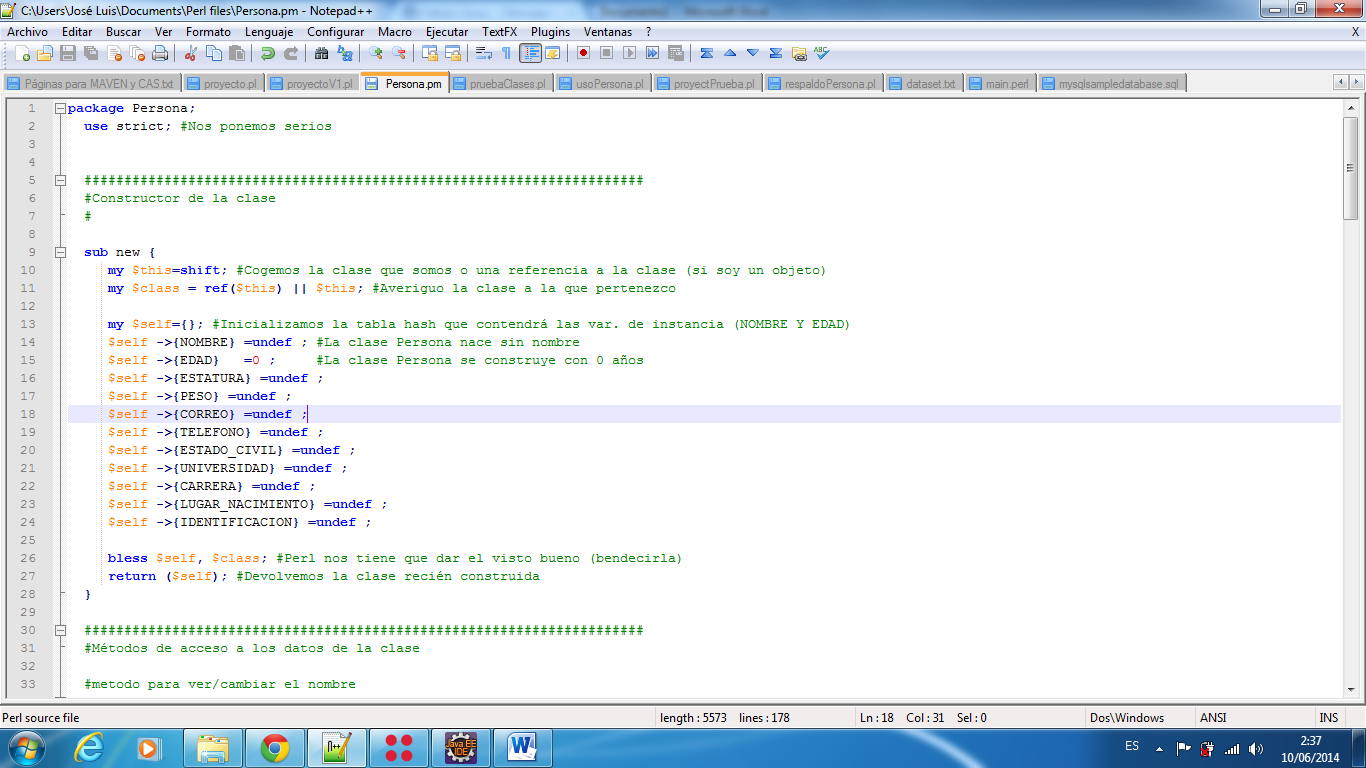
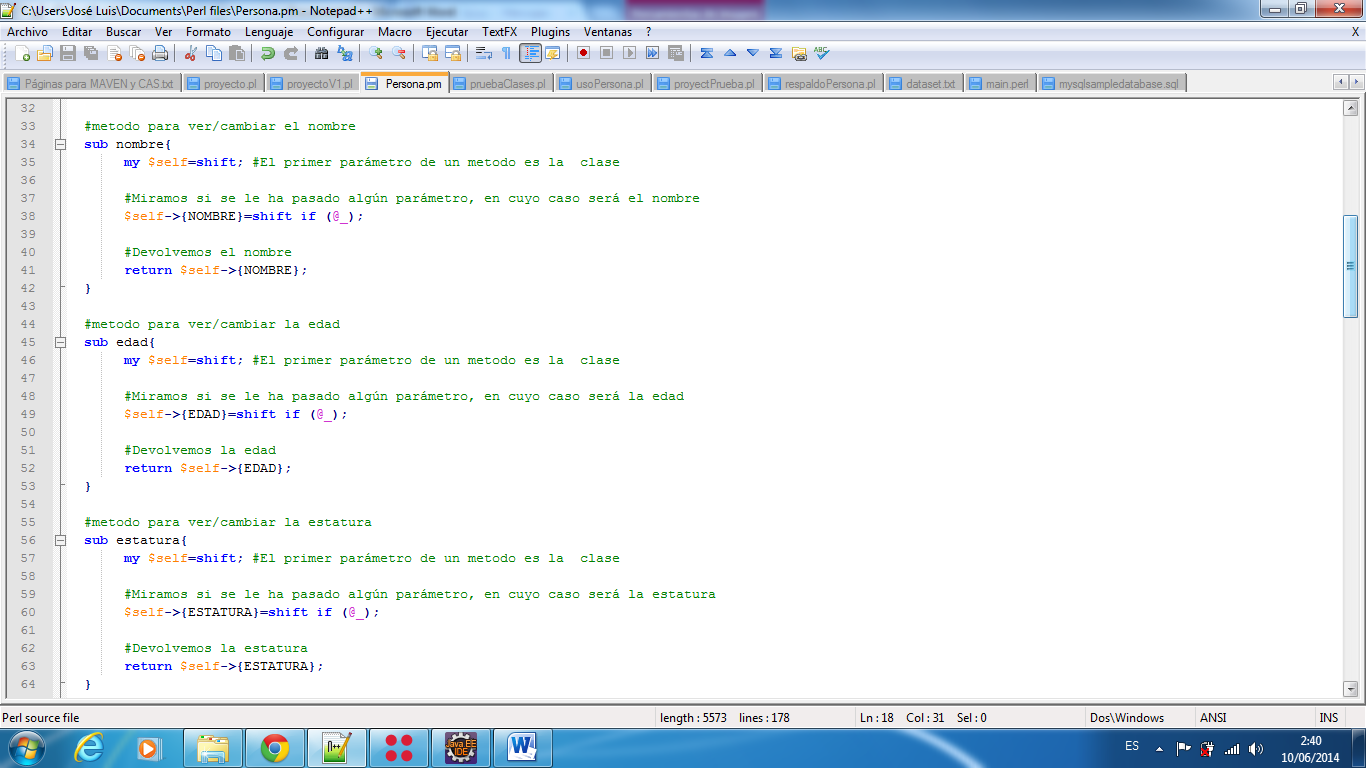
**DOCUMENTACION PROYECTO DE LENGUAJES DE PROGRAMACIÓN PRIMER PARCIAL**

