

Universidad Nacional Autónoma de México

Facultad de Ingeniería

Estructura y Programación de Computadoras

Semestre 2020 - 1

**Práctica 1: Ensamblador del MC68HC11**

Profesor:

Pedro Ignacio Rincón Gómez

Alumnos:

Murrieta Villegas, Alfonso

Reza Chavarria Sergio Gabriel

Mendieta Joaquín

Grupo: 4

Contacto: alfonsomvmx@comunidad.unam.mx

Índice

[Introducción 1](#_Toc22490941)

[Diseño y Estructura del Ensamblador 2](#_Toc22490942)

[Resultados y uso del ensamblador 30](#_Toc22490943)

[Conclusiones 35](#_Toc22490944)

[Referencias 35](#_Toc22490945)

# Introducción

En el presente proyecto se realizó un ensamblador - compilador para el micro de motorola MC68HC11 a través del lenguaje de alto nivel Java, esto con el fin de entender el funcionamiento y el proceso de compilación de un lenguaje ensamblador.

Como apartados importantes para el desarrollo de este proyecto se tomaron a consideración:

1. ***El lenguaje de programación***

El lenguaje que se utilizó para este proyecto fue Java sobre todo por la facilidad que el Paradigma Orientado a Objetos nos brindaría al momento de trabajar con diversos aspectos como la creación de clases para la segmentación del proyecto, el uso de paquetes para poder hacer modular nuestro proyecto y código, además de la gran cantidad de documentación existente para este lenguaje de programación.

Cabe destacar también que la interfaz gráfica de usuario (GUI) en el caso de Java es relativamente fácil de crear.

1. ***La implementación de la API POI***

Más adelante se especificará algunos aspectos respecto a por qué decidimos emplear esta forma de trabajar los mnemónicos del lenguaje ensamblador, pero por el momento el uso de la API de java creada por apache y denominada POI fue sobre todo para la lectura y captura de los datos del archivo Excel dado por nuestro profesor, el cual nos ofrecía todos los mnemónicos del lenguaje además de en qué tipo direccionamiento estaban disponibles y su respectiva cantidad de memoria RAM utilizada en el micro.

1. ***Uso de NetBeans como IDE***

Para poder desarrollar el proyecto de forma más rápida y con una de las mejores formas de depuración al momento de codificar se decidió utilizar un IDE, en el caso de Java existen IDEs como NetBeans o eclipse, sin embargo, se tomó la decisión de usar NetBeans debido a que es una herramienta ya conocida por todos los miembros del equipo.

***NOTAS:***

Cabe destacar que para este proyecto se utilizó la versión 12 de Java y la versión 4.1.1 de POI.

# Diseño y Estructura del Ensamblador

En este apartado se describirá cada una de las clases y paquetes empleados en la creación de este ensamblador:

# Generalización

Todo el proyecto está repartido de maneja general en 3 paquetes que se muestran en la imagen inferior:

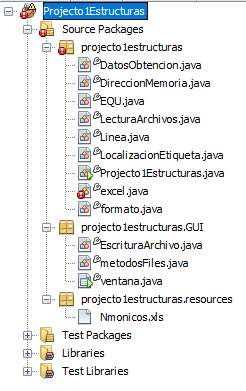


Imagen 1: En el cuadro rojo se encuentra el paquete principal denominado como “projecto1estructuras” el cual contiene toda la lógica del ensamblador, en el cuadro azul se encuentra el paquete que contiene las clases encargadas de la escritura de los archivos .LST y .HEX además de la que contiene la GUI, por último, en el cuadro verde se ve el último paquete el cual solamente contiene un archivo .XLS (Excel) el cual contiene todos los mnemónicos asociados al lenguaje ensamblador del micro.

A continuación, se mencionarán cada una de las clases encontradas en cada uno de los paquetes:

# Paquete de lógica del ensamblador

A continuación, se muestran capturas de pantalla de cada una de las clases utilizadas en este paquete:

**CLASE DATOS OBTENCIÓN**

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package projecto1estructuras;

import java.io.IOException;

import java.util.ArrayList;

/\*\*

\*

\* @author Sergio

\*/

public class DatosObtencion {

//Clase con el método para obtener que tipo de dato se encuentra en el código

public static void ObtencionDatos(ArrayList<Linea> datos, ArrayList<String> etiquetas,ArrayList <String> directiva, boolean correcto, boolean END,ArrayList <String> error) throws IOException{

for(int i=0; i<datos.size();i++){

if(etiquetas.contains(datos.get(i).etiqueta)){

System.out.print("\tEtiqueta "+datos.get(i).etiqueta);

error.add("");

}else if (!etiquetas.contains(datos.get(i).etiqueta)&&(datos.get(i).mnemonico.toUpperCase().equals("EQU")||datos.get(i).mnemonico.toUpperCase().equals("ORG")||datos.get(i).etiqueta.toUpperCase().equals("END")||datos.get(i).etiqueta.toUpperCase().equals("FCB"))){

System.out.print("\tDirectiva "+datos.get(i).mnemonico);

if(datos.get(i).mnemonico.toUpperCase().equals("END")){

END=true;

}

error.add("");

}else if(!etiquetas.contains(datos.get(i).etiqueta)&&excel.nmonico(datos.get(i).mnemonico.toUpperCase(),directiva)){

/\*

System.out.println("\t\tNmonico: "+datos.get(i).mnemonico);

System.out.println("\t\tDirectivas Soportadas: "+directiva);

\*/

//Directivas

//Casos especiales (BCLR, BRCLR, BSET, )

if(datos.get(i).mnemonico.toUpperCase().equals("BCLR")||datos.get(i).mnemonico.toUpperCase().equals("BSET")||datos.get(i).mnemonico.toUpperCase().equals("BRCLR")||datos.get(i).mnemonico.toUpperCase().equals("BRSET")){

System.out.println("Dato "+datos.get(i).dato);

if(datos.get(i).dato.toUpperCase().contains(",X,#")&&directiva.contains("INDX")){

System.out.print("\t"+datos.get(i).mnemonico+"\t INDEXADO EN X");

datos.get(i).directiva = "INDX";

error.add("");

}else if (datos.get(i).dato.toUpperCase().contains(",Y,#")&&directiva.contains("INDY")){

System.out.print("\t"+datos.get(i).mnemonico+"\t INDEXADO EN Y");

datos.get(i).directiva = "INDY";

error.add("");

}else if(directiva.contains("DIR")&&!datos.get(i).dato.toUpperCase().contains("X")&&!datos.get(i).dato.toUpperCase().contains("Y")){

System.out.print("\t"+datos.get(i).mnemonico+"\t DIRECTO");

datos.get(i).directiva = "DIR";

error.add("");

}

}else{

if(datos.get(i).dato.contains("$")&&!datos.get(i).dato.contains("#")){

if(datos.get(i).dato.toUpperCase().contains("X") || datos.get(i).dato.toUpperCase().contains("Y"))

{

if((datos.get(i).dato.contains("X")||datos.get(i).dato.contains("x"))&&directiva.contains("INDX")){

System.out.print("\t"+datos.get(i).mnemonico+"\t INDEXADO EN X");

datos.get(i).directiva = "INDX";

error.add("");

}else if ((datos.get(i).dato.contains("Y")||datos.get(i).dato.contains("y"))&&directiva.contains("INDY")){

System.out.print("\t"+datos.get(i).mnemonico+"\t INDEXADO EN Y");

datos.get(i).directiva = "INDY";

error.add("");

}

}else{

if(((datos.get(i).dato.length()<5))&&directiva.contains("DIR")&&!datos.get(i).dato.contains("#")){

System.out.print("\t"+datos.get(i).mnemonico+"\t DIRECTO");

datos.get(i).directiva ="DIR";

error.add("");

}else if((datos.get(i).dato.length()==5)&&directiva.contains("EXT")&&!datos.get(i).dato.contains("#")){

System.out.print("\t"+datos.get(i).mnemonico+"\t EXTENDIDO");

datos.get(i).directiva= "EXT";

error.add("");

}else if((datos.get(i).dato.length()==3)&&directiva.contains("EXT")&&!datos.get(i).dato.contains("#")){

System.out.print("\t"+datos.get(i).mnemonico+"\t EXTENDIDO");

datos.get(i).directiva= "EXT";

datos.get(i).dato=datos.get(i).dato.replace("$", "$00");

error.add("");

}

}

}else if(datos.get(i).dato.contains("$")&&datos.get(i).dato.contains("#")&&directiva.contains("IMM")){

System.out.print("\t"+datos.get(i).mnemonico+"\t INMEDIATO");

datos.get(i).directiva = "IMM";

error.add("");

}else if(!datos.get(i).dato.contains("$")&&directiva.contains("INH")&&!etiquetas.contains(datos.get(i).dato)){

System.out.print("\t"+datos.get(i).mnemonico+"\t INHERENTE");

datos.get(i).directiva = "INH";

error.add("");

}else if(!datos.get(i).dato.contains("$")&&directiva.contains("REL")&&etiquetas.contains(datos.get(i).dato)){

System.out.print("\t"+datos.get(i).mnemonico+"\t RELATIVO");

datos.get(i).directiva = "REL";

error.add("");

}else if(datos.get(i).mnemonico.toUpperCase().equals("JSR")||datos.get(i).mnemonico.toUpperCase().equals("JMP")){

System.out.print("\t"+datos.get(i).mnemonico+"\t EXTENDIDO");

datos.get(i).directiva = "EXT";

error.add("");

}else{

//Errores posibles

//Etiqueta inexiste

if(directiva.contains("REL")&&!etiquetas.contains(datos.get(i).dato)){

System.out.print("003\t"+datos.get(i).mnemonico+"\t Relativo no contiene etiqueta válida");

error.add("\t"+datos.get(i).mnemonico+"\t Relativo no contiene etiqueta válida");

//Instruccion carece de operandos

}else if(!directiva.contains("INH")&&(datos.get(i).dato.contains("$")||datos.get(i).dato.contains("#"))){

System.out.print("005\t"+datos.get(i).mnemonico+"\t Necesita de dato");

error.add("\t"+datos.get(i).mnemonico+"\t Necesita de dato");

//Constante inexistente

}else if(!directiva.contains("INH")&&!etiquetas.contains(datos.get(i).dato)){

System.out.print("001\t"+datos.get(i).mnemonico+"\t constante inexistente");

error.add("001\t"+datos.get(i).mnemonico+"\t constante inexistente");

