UNIVERSIDAD POLITÉCNICA DE PUEBLA
Ingeniería en Informática

Proyecto de Estancia Práctica en Desarrollador en Sistemas de Software y Administrador de Redes

"Análisis de información aplicando minería de datos"

Área temática del CONACYT: VII
Ingenierías y tecnologías

Presenta:
Obed Zeferino Ureiro Ruiz

Asesor técnico
Dr. Jorge De la Calleja Mora

Asesor académico
M.C. Rebeca Rodríguez Huesca

Juan C. Bonilla, Puebla, México. 19 de diciembre de 2018

Resumen

En este documento se encontrara una visión general de un sistema de análisis de datos el cual utiliza minería de datos, se anexa al documento los siguientes elementos: la razón por la cual se desarrolló el proyecto y los objetivos a alcanzar, la metodología que se siguió durante el periodo de vida del proyecto, las herramientas que se utilizaron tanto sus ventajas y desventajas, y los resultados que se obtuvieron siguiendo la metodología descrita.

El sistema fue realizado con la metodología por prototipos, en el lenguaje de programación de Python versión 2.7.9, utilizando las librerías de: Numpy versión 1.15.2, Matplotlib versión 2.2.3, SciPy versión 1.1.0 y PyQt4 versión 4.8.7. Con respecto a los algoritmos de minería de datos se utiliza la librería de Weka y la librería de javabridge 1.0.18, se recalca que se debe utilizar al menos como versión mínima las versiones mencionadas por sobretodo la version de Python dado a que existen muchas incompatibilidades entre las librerías.

Índice

1. Introducción

En este capítulo se presentará: la problemática o necesidad de la situación que se desea solucionar, la justificación de la propuesta, el objetivo general y los objetivos específicos.

1.1. Descripción del problema o necesidad

Se requiere un sistema sencillo, fácil de entender y que sea capaz de analizar una gran cantidad de datos de una forma rápida para que se pueda realizar una toma de decisiones a partir de los resultados arrojados.

1.2 Justificación

Actualmente es difícil integrar y configurar diferentes algoritmos de minería de datos en un sólo sistema dado a que ocupan un gran espacio de memoria y no pueden ser tan óptimos para procesar grandes volúmenes de información, por esta razón es necesario desarrollar un sistema amigable para aquellas personas que no son expertas en el área y que no tengan ningún problema el rendimiento y en la interpretación de los resultados.

1.3 Objetivo General y Específicos

Realizar un sistema de análisis de información utilizando distintos algoritmos de minería de datos

Objetivos Específicos

- Investigar la herramienta Weka.
- Investigar las librerías necesarias para Python para implementar Weka.
- Investigar los métodos más comunes de minería de datos.
- Adecuar los algoritmos de Weka en Python.
- Realizar las interfaces gráficas del sistema.
- Desarrollar un manual de usuario.
- Desarrollar un manual técnico.

2. Metodología y herramientas

En este capítulo se presentará: la metodología a seguir, las ventajas y desventajas de las herramientas tecnológicas que se ocuparan en el desarrollo del sistema.

2.1 Modelo por prototipos

El paradigma de hacer prototipos es una técnica iterativa que se utiliza cuando no se conoce exactamente cómo desarrollar un determinado producto, su objetivo primordial es lograr un producto intermedio y que este sea evaluado, para realizar un producto final con las especificaciones que se requieran.

El paradigma se compone de las siguientes fases [1], como se muestra en la Figura 1:

1. **Comunicación**
   El grupo de trabajo se reúne para definir los objetivos generales del software e identifican los requerimientos y las áreas de mayor de impacto.
2. **Plan rápido**
   Se planea una iteración del para hacer el prototipo si es necesario.
3. **Modelado, diseño rápido**
   Se centra en los aspectos del software que serán visibles para los usuarios finales.
4. **Construcción del prototipo**
   Es el periodo de codificación del prototipo.
5. **Despliegue, entrega y retroalimentación**
   El prototipo es entregado y evaluado por los participantes, esto permite entender mejor lo que se necesita implementar.

