Criterio C: Desarrollo

Técnicas Utilizadas

Para el desarrollo de este programa se utilizaron dos IDE (Integrated Development Environment), en un principio Dr Java y posteriormente se terminó usando Net Beans. Ya que a mi parecer el segundo demuestra una mayor facilidad para el ordenamiento dentro del programa, ya que permita la separación por capas MVC (Modelo Vista Controlador). Además que señala los errores y advertencias inmediatamente, sin necesidad de presionar un botón de compilación, y muestra maneras de solucionar cada uno de estos.

1-. Clases

Dentro del proyecto se realizaron nueve clases, las cuales son mostradas en la figura 1. Dentro de estas clases dos extienden de JFrame ya que hacen referencia a las pantallas de la interfaz gráfica, estas son “MainFrame1” y “UserFrame1”, mostradas en las figuras 2 y 3. Dentro de la creación de clases, se hizo uso de herencia, en las clases “Cliente” y “Empleado” mostradas en las figuras 8 y 9, las cuales extienden de la clase “Persona” (Figura 6),de este forma se reutiliza código (el de la clase Persona) y facilita la complejidad del código. También se llegó a establecer una relación entre las clases de acuerdo a los atributos que tenían, esta se da en la clase “Evento” (Figura 7), la cual tiene como atributo un objeto de la clase “Cliente” (Figura 8), de esta manera se puede hacer referencia a que cada objeto de tipo evento tiene un objeto de tipo cliente que extiende de la clase persona, logrando así una mejor conexión en la información. Por ultimo existen dos clases que son independientes a las demás la clase “n2t” (Figura 5) y la clase “Usuario” (Figura 4), la primera tiene métodos que convierten un número en las palabras del mismo, la segunda clase tiene como fin almacenar los usuarios de acceso de manera ordenada junto con sus atributos. Es importante mencionar que en cuenta a las clases de las figuras 4, 6 7 8 y 9, contienen un área de “gets” y “sets” de sus atributos respectivos, tal y como se muestra en la Figura 10.

2-.Archivos

En vista de que al cerrar el programa se necesita que guarde información el programa para seguir utilizándola posteriormente cuando se reanude el programa. Se optó por la implementación de archivos binarios, exactamente 6 archivos binarios (Figura 11). Se decidieron usar este tipo de archivos ya que cada uno de estos guarda información del área que su nombre indica, de manera segura y ligera. Para su implementación se usó la clase “RandomAccessFile” para leer y escribir en los archivos.

1. Código leer archivo
   1. No se realizaron métodos para leer, ya que solo se hace una vez, al empezar el programa. Al leer el archivo se basa en obtener primero la cantidad de registros que se obtienen y posteriormente se pasan un arreglo del tipo de la clase que se esté hablando, creando objetos de este tipo, un ejemplo se encuentra en la figura 12. Este proceso se sigue al leer los archivos “InfoEvento”, ”InfoComidas”, ”CuentaLogIn” y “Empleados”. Como en el ejemplo de la Figura 12.
   2. Los dos restantes se basan en leer una secuencia simple que no se necesita revisar el número de registros que existen. Como en el ejemplo de la Figura 14.
2. Método de guardar archivo
   1. Cada archivo cuenta con un método de guardar. Al igual que al leer el archivo el proceso de 4 archivos binarios es diferente al de los dos restantes. En los archivos “InfoEvento”, ”InfoComidas”, ”CuentaLogIn” y “Empleados”, se guarda el número de registros primero y posteriormente se recorre todo el arreglo del tipo del objeto respectivo usando los métodos “gets” guardando cada uno de los registros. Como en el ejemplo de la Figura 13.
   2. En los dos restantes simplemente se guarda la información tal cual, del mismo orden que se leyó. Como en el ejemplo de la Figura 15.