//Instruccion no lleva operadores

}else if(datos.get(i).dato.contains("$")&&directiva.contains("INH")){

System.out.print("006\t"+datos.get(i).mnemonico+"\t Inherente contiene valores");

error.add("\t"+datos.get(i).mnemonico+"\t Inherente contiene valores");

//Mnemonico inexistente

}else{

System.out.print("004\t"+datos.get(i).mnemonico+"\t no existe");

error.add("\t"+datos.get(i).mnemonico+"\t no existe");

}

correcto=false;

}

}

directiva.clear();

}else{

System.out.print("Datos: "+datos.get(i).dato);

}

System.out.println("");

}

}

public static void PrimeraPasada(ArrayList<Linea> datos,ArrayList<DireccionMemoria> DirMem,ArrayList<LocalizacionEtiqueta> LocEti, ArrayList<String> etiquetas, boolean END, ArrayList<Integer> ListaOrg,ArrayList <String> error){

int cont, org=0 , firstOrg=0;

boolean PrimerOrg=false;

for(int m=0; m<datos.size();m++){

//Inicio para el guardado

if(etiquetas.contains(datos.get(m).etiqueta)){

LocEti.add(new LocalizacionEtiqueta(org,datos.get(m).etiqueta));

}

//Caso de la directiva ORG

if(datos.get(m).mnemonico.toUpperCase().equals("ORG")){

datos.get(m).dato=datos.get(m).dato.replace("$","0x");

org=(Integer.decode(datos.get(m).dato)).intValue();

if(PrimerOrg==false){

firstOrg=(Integer.decode(datos.get(m).dato)).intValue();

PrimerOrg=true;

}

ListaOrg.add(org);

//System.out.println("Org String: "+datos.get(m).dato);

//System.out.println("Org Dato:"+org);

//Caso de la Directiva END

}else if(datos.get(m).mnemonico.toUpperCase().equals("END")){

END=true;

datos.get(m).dato=datos.get(m).dato.replace("$","0x");

if(!datos.get(m).dato.equals("")){

org=(Integer.decode(datos.get(m).dato)).intValue();

if(org==firstOrg&&END){

System.out.println("Con END");

}else{

System.out.println("Sin END");

error.get(m).concat("SIN END");

}

}else{

System.out.println("Con END");

}

//Caso directiva FCB

}else if(datos.get(m).mnemonico.toUpperCase().equals("FCB")){

String dSep=datos.get(m).dato.replace("$", "");

String dSepArr[]=dSep.split(",");

DirMem.add(new DireccionMemoria(org,dSepArr[0],"DATO",0,datos.get(m).mnemonico));

System.out.println(Integer.toHexString(org)+" "+dSepArr[0]);

org++;

DirMem.add(new DireccionMemoria(org,dSepArr[1],"DATO",0,datos.get(m).mnemonico));

System.out.println(Integer.toHexString(org)+" "+dSepArr[1]);

org++;

//Caso directiva ORG

}else if(!datos.get(m).mnemonico.toUpperCase().equals("ORG")&&!etiquetas.contains(datos.get(m).dato.toUpperCase())){

cont=0;

String []Opcode;

Opcode=datos.get(m).opcode.split(" ");

String dSep=datos.get(m).dato.replace("$", "");

dSep=dSep.replace("", "");

dSep=dSep.replace("#", "");

dSep=dSep.replace(",X","");

dSep=dSep.replace(",Y","");

dSep=dSep.replace(",x","");

dSep=dSep.replace(",y","");

dSep=dSep.replace(",","");

char [] dSepArry=dSep.toCharArray();

//System.out.println("DSEP "+dSep);

//Diferentes tipos de escritura de los códigos operacionales y de los datos

while(cont<datos.get(m).TamBite){

if (cont==0 && Opcode.length==1){

DirMem.add(new DireccionMemoria(org,""+Opcode[0],"OPCODE",datos.get(m).TamBite,datos.get(m).mnemonico));

System.out.println(Integer.toHexString(org)+" "+Opcode[0]);

cont++;

org++;

}else if((cont==0 && Opcode.length>=2)){

DirMem.add(new DireccionMemoria(org,Opcode[0],"OPCODE",datos.get(m).TamBite,datos.get(m).mnemonico));

System.out.println(Integer.toHexString(org)+" "+Opcode[0]);

org++;

cont++;

DirMem.add(new DireccionMemoria(org,Opcode[1],"OPCODE",datos.get(m).TamBite-1,datos.get(m).mnemonico));

System.out.println(Integer.toHexString(org)+" "+Opcode[1]);

org++;

cont++;

}else if(cont!=0 && dSepArry.length==1){

String junto1="0"+dSepArry[0];

DirMem.add(new DireccionMemoria(org,junto1,"DATO",0));

System.out.println(Integer.toHexString(org)+" "+junto1);

org++;

cont++;

}else if(cont!=0 && dSepArry.length==2&& !datos.get(m).directiva.equals("EXT")){

String junto1=""+dSepArry[0]+dSepArry[1];

DirMem.add(new DireccionMemoria(org,junto1,"DATO",0));

System.out.println(Integer.toHexString(org)+" "+junto1);

org++;

cont++;

}else if(cont!=0 && dSepArry.length==2&& datos.get(m).directiva.equals("EXT")){

DirMem.add(new DireccionMemoria(org,"00","DATO",0));

String junto1=""+dSepArry[0]+dSepArry[1];

org++;

cont++;

DirMem.add(new DireccionMemoria(org,junto1,"DATO",0));

System.out.println(Integer.toHexString(org)+" "+junto1);

org++;

cont++;

}else if(cont!=0 && dSepArry.length==3){

DirMem.add(new DireccionMemoria(org,"0"+dSepArry[0],"DATO",0));

String junto1=""+dSepArry[1]+dSepArry[2];

org++;

cont++;

DirMem.add(new DireccionMemoria(org,junto1,"DATO",0));

System.out.println(Integer.toHexString(org)+" "+junto1);

org++;

cont++;

}else if(cont!=0 && dSepArry.length==4){

String junto1=""+dSepArry[0]+dSepArry[1];

DirMem.add(new DireccionMemoria(org,junto1,"DATO",0));

System.out.println(Integer.toHexString(org)+" "+junto1);

org++;

cont++;

String junto2=""+dSepArry[2]+dSepArry[3];

System.out.println(Integer.toHexString(org)+" "+junto2);

DirMem.add(new DireccionMemoria(org,junto2,"DATO",0));

org++;

cont++;

}else if(cont!=0 && dSepArry.length>4){

String junto1=""+dSepArry[0]+dSepArry[1];

DirMem.add(new DireccionMemoria(org,junto1,"DATO",0));

System.out.println(Integer.toHexString(org)+" "+junto1);

org++;

cont++;

String junto2=""+dSepArry[2]+dSepArry[3];

System.out.println(Integer.toHexString(org)+" "+junto2);

DirMem.add(new DireccionMemoria(org,junto2,"DATO",0));

org++;

cont++;

String junto3="";

for(int i=4;i<dSepArry.length;i++){

junto3=junto3+dSepArry[i];

}

DirMem.add(new DireccionMemoria(org,junto3,"DATO",0));

System.out.println(Integer.toHexString(org)+" "+junto3);

org++;

cont++;

}

}

//caso que los datos tengan etiquetas

}else if(!datos.get(m).mnemonico.toUpperCase().equals("ORG")&&etiquetas.contains(datos.get(m).etiqueta.toUpperCase())){

org++;

//caso de las etiquetas con las directivas relativas

}else if(!datos.get(m).mnemonico.toUpperCase().equals("ORG")&&etiquetas.contains(datos.get(m).dato.toUpperCase())&&datos.get(m).directiva.toUpperCase().equals("REL")){

DirMem.add(new DireccionMemoria(org,datos.get(m).opcode,"OPCODE",datos.get(m).TamBite,datos.get(m).mnemonico));

System.out.println(Integer.toHexString(org)+" "+datos.get(m).opcode);

org++;

DirMem.add(new DireccionMemoria(org,datos.get(m).dato,"DATO",0));

System.out.println(Integer.toHexString(org)+" "+datos.get(m).dato);

org++;

//caso de las etiquetas que son JSR Y JMP

}else if(!datos.get(m).mnemonico.toUpperCase().equals("ORG")&&etiquetas.contains(datos.get(m).dato.toUpperCase())){

DirMem.add(new DireccionMemoria(org,datos.get(m).opcode,"OPCODE",datos.get(m).TamBite,datos.get(m).mnemonico));

System.out.println(Integer.toHexString(org)+" "+datos.get(m).opcode);

org++;

DirMem.add(new DireccionMemoria(org,datos.get(m).dato,"DATO",0));

System.out.println(Integer.toHexString(org)+" "+datos.get(m).dato);

org++;

DirMem.add(new DireccionMemoria(org,datos.get(m).dato,"DATO",0));

System.out.println(Integer.toHexString(org)+" "+datos.get(m).dato);

org++;

//caso de las instrucciones inherentes

}else if(!datos.get(m).mnemonico.toUpperCase().equals("ORG")&&datos.get(m).dato.equals("")){

DirMem.add(new DireccionMemoria(org,datos.get(m).opcode,"OPCODE",datos.get(m).TamBite,datos.get(m).mnemonico));

System.out.println(Integer.toHexString(org)+" "+datos.get(m).dato);

org++;

}

}

}

public static void SegundaPasada(ArrayList<DireccionMemoria> DirMem,ArrayList<LocalizacionEtiqueta> LocEti, ArrayList<String>error){

for(int i=0; i<DirMem.size();i++){

for(int k=0; k<LocEti.size();k++){

if(LocEti.get(k).etiqueta.equals(DirMem.get(i).dato)){

if((DirMem.get(i+1).espacio>LocEti.get(k).espacio)&&((DirMem.get(i+1).espacio-LocEti.get(k).espacio)<127)){

//System.out.println(DirMem.get(i).dato+"\t"+DirMem.get(i+1).espacio);

//System.out.println(LocEti.get(k).etiqueta+"\t"+LocEti.get(k).espacio);

int res=DirMem.get(i).espacio-LocEti.get(k).espacio;

res=255-res;

String dato=Integer.toHexString(res);