**Creación de la clase Persona.**

Se creó el archivo llamado Persona.pm que consiste en un paquete el cual usamos en nuestro proyecto principal y que nos sirve para almacenar los datos de una persona, en la primera imagen se define el constructor de la clase con cada uno de los atributos que va a tener.



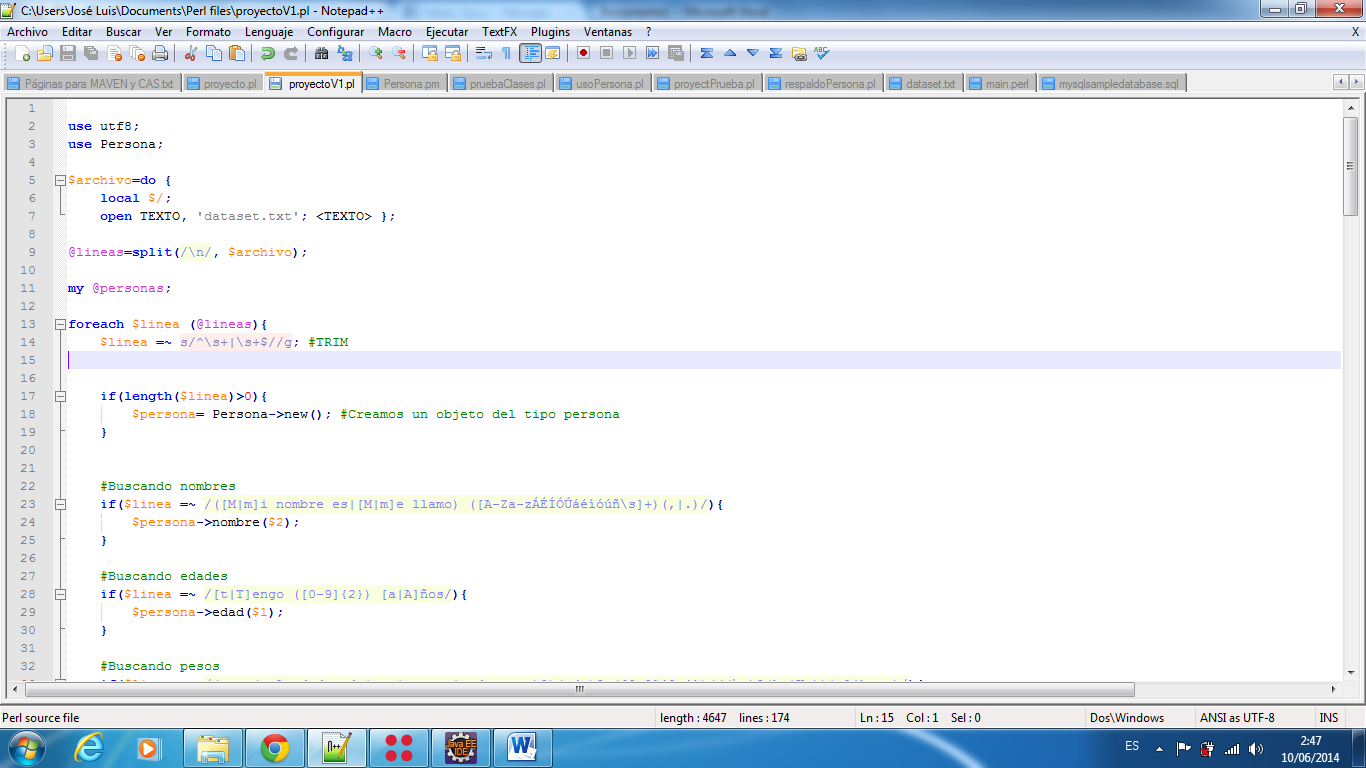
Luego se definieron los métodos que nos permiten el acceso a cada uno de los datos (atributos) de la clase. En la imagen se puede ver la implementación de los métodos para acceder al valor de las atributos nombre y edad, para el resto de atributos se definen métodos similares.



**Implementación del proyecto**

Se creó el archivo llamado proyectoV1.pl para implementar el código del proyecto, a continuación se detallará por secciones de código lo que se ha implementado.

