**Universidad de Guadalajara**

**Centro Universitario de Ciencias Exactas e Ingenierías**

**Taller de programación de sistemas**

**Reporte # 4**

**ALUMNO:**

Oscar Alberto Palomares López Sección D15

Quiñones Nunñez Karla Patricia Sección D07

**Descripción de las variables y funciones utilizadas**

Para esta Segunda práctica seguimos empleando las 2 clases pasadas y ahora segentamos el codigo en otras mas siendo Tabop Tabsim Temporal y Validador, aunque solamente mandamos los metodos de validación a la clase Validador y los metodos del Tabop a la clase del Tabop etc aquí listaremos los métodos nuevos que creamos y para lo que sirven.

**Variables nuevas de la clase EnsambladorHC12Raw:**

private String contenidoArchivoTemporalTxt;

-Contiene el contenido del archivontemporal en formato string

private String contenidoTabsim;

-contiene el contenido del archivo tabsim en formato string

private final String TEMPORAL\_FILE\_NAME = "/P4TMP.txt";

-es el nombre que tendra por defecto el archivo temporal

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

**Métodos nuevos:**

**public sataric srring agregarCeros(): Se encarga de obtener una string o un decimal y agregarle ceros a la izquierda de este**

**public static String procesarTabsim(String contenidoArchivoTemporalTxt) { Se encarga de procesar el contenido del archivo temporal y crear el contenido del tabsim usando solamente un ciclo que itera sobre las lineas del contenido del archivo temporal y hacer validaciónes básicas como if else instanciando los metodos de validación de la clase validación como validarEtiqueta para agregar las etiquetas al archivo Tabsim etc**

* **public static String procesarTemporal(String contenidoProcesado**e Se encarga de procesar el contenido de la variable contenidoArchivoProcesadoTxt ya anteriormente habladado y analizarlo para crear el contenido que ira dentro del archivo temporal al igual que el de Tabsim se utiliza un ciclo, se iteran las lineas de la variable contenidoArchivoProcesadoTxt y se mandan a llamar validaciones mediante if else y los metodos de la clase Validador.
* **Los siguientes metodos son los que validan cada una de los tipos de directivas constantes que se hablaron en la practica**
* **public static String validaTipoDirectivaConstante(String codop, String operando, int LINE\_NUMBER) {**
* public static String validaCONSTANTEdeReserva2byte(String operando, int LINE\_NUMBER){
* public static String validaCONSTANTEdeReserva1byte(String operando, int LINE\_NUMBER){
* public static String validaCONSTANTEdeCaracteres(String operando, int LINE\_NUMBER){
* public static String validaCONSTANTEde2Byte(String operando, int LINE\_NUMBER){
* public static String validaCONSTANTEde1Byte(String operando, int LINE\_NUMBER){

En estos métodos se hacen validaciones básicas usando por ejemplo las funciones Regex de Java, para ORG END EQU , CONT\_LOC y DIR\_INIC sus validaciones están localizadas en el método procesarTemporal y se validan usando if else como if codop.equals(“END”) etc.

**Clase public class EnsambladorHC12UI extends javax.swing.JFrame :**

En esta clase se agregó 2 nuevas ventanas donde se muestra el contenido del Archivo Temporal y en su caso de haber contenido el del Tabsim

**Nuevas Variables:**

|  |
| --- |
| private javax.swing.JLabel jLabel4; |
|  |
|  |
|  |
| private javax.swing.JLabel jLabel5; |
|  |
|  |
| ; |
| private javax.swing.JTextArea jTextArea5; |
|  |