//System.out.println("1Valor "+res);

if(dato.length()==1){

DirMem.get(i).dato="0"+dato;

}else{

DirMem.get(i).dato=dato;

}

}else if((DirMem.get(i+1).espacio<LocEti.get(k).espacio)&&(DirMem.get(i+1).espacio-LocEti.get(k).espacio)<128){

//System.out.println(DirMem.get(i).dato+"\t"+DirMem.get(i+1).espacio);

//System.out.println(LocEti.get(k).etiqueta+"\t"+LocEti.get(k).espacio);

int res=LocEti.get(k).espacio-DirMem.get(i+1).espacio+DirMem.get(i+1).sizeOp;

String dato=""+res;

if(dato.length()==1){

DirMem.get(i).dato="0"+dato;

}else{

DirMem.get(i).dato=dato;

}

//System.out.println("2Valor "+res);

}else{

//Salto relativo lejano

error.add("Salto en "+DirMem.get(i+1).espacio+" fuera de rango");

}

}

}

}

}

}

**CLASE DIRECCIÓN MEMORIA**

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package projecto1estructuras;

/\*\*

\*

\* @author Sergio

\*/

public class DireccionMemoria{

public int espacio,sizeOp=0;

public String dato, tipoDato;

public String nmemo="";

public DireccionMemoria(int espacio, String dato, String tipoDato, int sizeOp, String nmemo){

this.espacio=espacio;

this.dato=dato;

this.tipoDato=tipoDato;

this.sizeOp=sizeOp;

this.nmemo=nmemo;

}

public DireccionMemoria(int espacio, String dato, String tipoDato, int sizeOp){

this.espacio=espacio;

this.dato=dato;

this.tipoDato=tipoDato;

this.sizeOp=sizeOp;

}

}

**CLASE EQU**

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package projecto1estructuras;

/\*\*

\*

\* @author Sergio

\*/

public class EQU {

//Método para obtener todas las etiquetas que utilizan la directiva EQU y asigna el vslor de memoria obtenida

public String etiqueta,valor;

public EQU(String etiqueta, String valor){

this.etiqueta=etiqueta;

this.valor=valor;

}

}

**CLASE LECTURA ARCHIVOS**

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package projecto1estructuras;

import java.io.\*;

import java.util.StringTokenizer;

import java.util.ArrayList;

//Clase para la lectura del archivo de entrada .asc

public class LecturaArchivos {

//

/\*

public void Direccionarchivo(File archivo){

this.archivo=archivo;

}\*/

public static void obtencionDatosArchivo(File archivo,ArrayList<String> etiqueta, ArrayList<Linea> linea3, ArrayList<String>lineasArch) {

//File archivo = null;

FileReader fr = null;

BufferedReader br = null;

String dato1="",dato2="",dato3="";

int contador=0;

boolean espacio=false;

try {

// Apertura del fichero y creacion de BufferedReader para poder

// hacer una lectura comoda (disponer del metodo readLine()).

//archivo = new File (archivo2.getAbsolutePath());

fr = new FileReader (archivo);

br = new BufferedReader(fr);

// Lectura del fichero

String linea;

String [] linea2;

//Deteccion de Etiquetas y de NMONICOS

while((linea=br.readLine())!=null){

linea=linea.replaceAll("\\s"," ");

lineasArch.add(linea);

linea2=linea.split(" ");

dato1="";

dato2="";

dato3="";

contador=0;

for(int x=0; x<linea2.length; x++){

//System.out.println(x+linea2[x]);

if(linea2[x].contains("\*"))

{

break;

}else if(!linea2[x].equals("") && x==0){

//System.out.println("Etiqueta["+linea2[x]+"]");

dato1=linea2[x];

etiqueta.add(linea2[x]);

System.out.println(dato1);

contador++;

espacio=false;

}else if(!linea2[x].equals("") && x!=0){

//System.out.println("Obtenido["+linea2[x]+"]");

if(!dato1.equals("")&&dato2.equals("")){

dato2=linea2[x];

System.out.println(dato2);

contador++;

espacio=false;

}else if(dato1.equals("")&&dato2.equals("")){

dato2=linea2[x];

System.out.println(dato2);

contador++;

espacio=true;

}

else if(!dato2.equals("")&&dato3.equals("")){

dato3=linea2[x];

//System.out.println(dato3);

contador++;

}else if(!dato3.equals("")){

dato3=dato3+" "+linea2[x];

//System.out.println(dato3);

contador++;

}

}

}

//System.out.println("contador"+contador);

if(contador==1 && espacio==false){

linea3.add(new Linea(dato1,espacio));

}else if(contador==1 && espacio==true){

linea3.add(new Linea(dato2));

}else if(contador==2){

linea3.add(new Linea(dato2,dato3,espacio));

}else if(contador==3){

linea3.add(new Linea(dato1,dato2,dato3,espacio));

}

}

}

catch(Exception e){

e.printStackTrace();

}finally{

// En el finally cerramos el fichero, para asegurarnos

// que se cierra tanto si todo va bien como si salta

// una excepcion.

try{

if( null != fr ){

fr.close();

}

}catch (Exception e2){

e2.printStackTrace();

}

}

}

}

**CLASE LINEA**

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package projecto1estructuras;

/\*\*

\*

\* @author Sergio

\*/

public class Linea {

public String etiqueta="", mnemonico="", dato="", directiva="", opcode="";

public int TamBite=0;

boolean espacioInicial=false;

//Constructor para la lectura de una linea con 3 datos

public Linea (String etiqu, String mnemo, String dato, boolean esp){

this.etiqueta=etiqu;

this.mnemonico=mnemo;

this.dato=dato;

this.espacioInicial=esp;

}

//Constructor para la lectura de una linea con 2 datos

public Linea(String mnemo, String dato, boolean esp){

this.mnemonico=mnemo;

this.dato=dato;

this.espacioInicial=esp;

}

//Constructor con 1 dato

public Linea(String etiq, boolean esp ){

this.etiqueta=etiq;

this.espacioInicial=esp;

}

public Linea(String mnemo){

this.mnemonico=mnemo;

this.espacioInicial=true;

}

}

**CLASE LOCALIZACIÓN ETIQUETA**

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package projecto1estructuras;

/\*\*

\*

\* @author Sergio

\*/

public class LocalizacionEtiqueta {

public int espacio;

public String etiqueta;

public LocalizacionEtiqueta(int espacio, String etiqueta){

this.espacio=espacio;

this.etiqueta=etiqueta;

}

}

**CLASE PROYECTO1ESTRUCTURAS**

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package projecto1estructuras;

import java.io.IOException;

import java.util.ArrayList;

import java.util.Iterator;

import java.util.LinkedList;

import projecto1estructuras.GUI.ventana;

/\*\*

\*

\* @author Sergio

\*/

public class Projecto1Estructuras {

/\*\*

\* @param args the command line arguments

\* @throws java.io.IOException

\*/

public static void main(String[] args) throws IOException {

//Instancias

ventana ventana1= new ventana();

ventana1.setVisible(true);

ventana1.setLocationRelativeTo(null);

}

}

**CLASE EXCEL**

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package projecto1estructuras;

import java.io.File;

import java.io.FileInputStream;

import java.io.IOException;

import java.util.ArrayList;

import java.util.Iterator;

//Importanciones de la Api APACHE POI

import org.apache.poi.hssf.usermodel.HSSFDateUtil;

import org.apache.poi.hssf.usermodel.HSSFSheet;

import org.apache.poi.hssf.usermodel.HSSFWorkbook;

import org.apache.poi.ss.usermodel.Cell;

import org.apache.poi.ss.usermodel.Row;

public class excel {

public static void arhivo() throws IOException{

//Archivo de Nmónicos

FileInputStream file = new FileInputStream(new File("../0 Projecto1Estructuras\\src\\projecto1estructuras\\resources\\Nmonicos.xls"));

HSSFWorkbook workbook = new HSSFWorkbook(file);

HSSFSheet sheet = workbook.getSheetAt(0);

Iterator<Row> rowIterator = sheet.iterator();

Row row;

//Ciclo de la fila

while (rowIterator.hasNext()){

row = rowIterator.next();

Iterator<Cell> cellIterator = row.cellIterator();

Cell celda;

//Ciclo de la celda

while (cellIterator.hasNext()){

celda = cellIterator.next();

switch(celda.getCellType()) {

case NUMERIC:

System.out.print(celda.getNumericCellValue());

break;

case STRING:

System.out.print(celda.getStringCellValue());

break;

case BOOLEAN:

System.out.print(celda.getBooleanCellValue());

break;

}

System.out.print("\t");

}

System.out.println("");

}

// cerramos el libro excel

workbook.close();

}

//Función que detecta si el nmonico existe y el tipo de nmonico

public static boolean nmonico(String nmonico, ArrayList <String> directiva)throws IOException{

FileInputStream file = new FileInputStream(new File("../0 Projecto1Estructuras\\src\\projecto1estructuras\\resources\\Nmonicos.xls"));

HSSFWorkbook workbook = new HSSFWorkbook(file);

HSSFSheet sheet = workbook.getSheetAt(0);

Iterator<Row> ri = sheet.iterator();

Row row;

int contador=0;

while (ri.hasNext()){

row = ri.next();

Iterator<Cell> cellIterator = row.cellIterator();

Cell celda;

//Ciclo de la celda

while (cellIterator.hasNext()){

celda = cellIterator.next();

switch(celda.getCellType()) {

case STRING:

//Obtiene si existe el mnemonico en la lista de excel

if(celda.getStringCellValue().toUpperCase().equals(nmonico))

{

//Empieza con -1 por la lectura de la primera celda, celda del mnemonico

contador=-1;

while(cellIterator.hasNext()&& contador<21){

celda = cellIterator.next();

//Switch que lee la columna del mnemonico

switch(celda.getCellType()){

//Caso del OPCODE HEXADECEMAL

case STRING:

//System.out.println(contador+"\t"+celda.getStringCellValue());

if(!celda.getStringCellValue().contains("--")){

switch(contador){

case 0:

directiva.add("IMM");

break;

case 3:

directiva.add("DIR");

break;

case 6:

directiva.add("INDX");

break;

case 9:

directiva.add("INDY");

break;

case 12:

directiva.add("EXT");

break;

case 15:

directiva.add("INH");

break;

case 18:

directiva.add("REL");

break;

default:

break;

