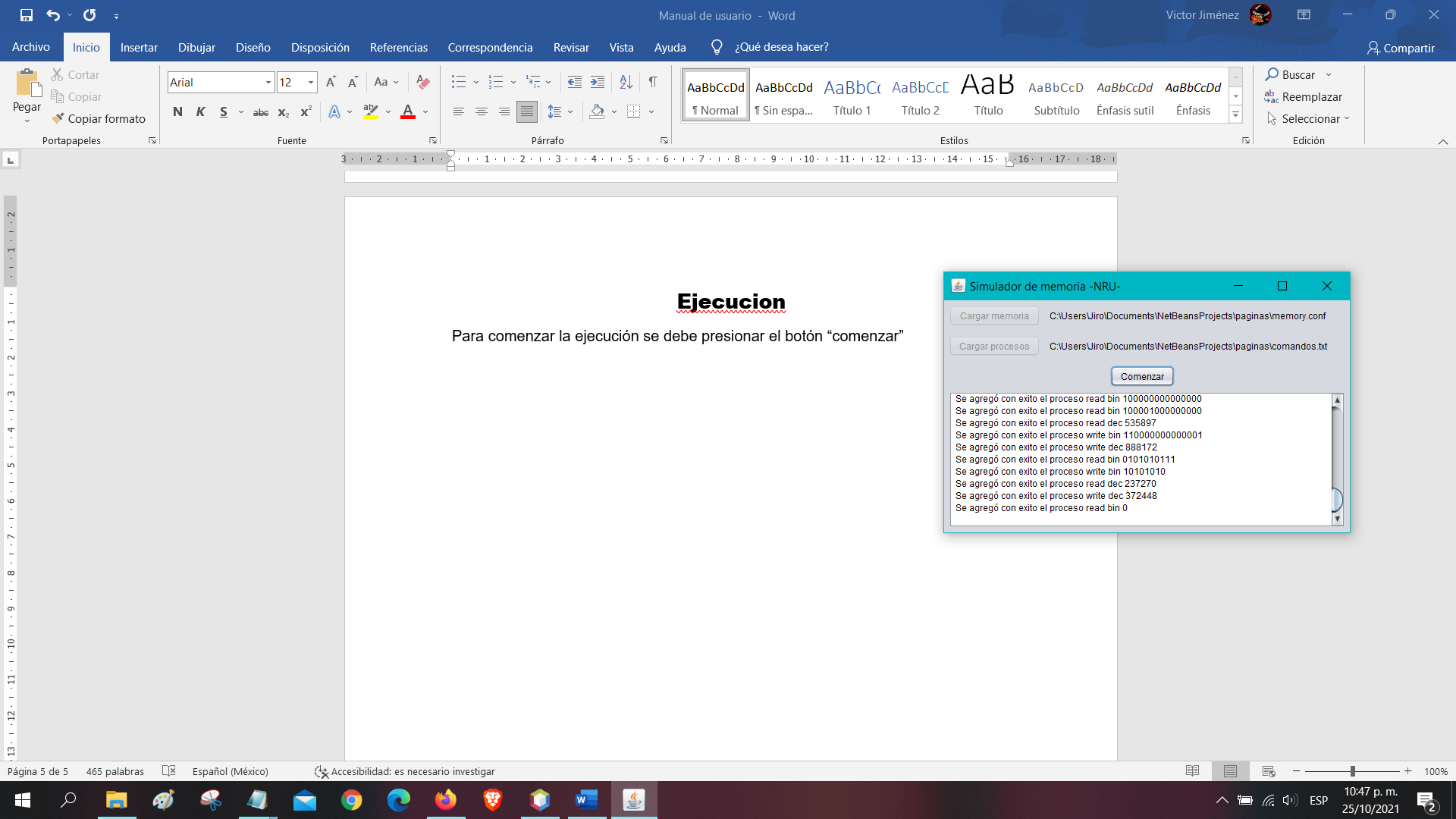


La configuración de la memoria marcará error con el siguiente mensaje: “Seleccione un archivo de configuracion valido (Ejemplo: memory.conf)” cuando se intente abrir un archivo con una extencion distinta a “.conf” y el siguiente mensaje: “Hay un error en la inicializacion de una pagina, esta inicializacion se ignorará” cuando no se siga el esquema de inicialización correcto de la pagina.