**1.**

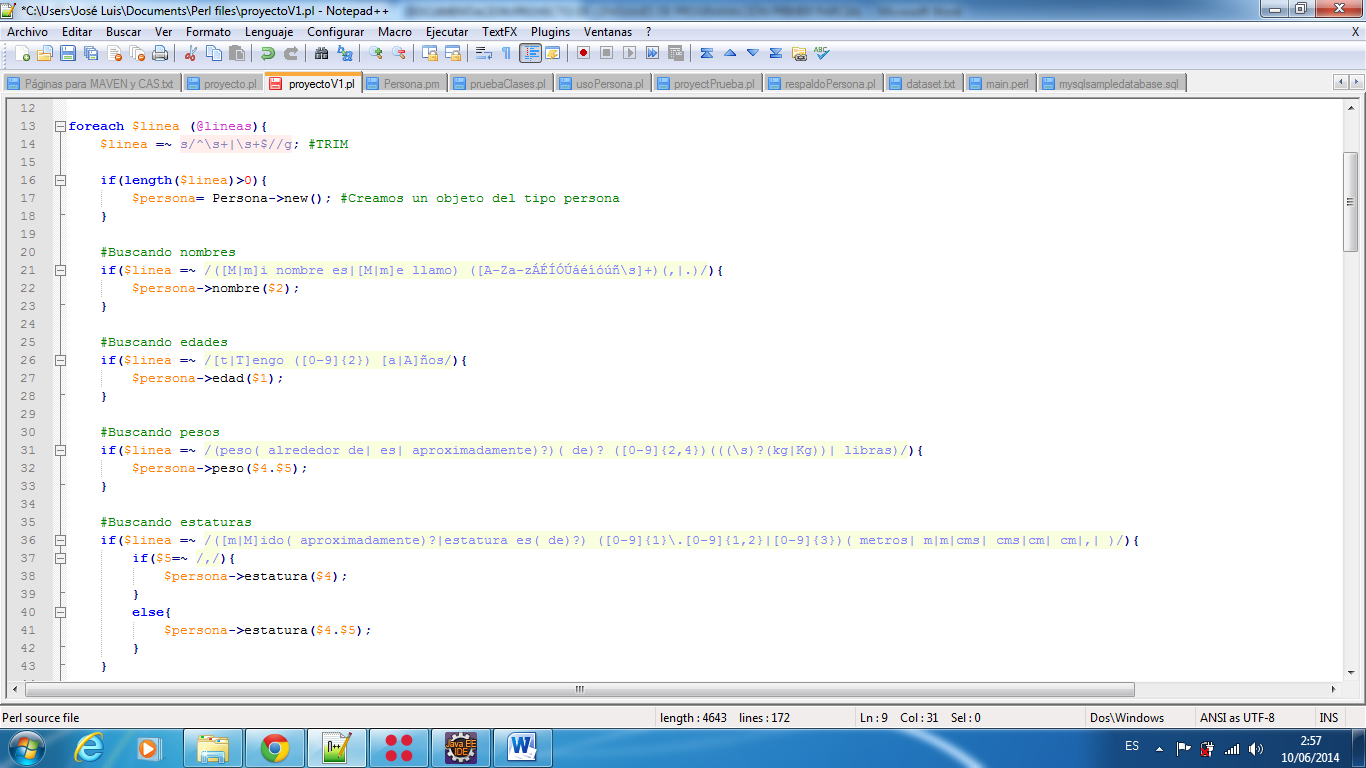


En esta primera parte se hace uso de utf8 para que se reconozcan los acentos y caracteres especiales cuando leamos el fichero “dataset.txt”, también se hace uso de la clase Persona que implementamos anteriormente.

A continuación se lee el fichero ‘dataset.txt’ y hacemos uso de la función split para separar la cadena en diversas cadenas, la separación se da por los “\n” de modo que cada una de las cadenas del arreglo representa la biografía de una persona, las cadenas se almacenan en la variable @lineas

Finalmente declaramos la variable “personas” que consiste en un array donde almacenaremos objetos “Persona”.