}

}

break;

case NUMERIC:

//Caso de OPCODE en digito

//System.out.println(contador+"\t"+celda.getNumericCellValue());

switch(contador){

case 0:

directiva.add("IMM");

break;

case 3:

directiva.add("DIR");

break;

case 6:

directiva.add("INDX");

break;

case 9:

directiva.add("INDY");

break;

case 12:

directiva.add("EXT");

break;

case 15:

directiva.add("INH");

break;

case 18:

directiva.add("REL");

break;

default:

break;

}

break;

default:

break;

}

contador++;

}

workbook.close();

return true;

}

break;

default:

break;

}

}

}

workbook.close();

return false;

}

public static void ObtenerOpCode(Linea dato)throws IOException{

FileInputStream file = new FileInputStream(new File("C:\\Users\\Sergio\\Desktop\\0 Projecto1Estructuras\\src\\projecto1estructuras\\resources\\Nmonicos.xls"));

HSSFWorkbook workbook = new HSSFWorkbook(file);

HSSFSheet sheet = workbook.getSheetAt(0);

Iterator<Row> ri = sheet.iterator();

Row row;

int caso=0;

int contador;

while (ri.hasNext()){

row = ri.next();

Iterator<Cell> cellIterator = row.cellIterator();

Cell celda;

//Ciclo de la celda

while (cellIterator.hasNext()){

celda = cellIterator.next();

switch(celda.getCellType()) {

case STRING:

//Obtiene si existe el mnemonico en la lista de excel

if(celda.getStringCellValue().toUpperCase().equals(dato.mnemonico))

{

contador=-1;

if(dato.directiva.equals("IMM")){

caso=0;

}else if(dato.directiva.equals("DIR")){

caso=3;

}else if(dato.directiva.equals("INDX")){

caso=6;

}else if(dato.directiva.equals("INDY")){

caso=9;

}else if(dato.directiva.equals("EXT")){

caso=12;

}else if(dato.directiva.equals("INH")){

caso=15;

}else if(dato.directiva.equals("REL")){

caso=18;

}

//System.out.println("\t"+caso+"\t"+celda.getStringCellValue());

while(cellIterator.hasNext()&& contador<=(caso+2)){

celda = cellIterator.next();

//Switch que lee la columna del mnemonico

if(contador>=caso){

switch(celda.getCellType()){

//Caso del OPCODE HEXADECEMAL

case STRING:

//System.out.println(contador+"\t"+celda.getStringCellValue());

if(contador==caso){

dato.opcode=celda.getStringCellValue();

}

break;

case NUMERIC:

//System.out.println(contador+"\t"+celda.getNumericCellValue());

if(contador==caso){

dato.opcode=""+celda.getNumericCellValue();

}

else if(contador==caso+2){

dato.TamBite=(int)celda.getNumericCellValue();

}

break;

default:

break;

}

}

contador++;

}

workbook.close();

}

break;

default:

break;

}

}

}

workbook.close();

}

}

**CLASE FORMATO**

package projecto1estructuras;

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

/\*\*

\*

\* @author Sergio

\*/

public class formato {

String valorA;

String espacio;

String valores;

String cadena;

public formato(String valA,String space, String val, String cad){

this.valorA=valA;

this.espacio=space;

this.valores=val;

this.cadena=cad;

}

}

# Paquete de la GUI y escritura de archivos

A continuación, se muestran capturas de pantalla de cada una de las clases utilizadas en este paquete:

**CLASE METODOS FILES**

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package projecto1estructuras.GUI;

import java.io.BufferedReader;

import java.io.File;

import java.io.FileNotFoundException;

import java.io.FileReader;

import java.io.FileWriter;

import java.io.PrintWriter;

import java.util.ArrayList;

import projecto1estructuras.DireccionMemoria;

import projecto1estructuras.EQU;

import projecto1estructuras.Linea;

import projecto1estructuras.formato;

/\*\*

\*

\* @author Sergio

\*/

public class EscrituraArchivo {

//Impresión del foramto del archivo hexadecimal

public static void escritoHEXA(ArrayList<DireccionMemoria> DirMem,ArrayList<Integer> Org){

FileWriter fichero = null;

PrintWriter pw = null;

int contadorOrg=0;

int contadorLista=0;

int cont=0;

try

{

fichero = new FileWriter("C:\\Users\\Sergio\\Desktop\\0 Projecto1Estructuras\\prueba.hex");

pw = new PrintWriter(fichero);

for(int i=0; i<DirMem.size();i++){

if(DirMem.get(i).espacio==Org.get(contadorOrg)&&contadorOrg==0){

pw.print("<"+Integer.toHexString(DirMem.get(i).espacio)+"> ");

if(Org.size()!=1){

contadorOrg++;

}

contadorLista=DirMem.get(i).espacio;

}else if(DirMem.get(i).espacio==Org.get(contadorOrg)&&contadorOrg!=0&&contadorOrg!=Org.size()){

pw.print("\n\n<"+Integer.toHexString(DirMem.get(i).espacio)+"> ");

if(contadorOrg!=Org.size()-1){

contadorOrg++;

cont=0;

}

contadorLista=DirMem.get(i).espacio;

}

if(cont!=15){

pw.print(DirMem.get(i).dato+" ");

cont++;

}else

{

pw.println(DirMem.get(i).dato+" ");

cont=0;

contadorLista+=16;

pw.print("\n<"+Integer.toHexString(contadorLista)+"> ");

}

}

} catch (Exception e) {

e.printStackTrace();

} finally {

try {

// Nuevamente aprovechamos el finally para

// asegurarnos que se cierra el fichero.

if (null != fichero)

fichero.close();

} catch (Exception e2) {

e2.printStackTrace();

}

}

}

public static void escritoLST(File archivo, ArrayList<DireccionMemoria> DirMem,ArrayList<String> etiquetas,ArrayList<EQU> equ, ArrayList<String> lineasArch, ArrayList<Integer> org,ArrayList <String> error) throws FileNotFoundException{

//Lectura

FileWriter fichero = null;

PrintWriter pw = null;

int contador=1;

int conEqu=0;

int conDir=0;

int conOrg=0;

int conEti=0;

boolean espacio=false;

try {

fichero = new FileWriter("C:\\Users\\Sergio\\Desktop\\0 Projecto1Estructuras\\lista.lst");

pw = new PrintWriter(fichero);

String linea;

pw.println("M68HC11 Absolute Assembler Version 1.0.0 "+archivo.getAbsolutePath());

for(int j=0;j<lineasArch.size();j++){

pw.print(j+"\tA\t");

if(lineasArch.get(j).startsWith("\*")){

pw.print("\t\t\t\t\t");

pw.println(lineasArch.get(j));

}else if(lineasArch.get(j).toUpperCase().contains("EQU")){

pw.print("0000\t\t\t\t\t");

pw.print(String.format("%-10s",equ.get(conEqu).valor));

pw.println(lineasArch.get(j));

conEqu++;

}else if(lineasArch.get(j).toUpperCase().contains("ORG")){

pw.print(String.format("%-10s",Integer.toHexString(org.get(conOrg))));

pw.println(lineasArch.get(j));

conOrg++;

}else if(lineasArch.get(j).toUpperCase().contains("END")){

pw.print("\t\t\t\t\t");

pw.println(lineasArch.get(j));

}else if(!lineasArch.get(j).equals("")){

if(lineasArch.get(j).contains(DirMem.get(conDir).nmemo)){

pw.print(Integer.toHexString(DirMem.get(conDir).espacio)+" ");

String cadena="";

for(int i=0; i<DirMem.get(conDir).sizeOp;i++){

if(i!=DirMem.get(conDir).sizeOp-1){

cadena=cadena+DirMem.get(conDir+i).dato;

pw.print(DirMem.get(conDir+i).dato+" ");

}else if(i==DirMem.get(conDir).sizeOp-1){

cadena=cadena+DirMem.get(conDir+i).dato;

pw.print(DirMem.get(conDir+i).dato+"\t");

}

}

pw.println(lineasArch.get(j));

if(conDir+DirMem.get(conDir).sizeOp>=DirMem.size())

conDir=conDir+DirMem.get(conDir).sizeOp-1;

else

conDir=conDir+DirMem.get(conDir).sizeOp;

}else{

//pw.print("\t\t\t\t\t");

for (int i = 0; i < etiquetas.size(); i++) {

if(lineasArch.get(j).contains(etiquetas.get(i))){

pw.print(String.format("%-10s",Integer.toHexString(DirMem.get(conDir).espacio)));

pw.println(lineasArch.get(j));

}

}

}

}else{

pw.println(Integer.toHexString(DirMem.get(conDir).espacio)+"\t\t\t\t\t");

}

}

for (int i = 0; i < error.size(); i++) {

pw.println(error.get(i));

}

} catch (Exception e) {

e.printStackTrace();

} finally {

try {

// Nuevamente aprovechamos el finally para

// asegurarnos que se cierra el fichero.

if (null != fichero)

fichero.close();

} catch (Exception e2) {

e2.printStackTrace();

}

}

}

}

**CLASE ESCRITURA ARCHIVOS**

package projecto1estructuras.GUI;

import java.io.\*;

/\*\*

\*

\* @author murryFly

\*/

public class metodosFiles {

FileInputStream entrada;

FileOutputStream salida;

File archivo;

public String openFile(File arch){

String texto="";

try{

entrada = new FileInputStream(arch);

int tx;

while((tx = entrada.read())!= -1){

char car = (char)tx;

texto += car;

}

}

catch(Exception xx){ }

return texto;

}

public String createFile(File arch){

String isOkay = null;

try{

salida = new FileOutputStream(arch);

isOkay = "ARCHIVO GUARDADO";

}

catch(Exception xx){

}

return isOkay;

}

}

**CLASE VENTANA**

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package projecto1estructuras.GUI;

import java.awt.Color;

import java.awt.Insets;

import java.io.File;

import java.io.IOException;

import java.util.ArrayList;

import javax.swing.\*;

import javax.swing.border.EmptyBorder;

import javax.swing.text.AttributeSet;

import javax.swing.text.SimpleAttributeSet;

import javax.swing.text.StyleConstants;

import javax.swing.text.StyleContext;

import projecto1estructuras.DatosObtencion;

import projecto1estructuras.DireccionMemoria;

import projecto1estructuras.EQU;

import projecto1estructuras.LecturaArchivos;

import projecto1estructuras.Linea;

import projecto1estructuras.LocalizacionEtiqueta;

import projecto1estructuras.excel;