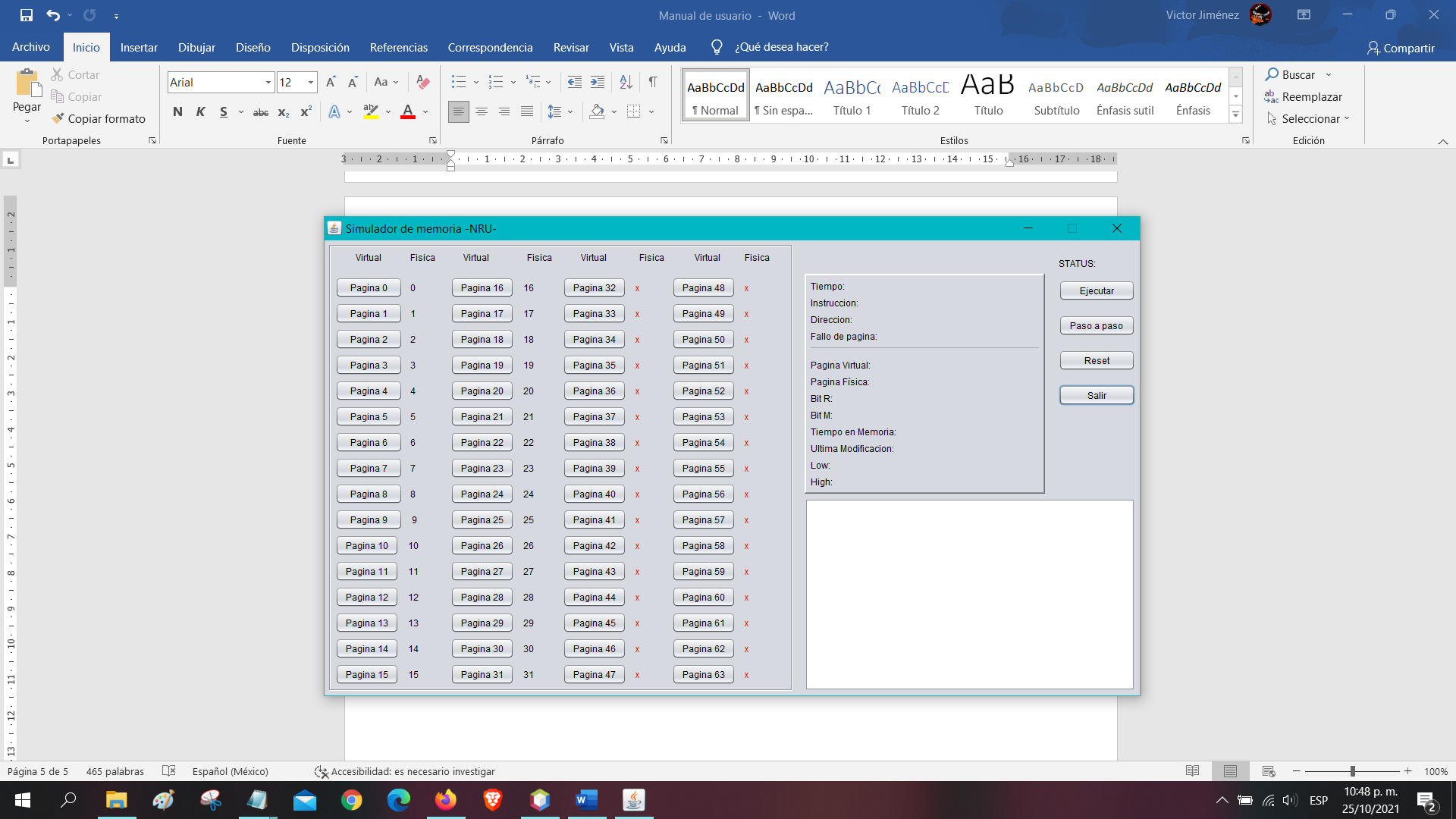


Ejecucion

Para comenzar la ejecución se debe presionar el botón “comenzar”



La ventana mostrara una representación de una memoria de 64 paginas y 64 marcos además de una sección de información, una ventana que muestra la salida de cada operación y 4 botones para controlar su comportamiento.