**2.**

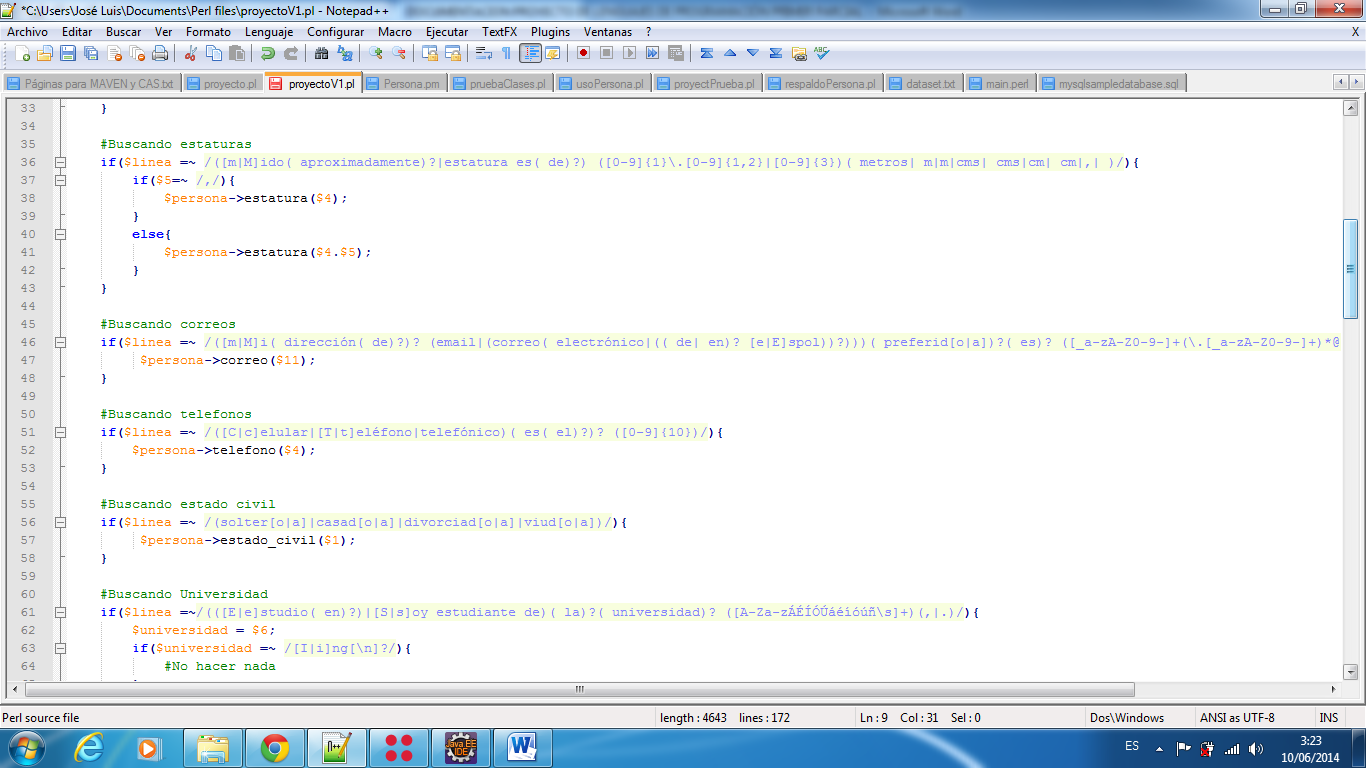


Recorremos el arreglo @lineas donde cada línea corresponde la biografía de una persona, luego hacemos uso de TRIM para eliminar espacios en blanco que puedan haber al principio o al final de la línea.

Si el tamaño de la línea es mayor a 0 significa que en esa línea hay información de una persona, entonces creamos una instancia de Persona.

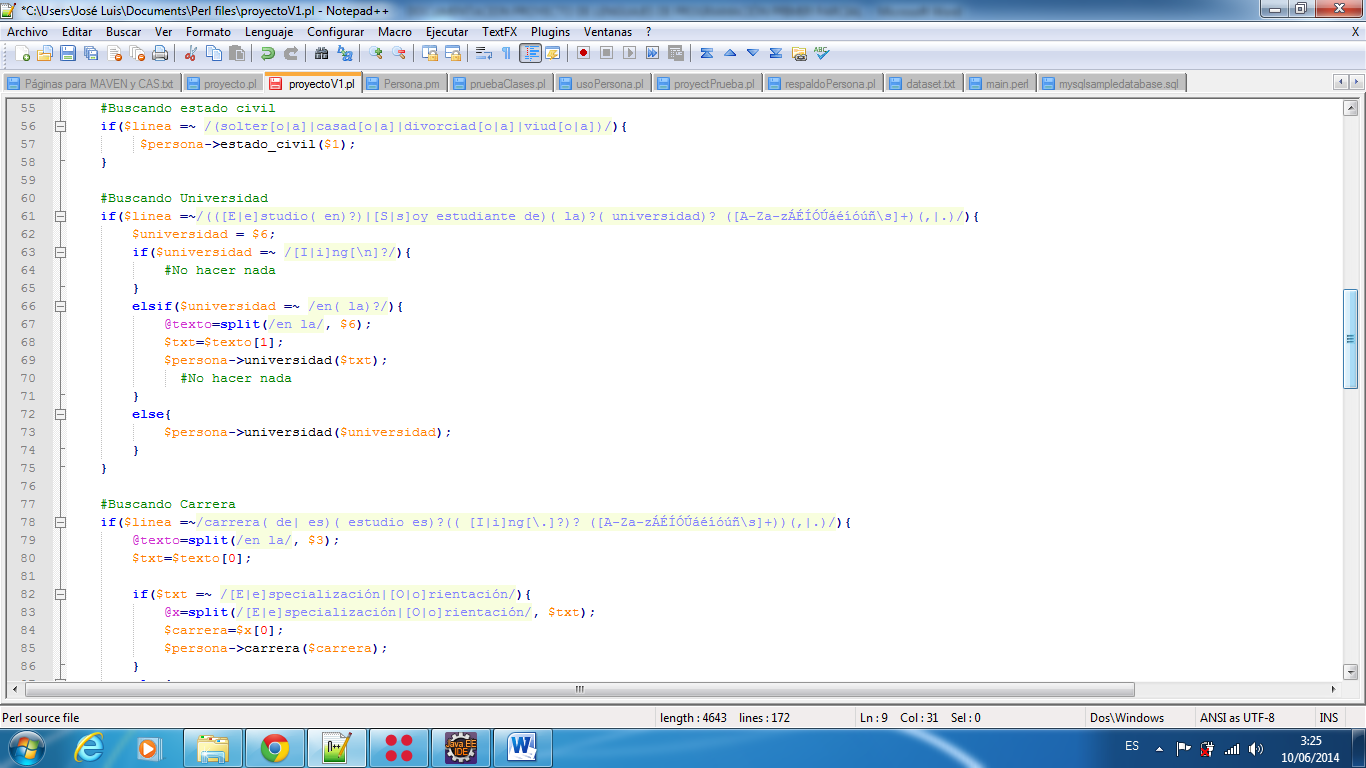
A continuación lo que hacemos es utilizar expresiones regulares para buscar coincidencias en la cadena según lo que se requiera, en la sección de código que se muestra en la imagen buscamos nombres, edades y pesos, la expresión regular se la hace según los patrones que hay en las cadenas al momento en que una persona especifica su nombre, edad, peso o lo que se quiera obtener. A cada expresión regular la separamos por paréntesis para almacenar variables temporales, de modo que podamos capturar el valor y almacenarlo en caso de que el patrón se haya cumplido.