/\*\*

\*

\* @author murryFly

\*/

public class ventana extends javax.swing.JFrame {

JFileChooser seleccion = new JFileChooser(); //Para seleccionar

File archivo; //Archivo

metodosFiles case1 = new metodosFiles();

JPanel topPanel=new JPanel();

JTextPane tPane;

//LecturaArchivos LArch= new LecturaArchivos();

public ventana() {

initComponents();

}

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-BEGIN:initComponents

private void initComponents() {

jScrollPane1 = new javax.swing.JScrollPane();

btn\_editar = new javax.swing.JButton();

jLabel1 = new javax.swing.JLabel();

jScrollPane2 = new javax.swing.JScrollPane();

jTextPane1 = new javax.swing.JTextPane();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

setTitle("Murrieta\_Valdespino\_Reza");

btn\_editar.setText("Examinar");

btn\_editar.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

btn\_editarActionPerformed(evt);

}

});

jLabel1.setFont(new java.awt.Font("Consolas", 1, 18)); // NOI18N

jLabel1.setText("Ensamblador TeaMancos");

jScrollPane2.setViewportView(jTextPane1);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()

.addContainerGap(332, Short.MAX\_VALUE)

.addComponent(btn\_editar, javax.swing.GroupLayout.PREFERRED\_SIZE, 155, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(35, 35, 35))

.addGroup(layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGap(153, 153, 153)

.addComponent(jLabel1))

.addGroup(layout.createSequentialGroup()

.addGap(58, 58, 58)

.addComponent(jScrollPane2, javax.swing.GroupLayout.PREFERRED\_SIZE, 394, javax.swing.GroupLayout.PREFERRED\_SIZE)))

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()

.addGap(30, 30, 30)

.addComponent(jLabel1)

.addGap(27, 27, 27)

.addComponent(jScrollPane2, javax.swing.GroupLayout.PREFERRED\_SIZE, 132, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 33, Short.MAX\_VALUE)

.addComponent(btn\_editar)

.addContainerGap())

);

pack();

}// </editor-fold>//GEN-END:initComponents

private void btn\_editarActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_btn\_editarActionPerformed

if(seleccion.showDialog(null, "Cargar Archivo")==JFileChooser.APPROVE\_OPTION){

archivo = seleccion.getSelectedFile();

if(archivo.canRead()){

if(archivo.getName().toUpperCase().endsWith("ASC") ){

//Colocar código ya del análisis del archivo -> ENSAMBLADOR

//LArch.Direccionarchivo(archivo);

try{

OperandoArchivo();

}catch (IOException e){

System.out.println("IO");

}

}else{

JOptionPane.showMessageDialog(null, "Selecciona un archivo asc");

}

}

}

}//GEN-LAST:event\_btn\_editarActionPerformed

public static void main(String args[]) {

java.awt.EventQueue.invokeLater(new Runnable() {

public void run() {

new ventana().setVisible(true);

}

});

}

public void OperandoArchivo() throws IOException{

ArrayList<String> etiquetas= new ArrayList<String>();

ArrayList<String> equEtiqueta= new ArrayList<String>();

ArrayList<EQU> equ=new ArrayList<EQU>();

ArrayList<String> listaArch=new ArrayList<String>();

ArrayList<Linea> datos= new ArrayList<Linea>();

ArrayList <String> directiva=new ArrayList<String>();

ArrayList<DireccionMemoria> DirMem= new ArrayList<DireccionMemoria>();

ArrayList<LocalizacionEtiqueta> LocEti= new ArrayList<LocalizacionEtiqueta>();

ArrayList<Integer> ListaOrg=new ArrayList<Integer>();

ArrayList<String>error= new ArrayList<String>();

Boolean correcto=true;

Boolean END=false;

int org=0x0;

String aux;

//txt\_datos.setText("Obtención de datos del archivo");

LecturaArchivos.obtencionDatosArchivo(this.archivo,etiquetas, datos,listaArch);

//Obtención de las EQU

for(int i=0; i<datos.size();i++){

if(datos.get(i).mnemonico.equals("EQU")){

if(datos.get(i).dato.contains("$00")&&(!datos.get(i).dato.equals("$00")||!datos.get(i).dato.equals("#$00"))){

datos.get(i).dato=datos.get(i).dato.replace("$00", "$");

}

equ.add(new EQU(datos.get(i).etiqueta,datos.get(i).dato));

}

}

for(int i=0;i<equ.size();i++){

for(int j=0;j<datos.size();j++){

//Cambio de las etiquetas que usan EQU por el valor.

if(datos.get(j).dato.equals(equ.get(i).etiqueta)&&!datos.get(j).mnemonico.toUpperCase().equals("EQU")&&!datos.get(j).dato.contains("#")){

datos.get(j).dato = equ.get(i).valor;

}else if(datos.get(j).dato.equals("#"+equ.get(i).etiqueta)&&!datos.get(j).mnemonico.toUpperCase().equals("EQU")){

datos.get(j).dato ="#"+equ.get(i).valor;

}

}

}

//Obtención de los tipos de datos y de las directivas

DatosObtencion.ObtencionDatos(datos, etiquetas, directiva, correcto, END,error);

//Obtecnión de OPCODE Y tamaño

for(int m=0; m<datos.size();m++){

excel.ObtenerOpCode(datos.get(m));

}

System.out.println("Primera pasada");

DatosObtencion.PrimeraPasada(datos, DirMem,LocEti, etiquetas, END,ListaOrg,error);

if(END=false){

System.out.println("Sin Directiva END");

}

System.out.println("Segunda Pasada");

//Caso de las directivas JMP Y JSR

for(int i=0; i<DirMem.size();i++){

int k=0;

if((DirMem.get(i).dato.contains("BD")||DirMem.get(i).dato.contains("7E"))&&DirMem.get(i).tipoDato.contains("OPCODE")){

while(!LocEti.get(k).etiqueta.contains(DirMem.get(i+1).dato)){

k++;

}

String vad=Integer.toHexString(LocEti.get(k).espacio);

char []dato=vad.toCharArray();

DirMem.get(i+1).dato=""+dato[0]+dato[1];

DirMem.get(i+2).dato=""+dato[2]+dato[3];

//System.out.println("\t\t"+DirMem.get(i+1).dato);

//System.out.println("\t\t"+DirMem.get(i+2).dato);

}

}

for(int i=0;i<DirMem.size();i++){

DirMem.get(i).dato=DirMem.get(i).dato.replace(" ","");

DirMem.get(i).dato=DirMem.get(i).dato.replace(".0","");

if(DirMem.get(i).dato.length()==1){

DirMem.get(i).dato="0"+DirMem.get(i).dato;

}

}

//Caso de las directivas relativas junto

DatosObtencion.SegundaPasada(DirMem, LocEti,error);

EscrituraArchivo.escritoHEXA(DirMem, ListaOrg);

EscrituraArchivo.escritoLST(archivo, DirMem,etiquetas, equ,listaArch,ListaOrg,error);

for(int i=0; i<DirMem.size();i++){

System.out.print("\u001B[0m"+Integer.toHexString(DirMem.get(i).espacio)+" ");

if(DirMem.get(i).tipoDato.equals("OPCODE")){

System.out.println("\u001B[31m"+DirMem.get(i).dato);

}else{

System.out.println("\u001B[34m"+DirMem.get(i).dato);

}

}

}

// Variables declaration - do not modify//GEN-BEGIN:variables

private javax.swing.JButton btn\_editar;

private javax.swing.JLabel jLabel1;

private javax.swing.JScrollPane jScrollPane1;

private javax.swing.JScrollPane jScrollPane2;

private javax.swing.JTextPane jTextPane1;

// End of variables declaration//GEN-END:variables

}

# Paquete del archivo Excel

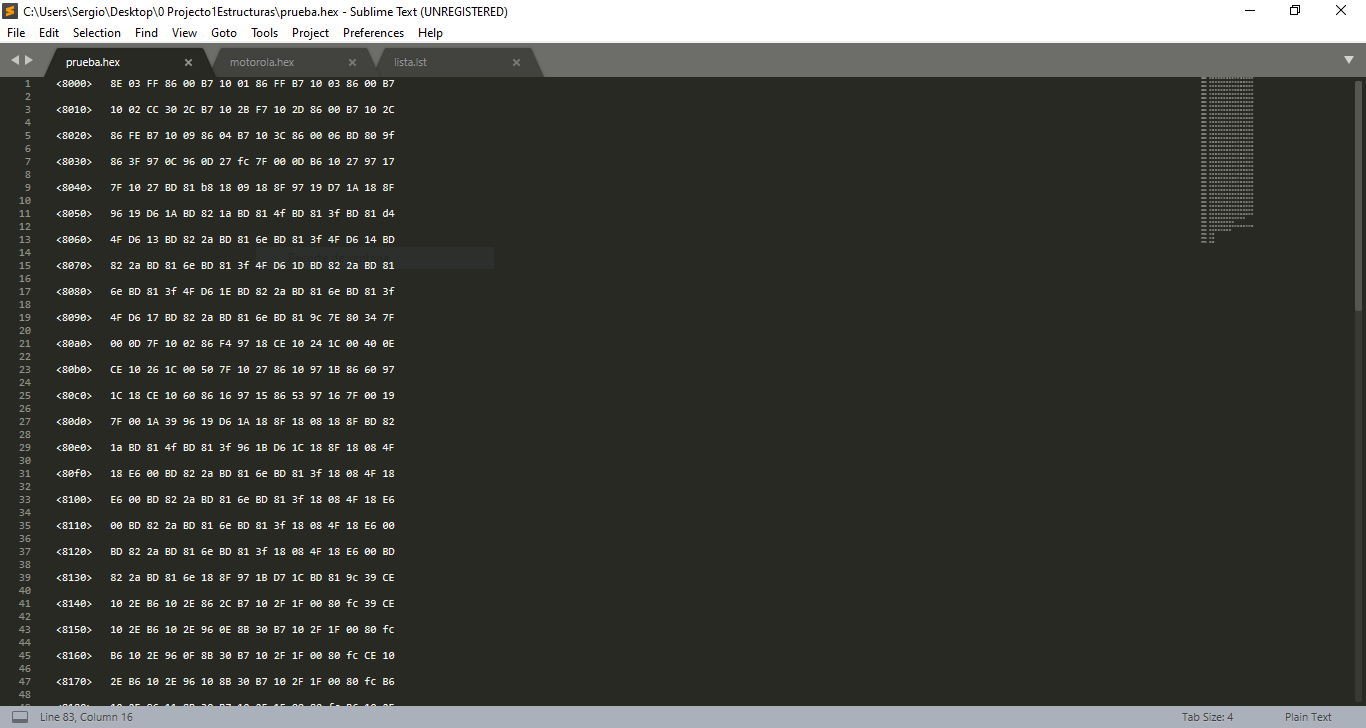
Para este apartado realmente no hay ninguna clase o algo que contenga código, realmente sólo se utilizó un archivo que previamente el profesor nos dio el cual contiene todos los mnemonicos del lenguaje ensamblador además de los modos de direccionamiento en los que trabajan y su respectiva cantidad e memoria utilizada.