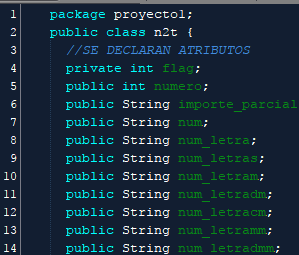
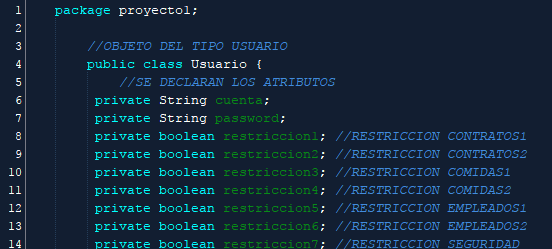
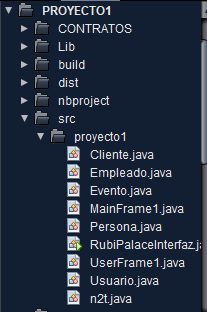
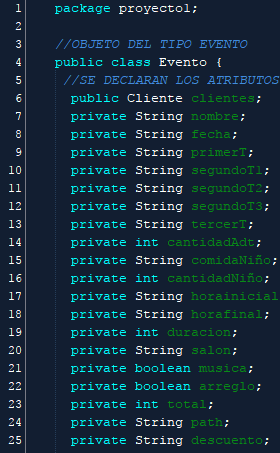
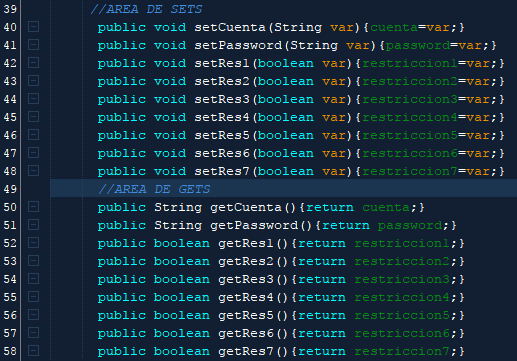
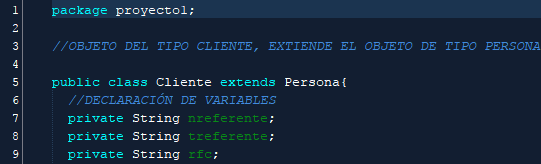
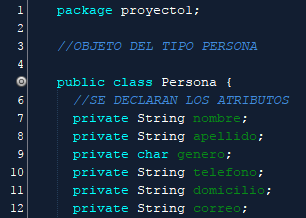
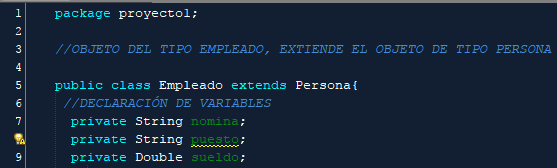
3-. Librerías

Para la realización de este proyecto se descargaron varias librerías de internet, encontradas en la figura 16. Se basaron para dos funciones principalmente, la primera para la implementación de un calendario (org.jdatepicker) y la segunda fue el API “APACHE POI” para manipular información dentro de archivos de Word (org.apache.poi). La primera se implementó con el fin de facilitarle al cliente el ingreso de fechas y la segunda con el fin de facilitar al cliente la creación de contratos.

1. Para la implementación del calendario se importó-“org.jdatepicker.impl.\*”. Se crearon objetos de tipo UtilDateModel y Properties se inicializo el segundo adecuadamente y se mandó junto con el primero como parámetros al constructor del nuevo objeto tipo JDatePanelImpl. También se creó un objeto de tipo AbstractFormatter. Por último se creó el componente JDatePickerImpl mandando como parámetros el JDatePanelImpl y el AbstractFormatter. Tal como se muestra en la Figura 17.
2. Para la implementación a la modificación de archivos de Word se importaron los “imports” encontrados en la Figura 18. Para su desarrollo se creó un método que recibía una matriz bidimensional que contenía en la primera columna la palabra que se encontraba actualmente en el archivo y en la segunda columna la palabra nueva, además el método recibía el número de contrato para establecer un nombre. Este método crea una copia del templeate y posteriormente empieza a llenarlo palabra por palabra recorriendo la matriz como se aprecia en la Figura 20 de la línea 282 a 287, ejemplo de inicialización de la matriz y llamado de método en la figura 19. En el método se creó un objeto de tipo XWPFDocument que hace referencia al documento, el método analiza todo su contenido y posteriormente inserta la información deseada con el método “.replace(x[i] [j], x[i] [j+1];” recibiendo como parámetro la palabra vieja (Columna 0)-“x[i] [j]” y la palabra nueva (Columna 1)-“x[i] [j+1]”. Tal como se muestra en la Figura 20 en la línea 285.

4-. Menús y Sub Menús

Para la implementación de menús y sub menús se usó un panel con un “CardLayout”. Se usó de esta forma con el fin que el menú principal siempre este presente, abriendo la posibilidad de cambiar a los submenús y a otros paneles. En la figura 21 se encuentra circulado en rojo el Menú principal siempre presente y en verde un ejemplo de un submenú, el lugar del submenú es donde se muestran todos los paneles incluidos en el “CardLayout” con su respectiva referencia, como se puede observar al comparar las figuras 21 y 22. Por ultimo en la figura 23, se muestra como se agrega cada panel con su referencia y cuantos paneles en totalidad tiene el panel con “CardLayout”.