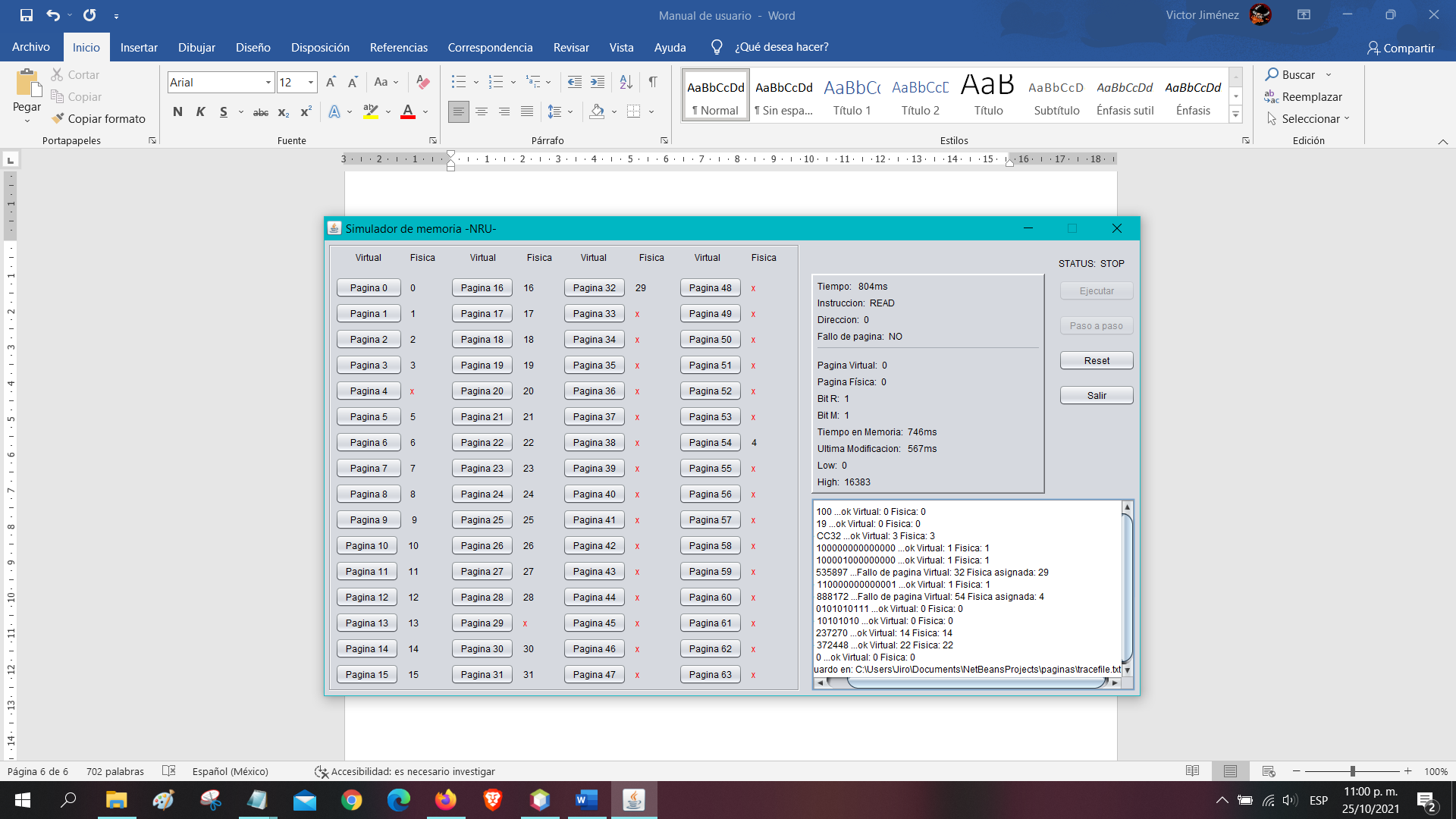
Al presionar el botón “Ejecutar” se bloquearan los demás botones hasta el termino de la ejecución mostrando en la ventana de información cada proceso.