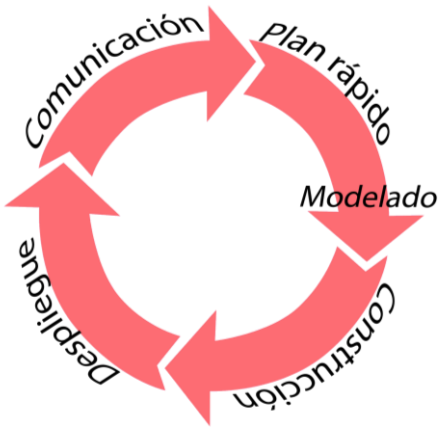

Figura 1 Ciclo de vida del modelo por prototipos

2.2 Herramientas tecnológicas utilizadas

En esta sección se presentarán las herramientas tecnológicas utilizadas, mostrando una breve descripción y sus principales ventajas y desventajas.

2.2.1 Python

Python [2] es un lenguaje de programación interpretado multi-paradigma puede soportar varios módulos y librerías, actualmente existen 3 versiones de este lenguaje pero las versiones 2 y 3 son las más utilizadas además se puede ejecutar en varios sistemas operativos como: Unix, Linux, Windows, Mac. La razón principal de su uso es que ocupa menos recursos que otros lenguajes de programación y es necesario procesar una gran cantidad de datos.

Ventajas

- Tiene un mejor rendimiento con el proceso de datos con respecto a otros lenguajes de programación.
- Se reduce las líneas de código gracias a su indentación.
- Es un lenguaje de fácil instalación y con amplia documentación.
- Es utilizado para la ciencia de datos.
- Existen varios módulos y librerías que facilitan el uso de funciones matemáticas además de una forma de representarlos gráficamente.

Desventajas

- No cuenta con un ambiente grafico que facilite la creación de código.
- No hay compatibilidad entre versiones.
- Es necesario reescribir todo el sistema para actualizar algún complemento de una librería o modulo.
- Toma más tiempo la codificación de un sistema.
- Se necesita de herramientas externas para producir más líneas de código en menor tiempo.

2.2.2 Weka

Weka [3] es un software gratuito que cuenta una colección de algoritmos de aprendizaje automático para tareas de minería de datos cuenta con una licencia libre y actualmente puede trabajar con distintos lenguajes de programación como lo puede ser: Java, Python, C#, además de contar con una amplia documentación. La razón principal de su uso es que es un software gratuito y con una gama amplia de algoritmos además es actualizado constantemente.

Ventajas

- Cuenta con una amplia sección de algoritmos por ejemplo: de clasificación, agrupación, selección, visualización y predicción.
- Es fácil la configuración y la manipulación de los algoritmos.

- Se pueden descargar e instalar paquetes adicionales desde un repositorio oficial.
- Se puede utilizar en distintos sistemas operativos como: Linux, Windows, Mac, Unix.
- No es necesario contar con conocimientos avanzados en el tema para utilizar la herramienta.

Desventajas

- Es necesario ciertas extensiones de archivos para la lectura de datos.
- Anqué existe documentación del software no hay una traducción al español.
- Dado a que los algoritmos que utiliza son muy complejos el sistema puede consumir muchos recursos.
- Es difícil encontrar algunos recursos o materiales del sistema dado a que el instituto tiene dichos archivos y son únicos para los alumnos.

3. Resultados

En este capítulo se mostrarán los resultados obtenidos a través de las distintas iteraciones de la metodología por prototipos.

3.1. Comunicación

A continuación en las Tablas: 1, 2, 3, 4, 5, 6 y 7 se muestran las historias de usuario, las cuales contienen un breve título, la actividad a realizar y las observaciones del alumno.

Tabla 1 Historia de Usuario 1: Implementación del algoritmo simple K-Means

		Historia de Usuario
ID:1	**Usuario:** Asesor Técnico	Iteración asignada: 1
Nombre historia: Implementación del algoritmo simple K-Means		
Descripción: El sistema contendrá el algoritmo simple K-Means, deberá mostrar los resultados de dicho algoritmo y mostrar una gráfica de clasificación de los datos de un archivo.		
Observaciones: • Investigar el funcionamiento del algoritmo simple K-Means. • Investigar la documentación de la librería de Weka, dado que cuenta con el algoritmo dicho. • El archivo tendrá una extensión .arff o .csv.		