*FIGURA 10*

*FIGURA 8*

*FIGURA 9*

*FIGURA 7*

*FIGURA 1*

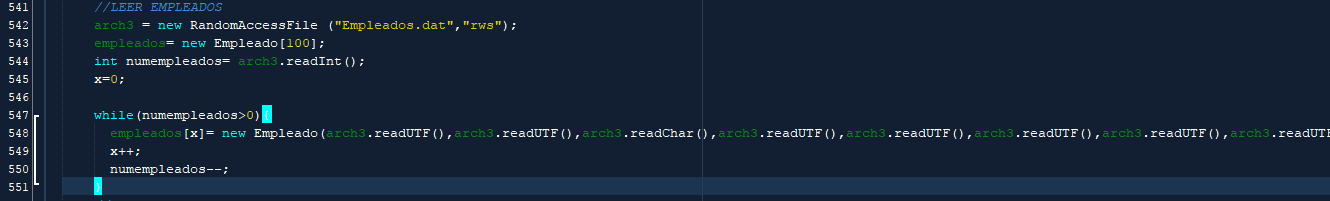
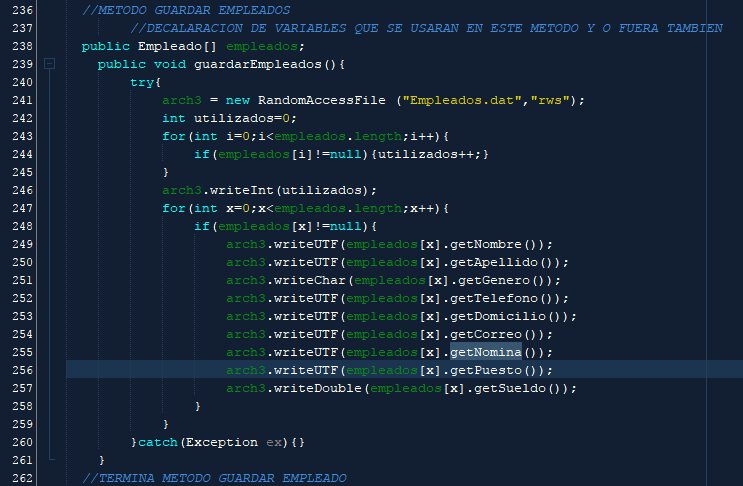
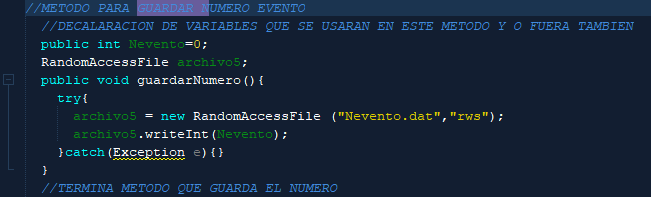
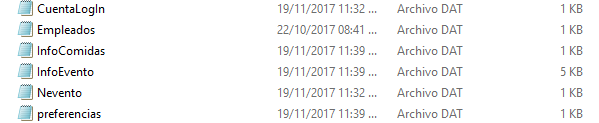
*FIGURA 4*