Al presionar el botón “Paso a paso” se recorrerán los procesos uno a la vez.

Al presionar el botón “Reset” se reestablecerá todo como al principio de la simulación.

Al presionar el botón “Salir” se regresara a la ventana de carga de archivos.

El algoritmo de reemplazo de pagina usado es el de NRU, usando los bits de referencia y de modificación como banderas a la hora de buscar candidatos a reemplazar de forma aleatoria entre los candidatos, buscando primero los candidatos con ambos bits en 0, en caso de no encontrar ninguno se realiza una búsqueda de candidatos con bit de referencia en 0 pero con bit de modificación en 1, en caso de no encontrar ninguno, buscara aquellos con bit de referencia en 1 y bit de modificación en 0 y por último, en caso de no encontrar ningún candidato, se realizara un reemplazo de forma aleatoria entre las paginas que contienen un marco asignado.



El algoritmo se muestra a continuación:

//NRU

if(e==1){

cand.removeAll();

candidato=-1;

for(int p=0;p<this.memory.length;p++){

//r=0 m=0

if (this.memory[p][1]>=0&&this.memory[p][2]==0 && this.memory[p][3]==0){

cand.add(p+"");

}

}

if (cand.getItemCount()==0){

cand.removeAll();

for(int s=0;s<this.memory.length;s++){

//r=0 m=1

if (this.memory[s][1]>=0&&this.memory[s][2]==0 && this.memory[s][3]==1){

cand.add(s+"");

}

}

if(cand.getItemCount()==0){

cand.removeAll();

for(int t=0;t<this.memory.length;t++){

//r=1 m=0

if (this.memory[t][1]>=0&&this.memory[t][2]==1 && this.memory[t][3]==0){

cand.add(t+"");

}

}

if(cand.getItemCount()==0){

//cualquiera r=1 m=1

while(true){

candidato=(int)(Math.random() \* this.d.getPvirtual());

if (this.memory[candidato][1]>=0){e=0;break;}

}

//System.out.println("r1 m1");

}else{e=0;candidato=Integer.parseInt(cand.getItem((int)(Math.random() \* cand.getItemCount())));}

}else{e=0;candidato=Integer.parseInt(cand.getItem((int)(Math.random() \* cand.getItemCount())));}

}else{e=0;candidato=Integer.parseInt(cand.getItem((int)(Math.random() \* cand.getItemCount())));}

//System.out.println(candidato);

//System.out.println(e);

if (e==0){

//hay candidatos

this.memory[this.indice][1]=this.memory[candidato][1];

this.memory[this.indice][2]=0;

this.memory[this.indice][3]=0;

this.memory[this.indice][4]=0;

this.memory[this.indice][5]=0;

this.memory[candidato][1]=-1;

for(int f=2;f<6;f++){

this.memory[candidato][f]=0;

}

JLabel l= (JLabel) getComponentByName(candidato+"");

JButton b= (JButton) getComponentByName("btn"+candidato);

l.setForeground(new java.awt.Color(255, 0, 0));

l.setText("x");

l= (JLabel) getComponentByName(this.indice+"");

b= (JButton) getComponentByName("btn"+this.indice);

l.setForeground(new java.awt.Color(0, 0, 0));

l.setText(""+memory[this.indice][1]);

cand.removeAll();

candidato=-1;

**Salida**

La salida del programa se guarda en un archivo llamado “tracefile.txt” al finalizar la ejecución automática o al presionar el botón “Paso a paso” una vez ha finalizado la ejecución del paso a paso, se guarda en la misma ruta que el archivo de configuración de memoria.

El archivo de salida contiene información como la siguiente:

Cuando la operación tuvo éxito:

write 110000000000001 ...ok, Virtual: 1 Fisica: 1

Cuando existio un fallo de pagina:

write 888172 ...Fallo de pagina, Virtual: 54 Fisica asignada: 4