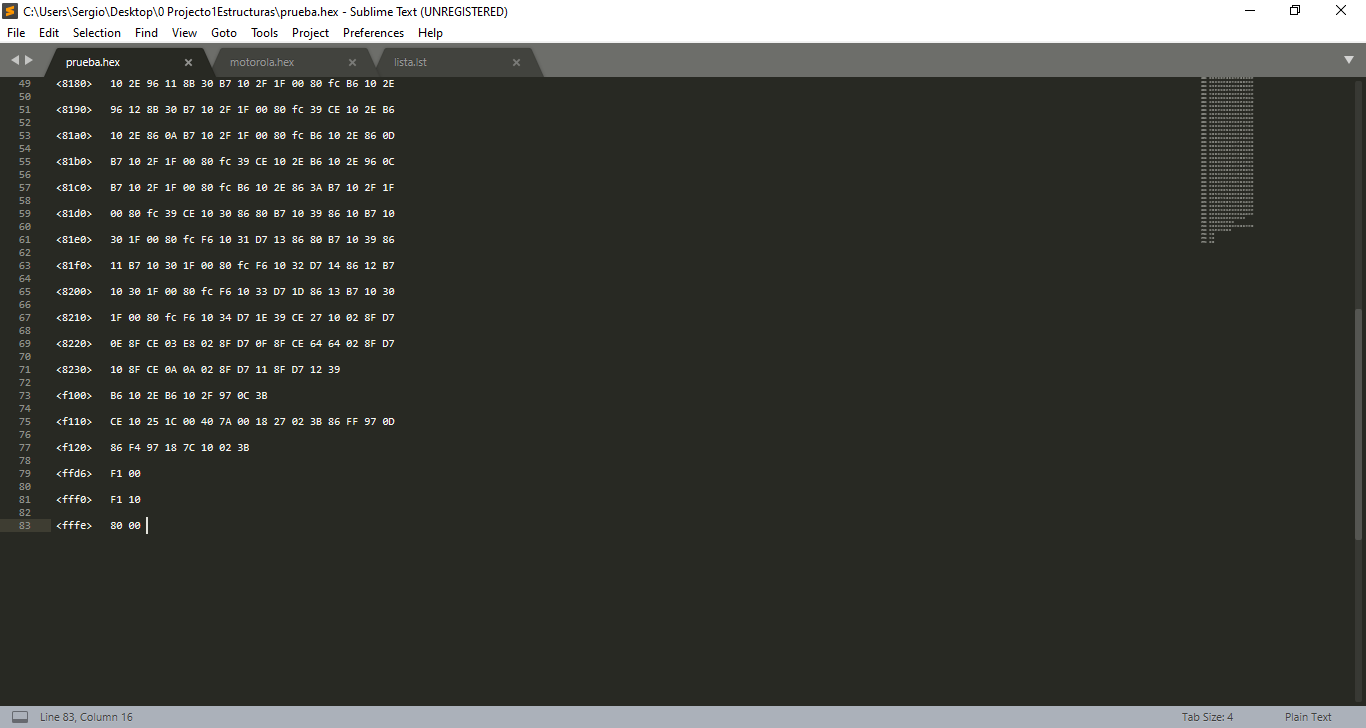
Cabe destacar que en este archivo se corrigieron algunos aspectos vistos en clase como la corrección de los modos de direccionamiento de BRCLR entre otros.

# Resultados y uso del ensamblador

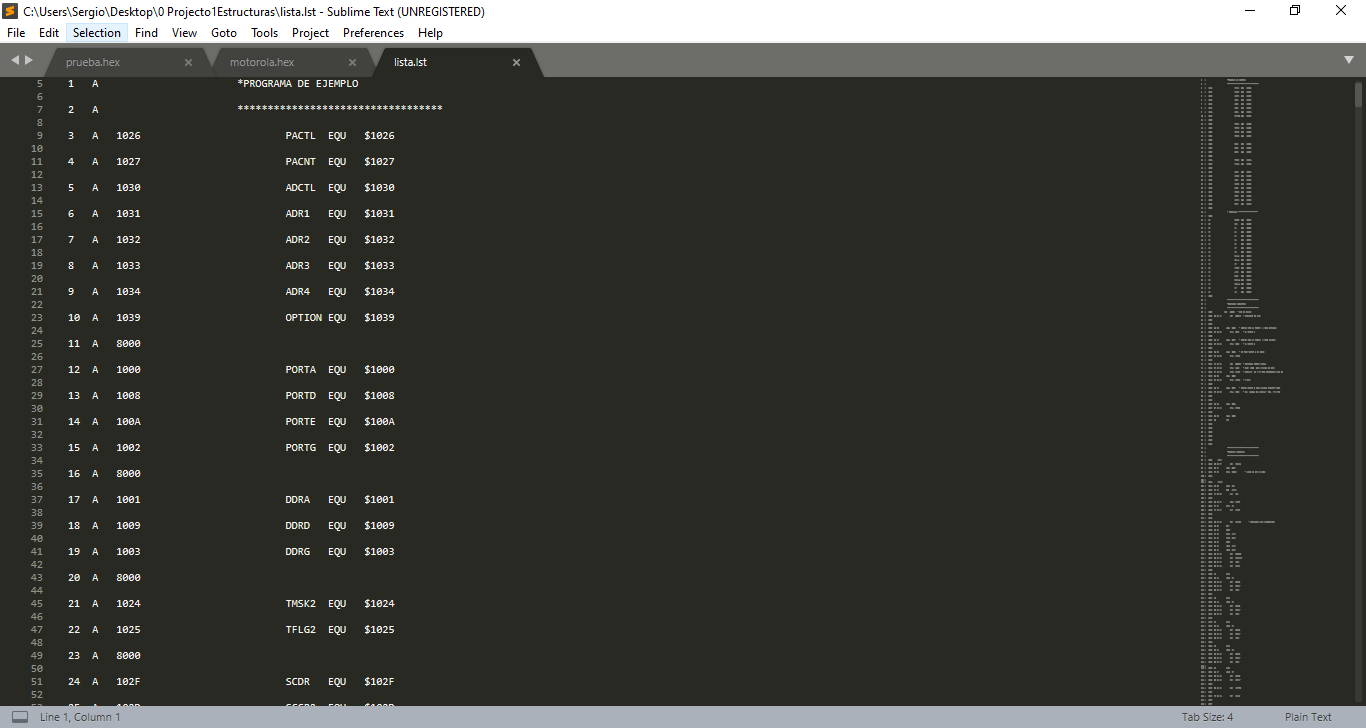
A continuación, se muestran los resultados obtenidos con el ensamblador al leer como archivo de entrada a exemplo:

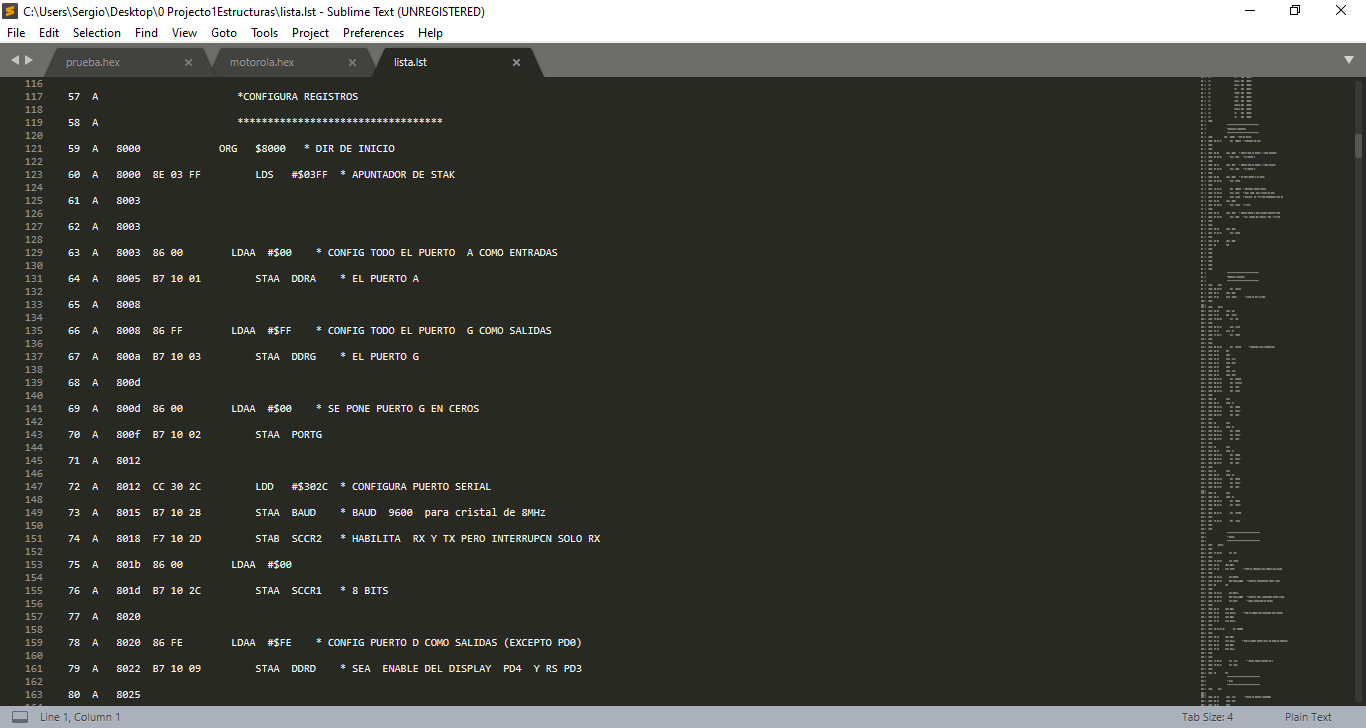
1. **Apartado del archivo HEX**

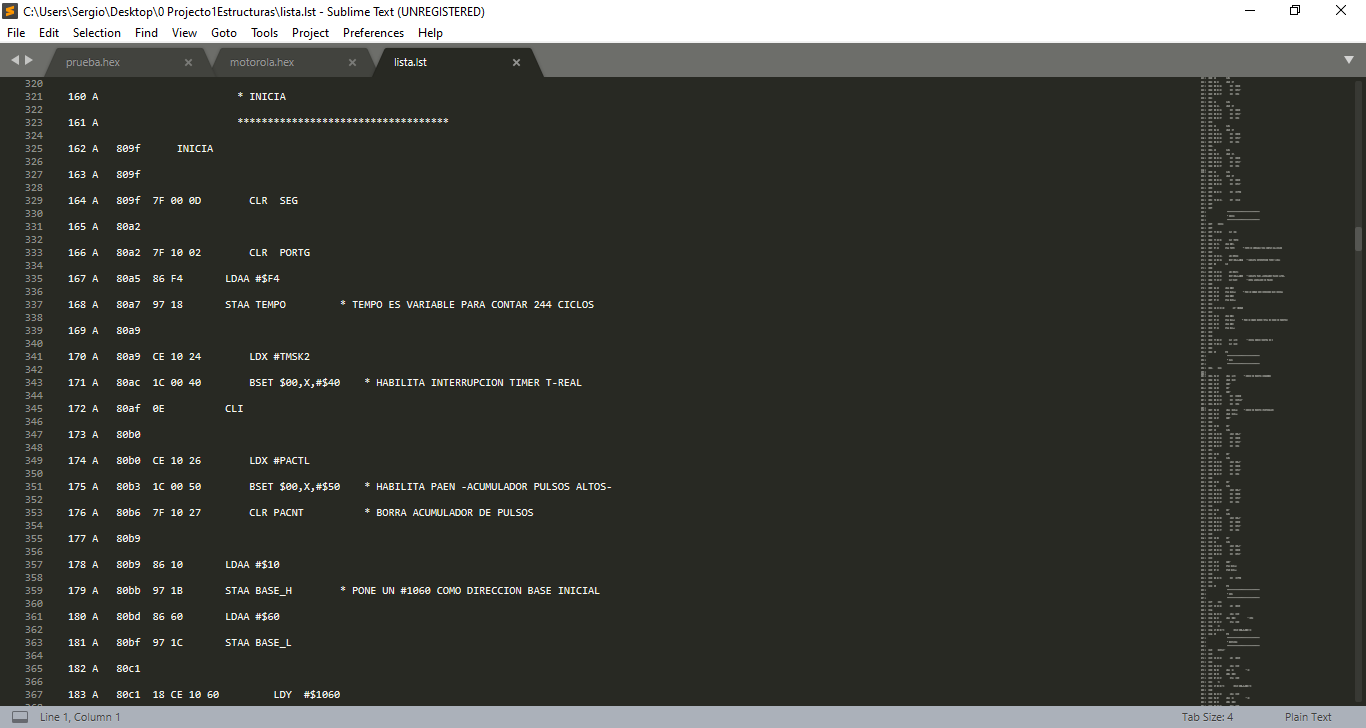


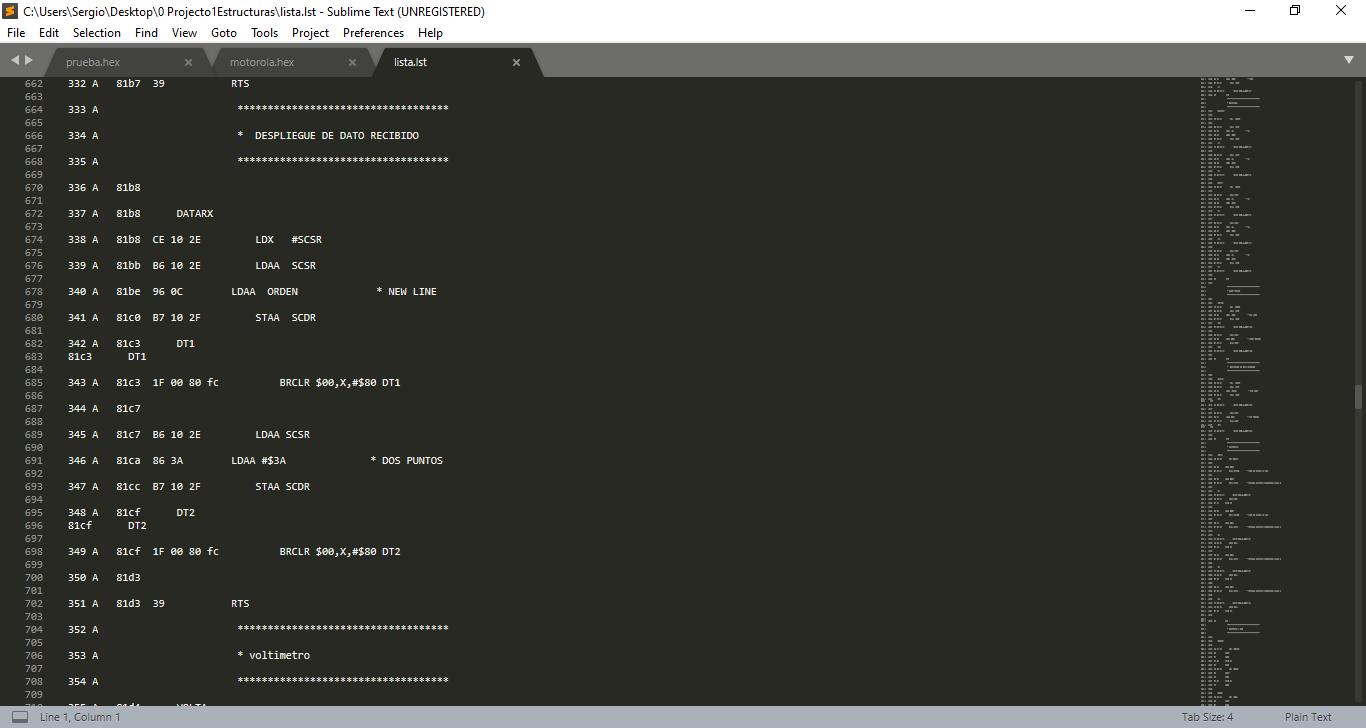


1. **Apartado del archivo LST**

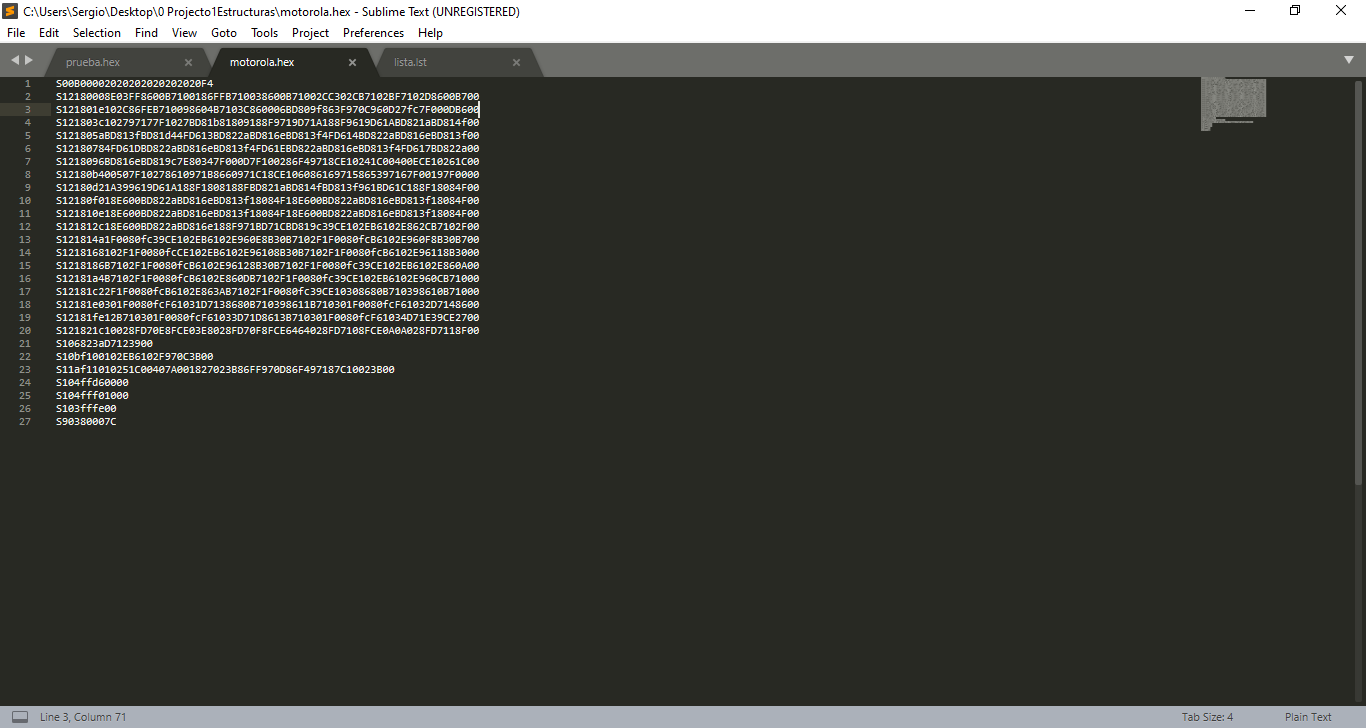






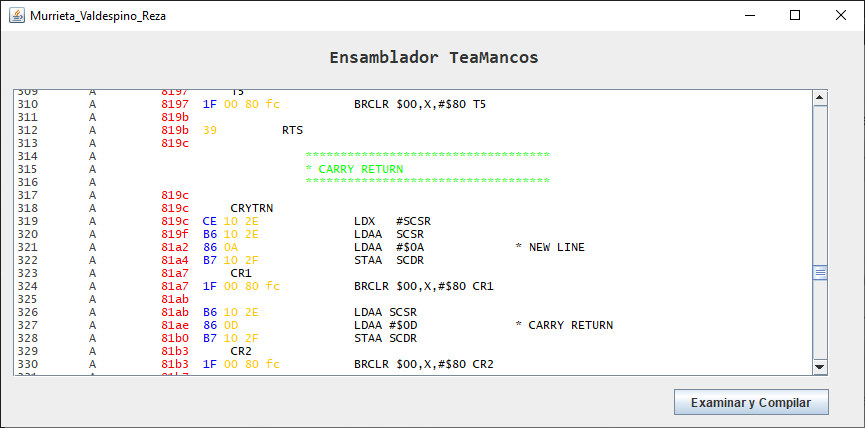


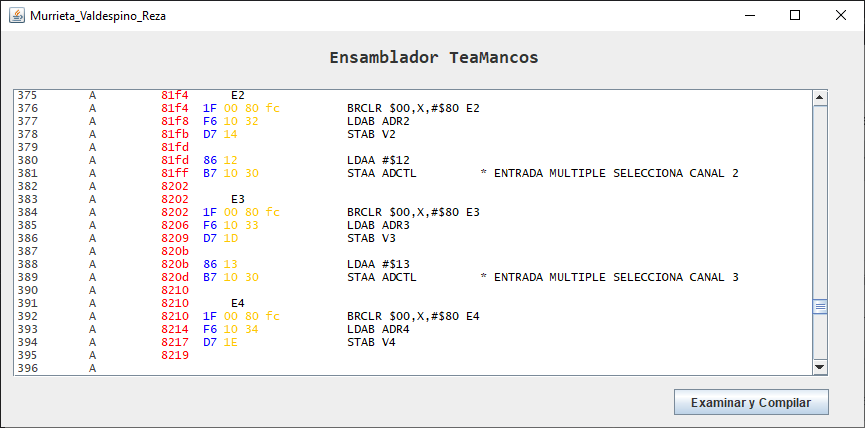
1. **Apartado del archivo HEX con el formato de motorola**



**NOTA**: Cabe destacar que al no saber bien como es que se obtenía el último byte en cada una de las líneas del archivo HEX con el formato de motorola fue como se decidió el colocar 00

1. **Apartado gráfico y de interpretación del código (GUI)**



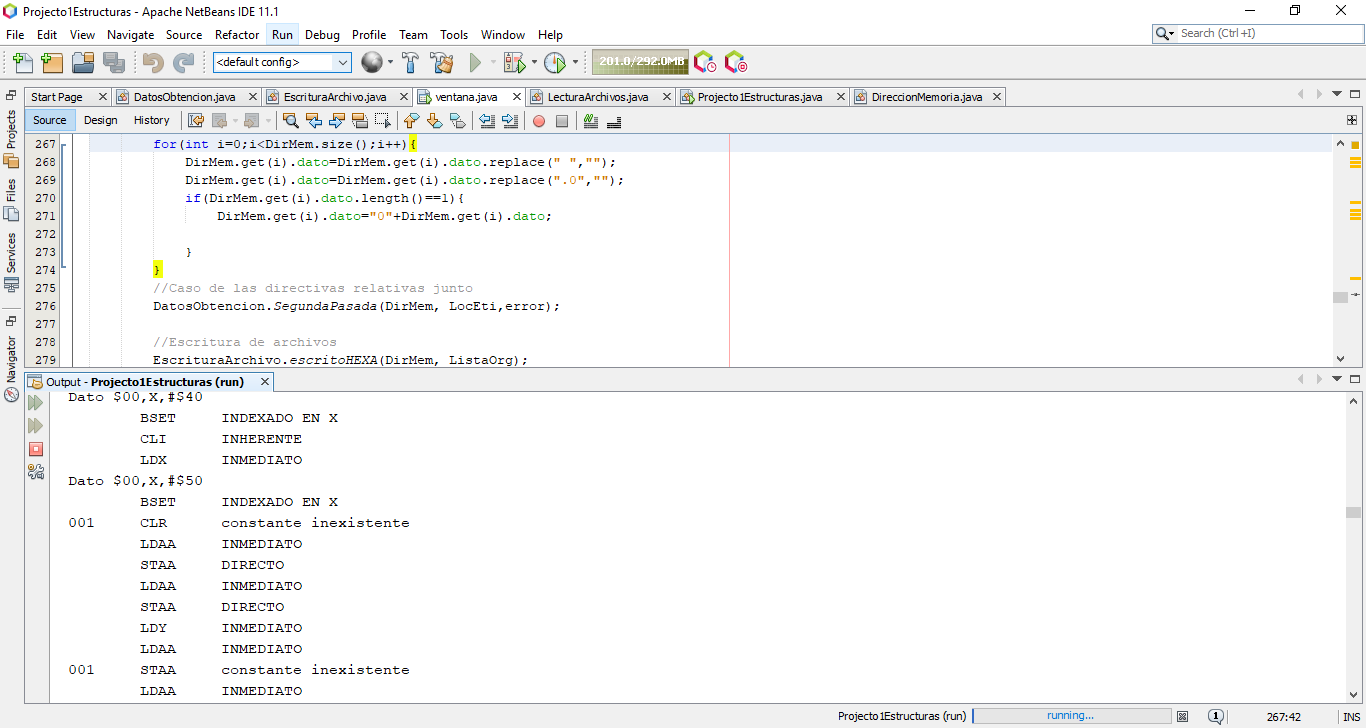


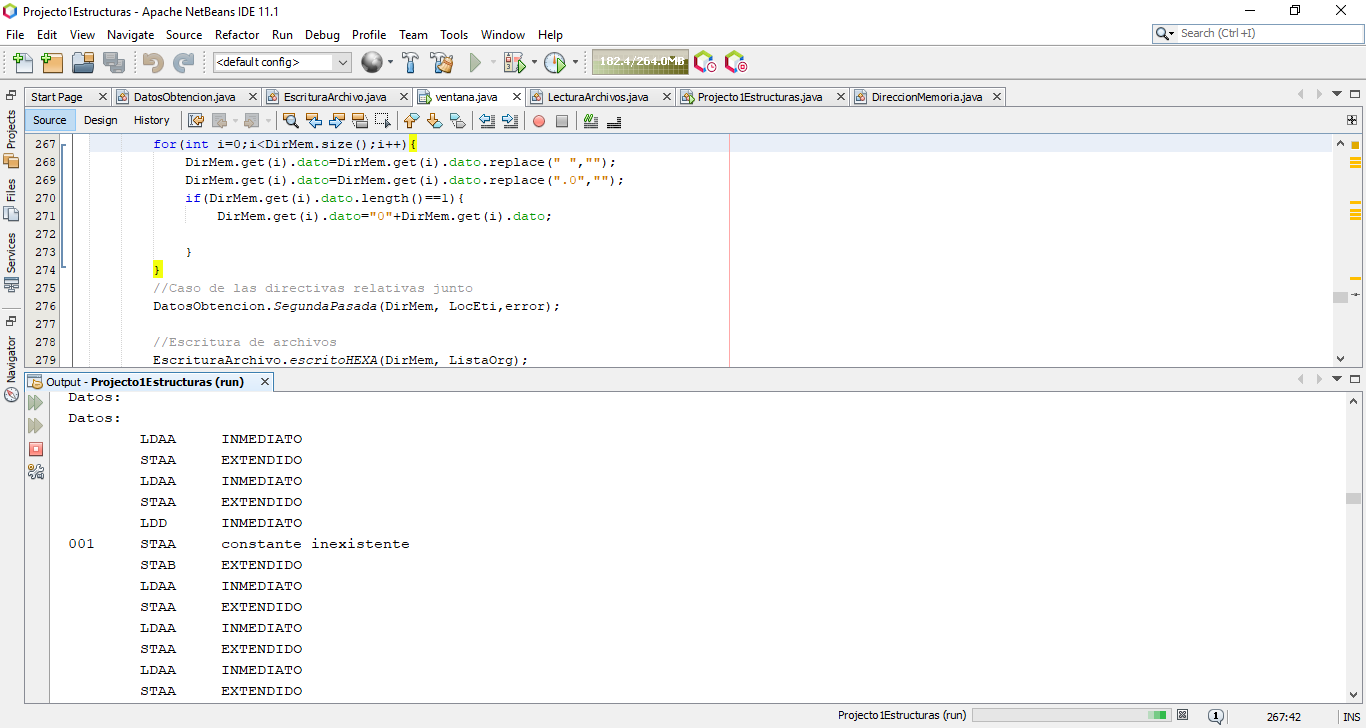
NOTA: Como complemento a nuestro compilador decidimos dar un apartado a color para que el usuario de esa forma pudiera visualizar cada uno de los elementos que conforma al código a ensamblar, de color negro se colocó el código fuente leído, de color verde los comentarios, de color rojo la posición en la memoria, de color azul el opcode y el color amarillo son los operandos.

1. **Manejo de errores**