*FIGURA 6*

*FIGURA 5*

*FIGURA 3*

*FIGURA 2*



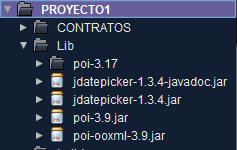
*FIGURA 11*

*FIGURA 15*

*FIGURA 14*

*FIGURA 13*

*FIGURA 12*



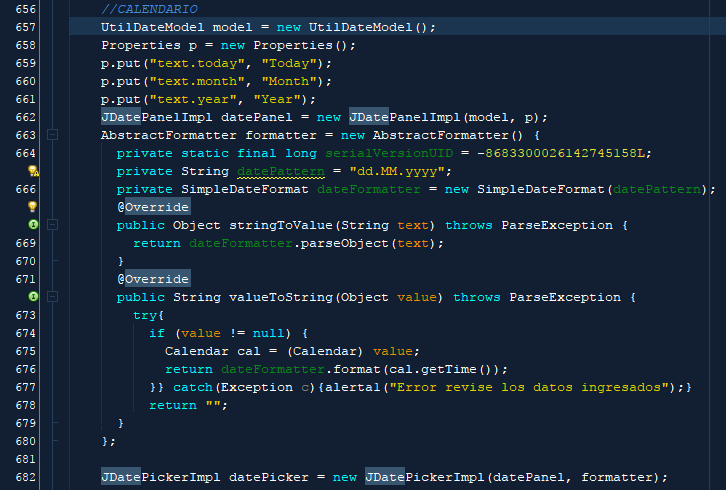
*FIGURA 18*



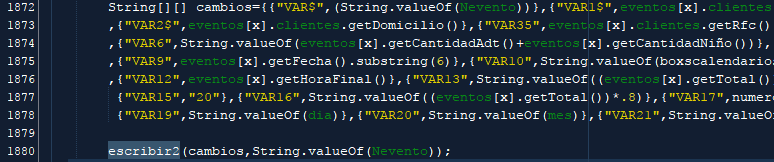
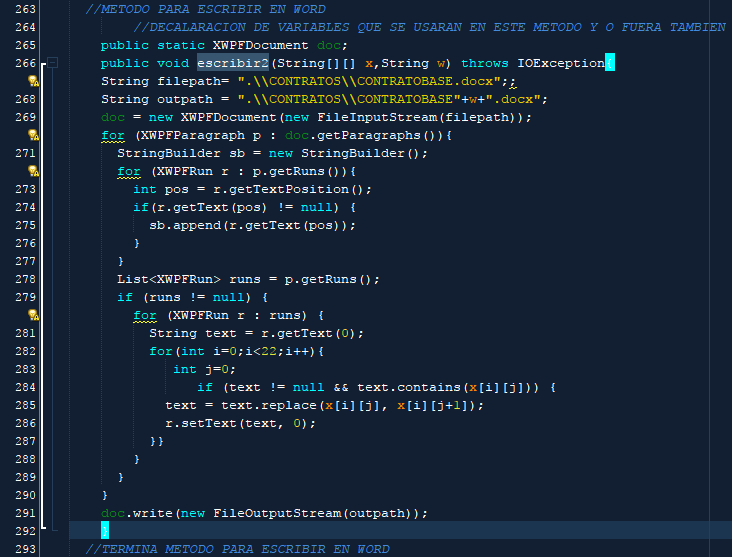
*FIGURA 20*

*FIGURA 19*

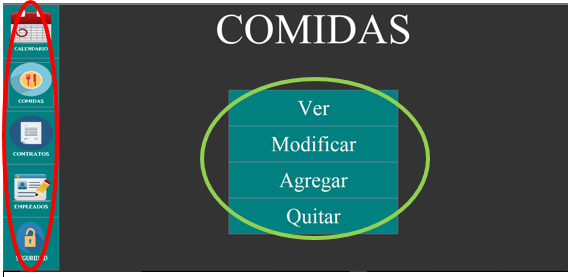
*FIGURA 16*

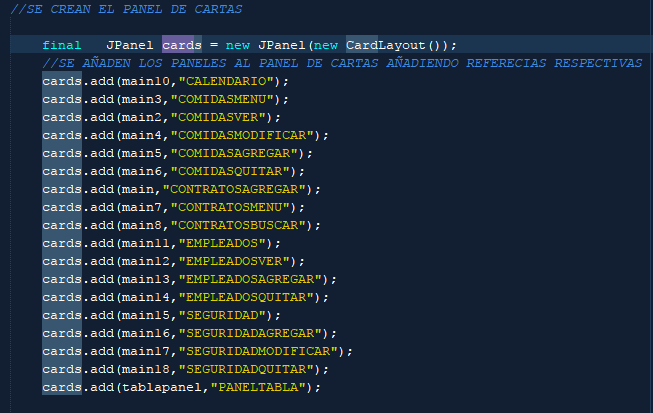


*FIGURA 17*



*FIGURA 19*





*FIGURA 23*

*FIGURA 22*

*FIGURA 21*

Referencias: Total de Palabras: 1093

CaveofProgramming. (2012, January 28). Advanced Java: Swing (GUI) Programming. Retrieved November 20, 2017, from <https://www.youtube.com/watch?v=DJqlT1d67jI&t=281s>

Font (Java Platform SE 7). Retrieved July 9, 2016, from <https://docs.oracle.com/javase/7/docs/api/java/awt/Font.html>

JComboBox (Java Platform SE 7). Retrieved August 6, 2016, from <https://docs.oracle.com/javase/7/docs/api/javax/swing/JComboBox.html>

Creating a JTable : JTable « Swing « Java Tutorial. Retrieved August 9, 2016, from <http://www.java2s.com/Tutorial/Java/0240__Swing/CreatingaJTable.htm>

JTable (Java Platform SE 7 ). Retrieved August 9, 2016, from <https://docs.oracle.com/javase/7/docs/api/javax/swing/JTable.html>

API for Microsoft Documents. (n.d.). Retrieved October 9, 2016, from <https://poi.apache.org/>

Stack Overflow - Where Developers Learn, Share, & Build Careers. (n.d.). Retrieved November 20, 2017, from <https://stackoverflow.com/>

How to use JDatePicker to display calendar component. (n.d.). Retrieved November 20, 2017, from <http://www.codejava.net/java-se/swing/how-to-use-jdatepicker-to-display-calendar-component>

JarDownload (n.d.). Retrieved November 20, 2017, from <http://www.java2s.com/Code/Jar/>

La web del programador (n.d.). Retrieved November 20, 2017, from <https://www.lawebdelprogramador.com/codigo/Java/338-Convertir-numeros-a-letras.html>