Tabla 2 Historia de Usuario 2: Función de graficación

		Historia de Usuario
ID:2	**Usuario:** Asesor Técnico	Iteración asignada: 2
Nombre historia: Función de graficación		
Descripción: Se requiere una función que sea capaz de graficar los atributos y las instancias de un achivo, utilizando las librerías de Matplotlib, Weka y Numpy.		
Observaciones: • Investigar la librería de Matplotlib • Investigar la librería de Numpy.		

Tabla 3 Historia de Usuario 3: Interfaces gráficas en PyQt4

		Historia de Usuario
ID:3	**Usuario:** Asesor Técnico	Iteración asignada: 3
Nombre historia: Interfaces gráficas en PyQt4		
Descripción: Realizar únicamente las ventanas y sub ventanas que contendrá el sistema con la librería PyQ4.		
Observaciones: • Investigar la librería PyQt4. • Realizar el boceto de las interfaces gráficas.		

Tabla 4 Historia de Usuario 4: Implementación de la sub ventana Archivo

Historia de Usuario		
ID:4	**Usuario:** Asesor Técnico	Iteración asignada: 4
Nombre historia: Implementación de la sub ventana Archivo		
Descripción: Codificar la sub ventana Archivo para que muestre: • El nombre del archivo • Número de atributos • Número de instancias • Mostrar una tabla ordenada con los nombres y el tipo de los atributos		
Observaciones:		

Tabla 5 Historia de Usuario 5: Implementación de la sub ventana Algoritmo

Historia de Usuario		
ID:5	**Usuario:** Asesor Técnico	Iteración asignada: 4
Nombre historia: Implementación de la sub ventana Algoritmo		
Descripción: Codificar la sub ventana Algoritmo para que realice el algoritmo simple K-Means. El número de clusters será seleccionado por el usuario		
Observaciones:		

Tabla 6 Historia de Usuario 6: Implementación de la sub ventana Gráficas

Historia de Usuario		
ID:6	**Usuario:** Asesor Técnico	Iteración asignada: 4
Nombre historia: Implementación de la sub ventana Gráficas		
Descripción: Codificar la sub ventana Gráficas para que realice la función realizada anteriormente. El usuario seleccionara el eje “x”, “y” junto con el color deseado para graficar.		
Observaciones: • Adaptar la función desarrollada para que sea soportada en PyQt4		

Tabla 7 Historia de Usuario 7: Implementación de la ventana Principal

Historia de Usuario		
ID:7	**Usuario:** Asesor Técnico	Iteración asignada: 4
Nombre historia: Implementación de la ventana Principal		
Descripción: Codificar la ventana principal, la cual podrá crear “n” sub ventanas de las anteriores desarrolladas.		
Observaciones: • Se debe implementar la máquina virtual de java para que pueda funcionar la librería de Weka sin problemas		

3.2. Plan rápido

A continuación en la Tabla 8, se muestra el plan de trabajo que se siguió durante el periodo del proyecto.

Tabla 8 Plan de trabajo

Iteración	Actividades	14 septiembre	21 septiembre	28 septiembre	5 octubre	12 octubre	19 octubre	26 octubre	2 noviembre	9 noviembre	16 noviembre	23 noviembre	30 noviembre	7 diciembre
1	Investigación de simple K-Means	X												
	Investigación de la librería de Weka		X											
	Implementación de simple K-Means en Python			X										
2	Investigación de la librería Numpy				X									
	Investigación de la librería Matplotlib					X								
	Implementación de la función de graficación						X							
3	Investigación de la librería PyQt4							X						
	Boceto de las interfaces gráficas								X					
	Implementación de las interfaces gráficas								X					
4	Codificación de la sub ventana Archivo									X				
	Codificación de la sub ventana Algoritmo										X			
	Codificación de la sub ventana Gráficas											X		
	Codificación de la ventana Principal												X	
	Elaboración de documentación del sistema													X

3.3. Modelado, Diseño rápido

En esta sección se mostrara los diagramas de actividad de las sub ventanas del sistema, como se puede ver en las Figuras 2, 3 y 4**.**