Cabe destacar que, para evitar problema con la generación de archivos, fue como se decidió que solamente en consola y en el archivo LST se pudiera ver el manejo de errores, esto con el fin de que el archivo HEX no se generara con algún error.

A continuación, se muestran en consola los errores obtenidos al leer el archivo error.s19 dado por el profesor:





# Conclusiones

Realmente la creación de este ensamblador nos resulta un gran logro como trabajo de equipo y como proyecto, realmente el poder desarrollador un ensamblador “básico” es fundamental para poder comprender conceptos claves dentro de los lenguajes ensambladores como los modos de direccionamiento, el manejo de memoria e incluso el como es que cada nemónico tiene ciertas limitantes al momento de usarlos.

Por otro lado, y con base a los conocimientos adquiridos en nuestra materia de Compiladores, realmente sabemos que este ensamblador tiene varios aspectos que podrían ser mejorables como el manejo de errores e incluso el poder desarrollar más a fondo apartados como el análisis semántico o incluso la optimización del código máquina.

# Referencias

1. Recuperado el 10 de octubre de 2019, de <https://poi.apache.org/>
2. M68HC11 Reference Manual. Motorola.
3. Recuperado el 10 de octubre de 2019, de http://www.iearobotics.com/proyectos/libro6811/libro-6811.pdf