Cada vez que se cumple un patrón en la expresión regular, guardamos el dato en el atributo que corresponda del objeto Persona que se creó al principio.

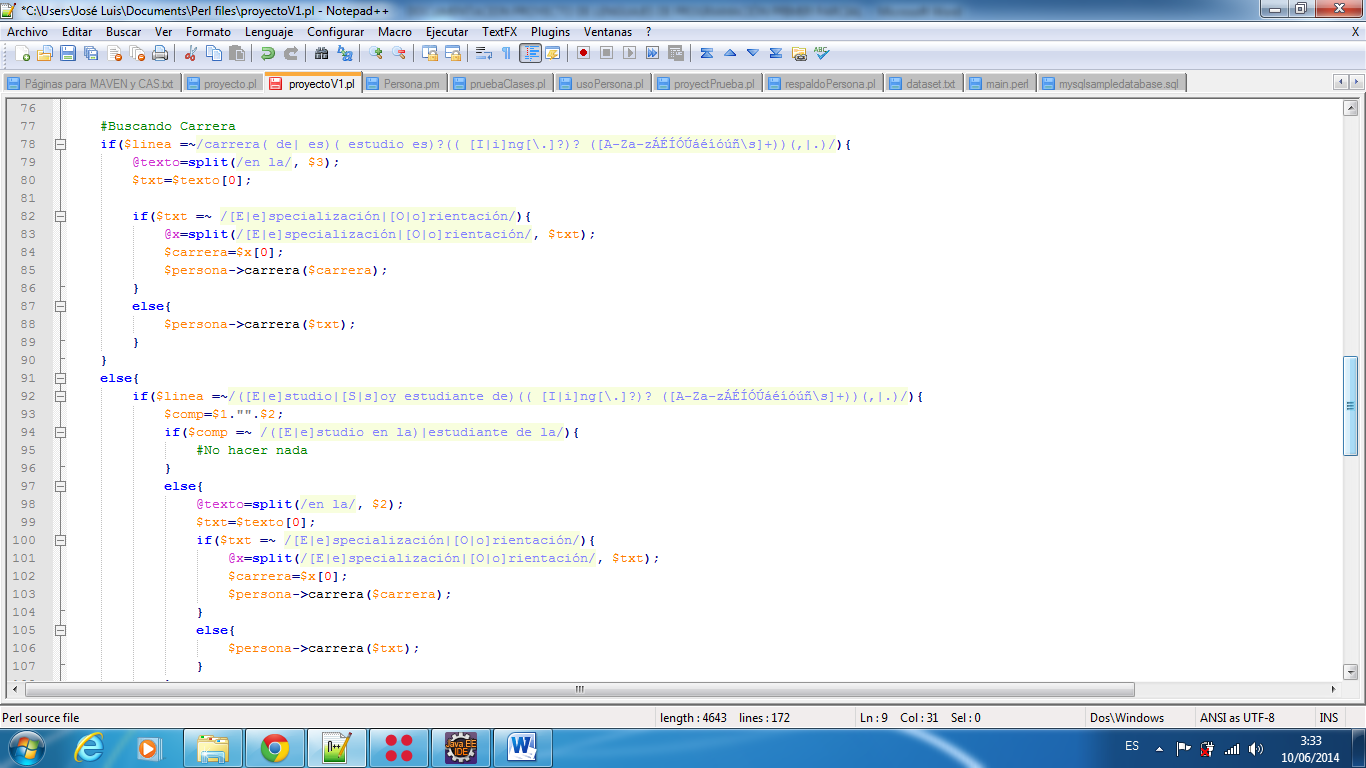


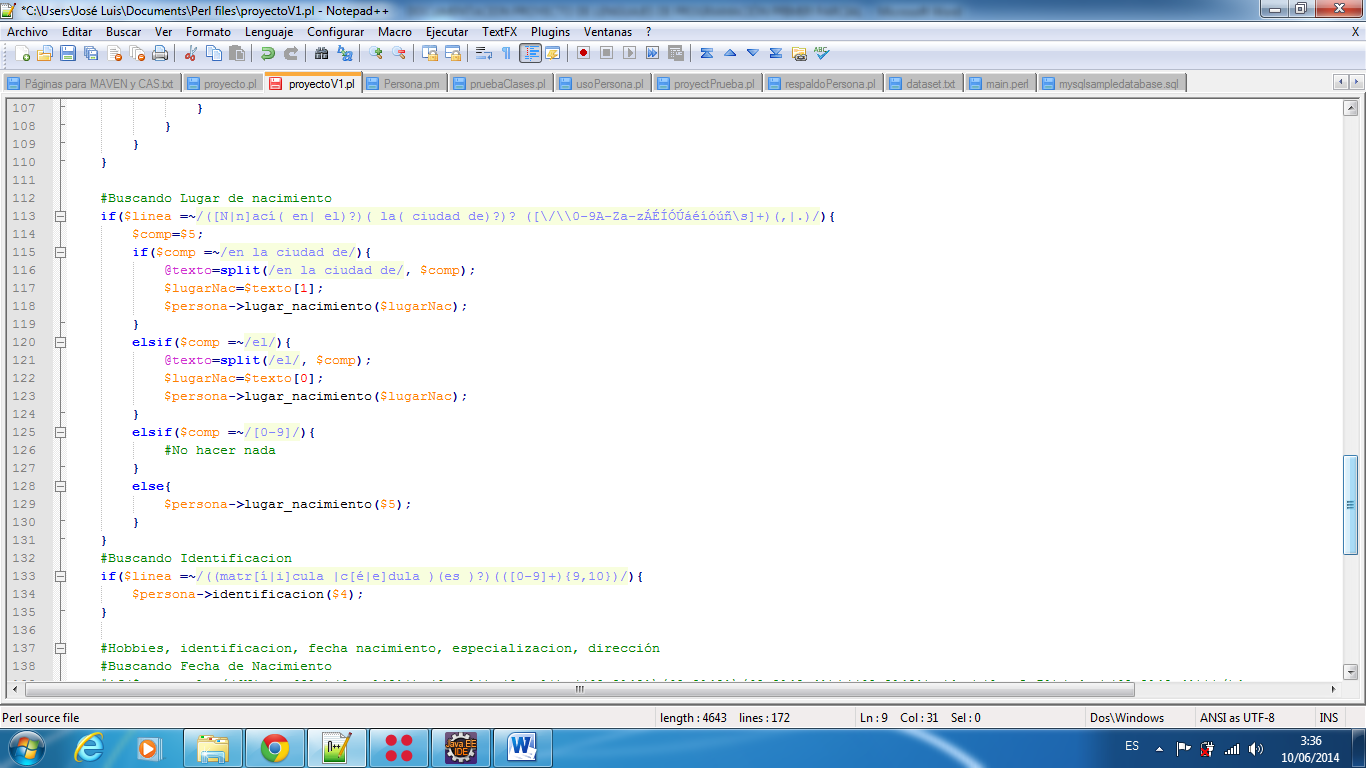
Repetimos el proceso de comparar la cadena con la expresión regular para verificar si hay más datos para guardar en el objeto Persona. En la imagen se encuentra el código para encontrar estatura, correo, teléfono y estado civil.

**3.**



Para encontrar la Universidad hacemos algo adicional, como se puede ver en el código, la variable temporal $6 debería guardar el valor de universidad, pero hay casos en que se guardan cadenas que no corresponden solo a la universidad, entonces es necesario volver hacer una comparación para ver si en la cadena se encuentra el texto “en la”, por lo general se encuentran estas líneas porque en el texto especifican la carrera y la universidad de manera seguida como por ejemplo: “estudio ingeniería en computación en la Espol”. Si se da el caso que se encuentra la cadena “en la” entonces debo usar la función Split para separar la cadena mediante el separador “en la” donde se almacenará la universidad en la segunda cadena del arreglo que devuelve split.