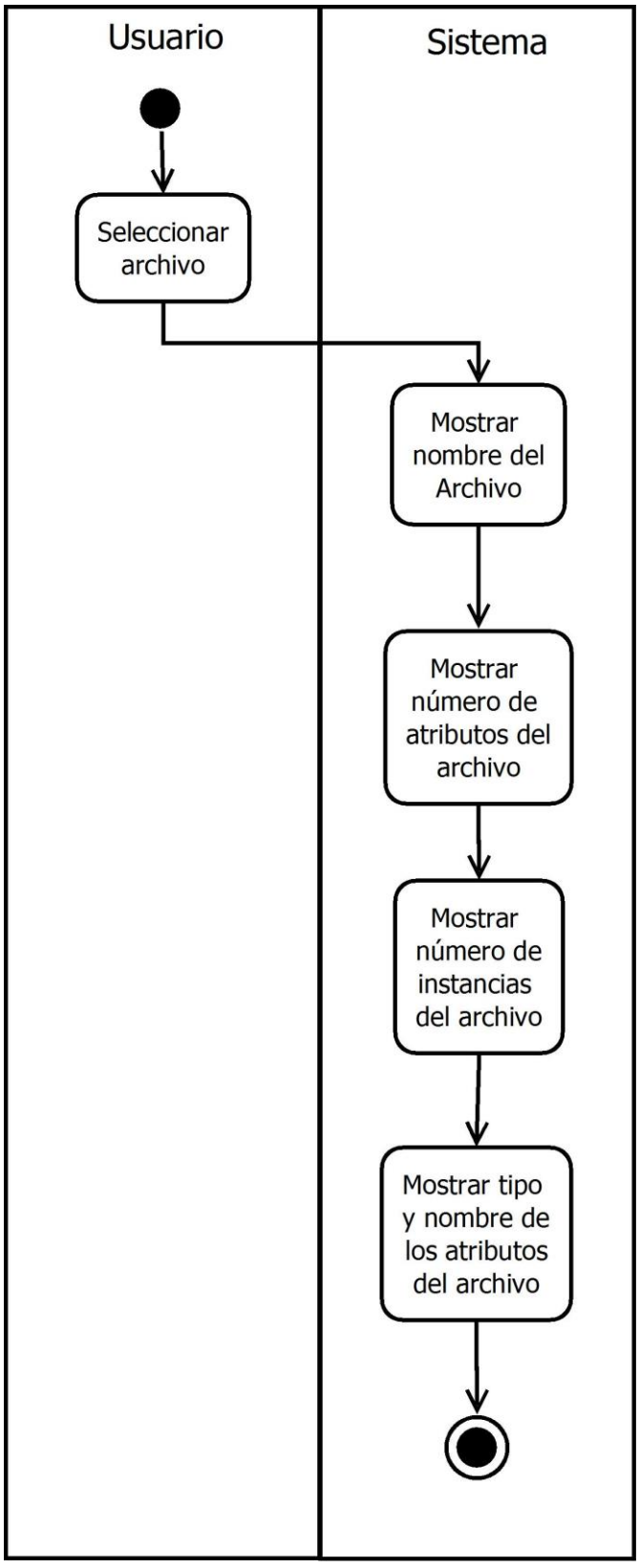

Figura 2 Diagrama de actividad de la sub ventana Archivo

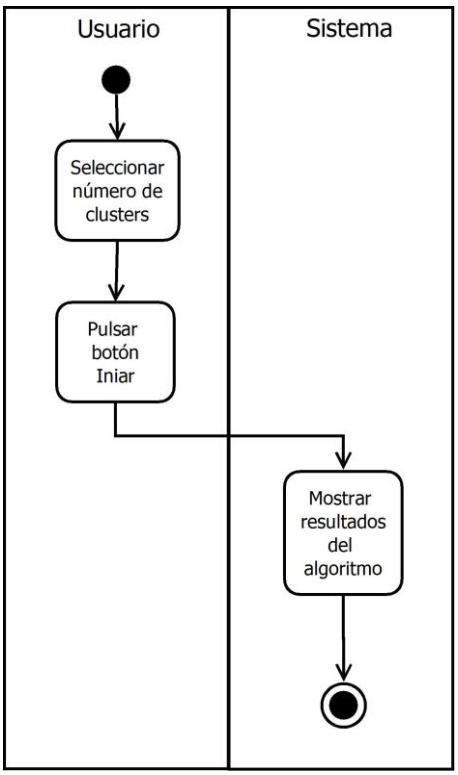

Figura 3 Diagrama de actividad de la sub ventana Algoritmo

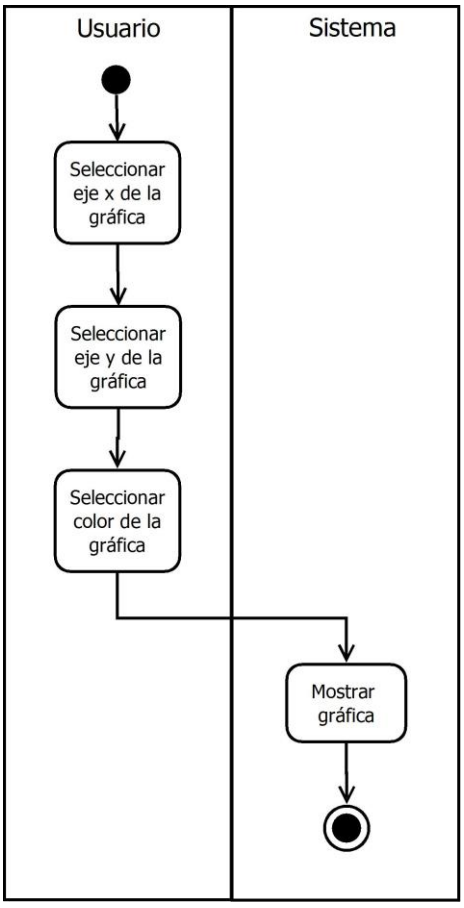

Figura 4 Diagrama de actividad de la sub ventana Gráficas

En las figuras 5, 6, 7 y 8 se muestran los bocetos de las interfaces gráficas del sistema.