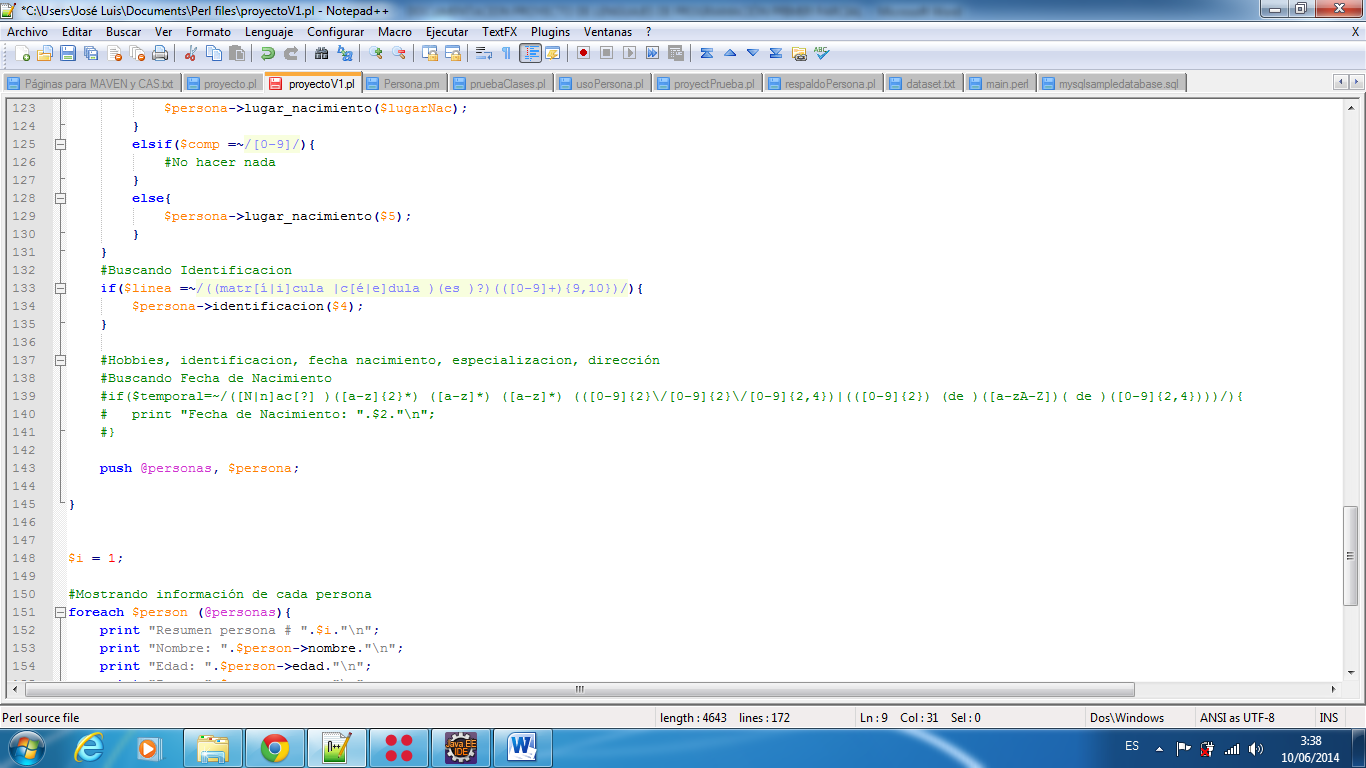




Para buscar la carrera y el lugar de nacimiento, suelen ocurrir problemas similares a los que ocurren cuando buscamos la universidad, por lo que es necesario volver a hacer comparaciones y hacer uso de la función split para separar la cadena.

**4.**

Una vez que se ha pasado por todas las búsquedas de datos, lo que falta es añadir el objeto persona, que se va creando cada vez que se ingresa al lazo for-each, al arreglo de personas que se definió en un principio, se lo hace con la siguiente línea de código.



**5.**

Finalmente, cuando se termina de recorrer todo el archivo “dataset.txt”, tenemos nuestro arreglo de personas donde está almacenada la información de cada una de las personas que se encontraban en el archivo. Lo único que queda ahora es recorrer dicho arreglo de personas y mostrar los resultados obtenidos.