Figura 5 Diseño de la ventana principal

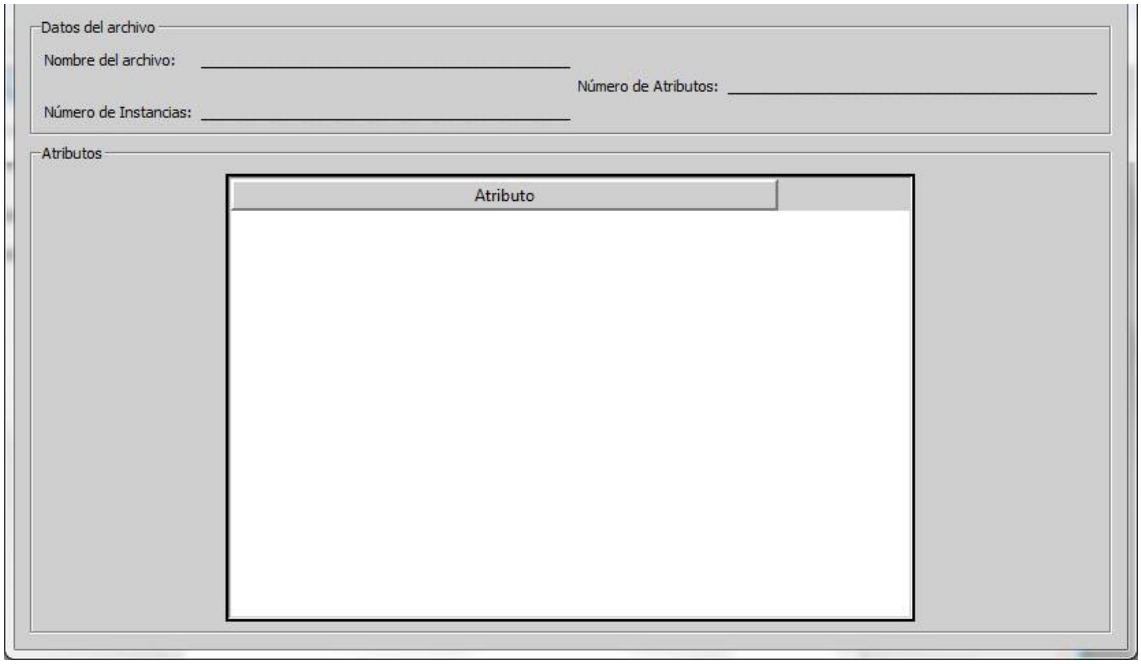

Figura 6 Diseño de la ventana Archivo

Figura 7 Diseño de la ventana Algoritmo

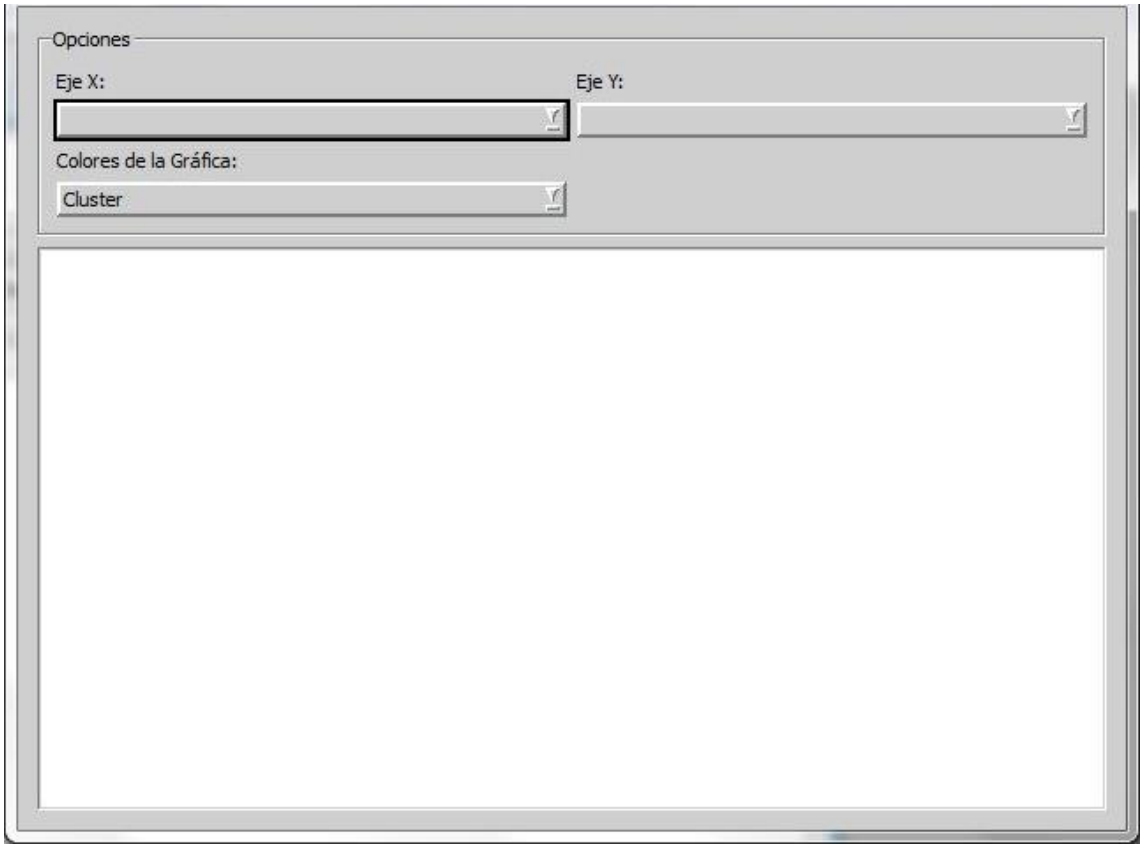

Figura 8 Diseño de la ventana Gráficas

3.4. Construcción del prototipo

En esta fase se realiza toda la codificación del diseño anterior, en las Figuras 9 y 10 se puede apreciar la codificación de la primera iteración, la cual consiste en la implementación del algoritmo simple K-Means en Python y que muestre una gráfica de clasificación.

```
kMeans
======

Number of iterations: 3
Within cluster sum of squared errors: 7.817456892309574

Initial starting points (random):

Cluster 0: 6.1,2.9,4.7,1.4,Iris-versicolor
Cluster 1: 6.2,2.9,4.3,1.3,Iris-versicolor
Cluster 2: 6.9,3.1,5.1,2.3,Iris-virginica

Missing values globally replaced with mean/mode

Final cluster centroids:
                        Cluster#
Attribute               Full Data           0           1           2
                        (150.0)         (50.0)      (50.0)      (50.0)
================================================================================
sepallength               5.8433          5.936       5.006       6.588
sepalwidth                 3.054           2.77       3.418       2.974
petallength               3.7587           4.26       1.464       5.552
petalwidth                 1.1987          1.326       0.244       2.026
class                   Iris-setosa Iris-versicolor Iris-setosa  Iris-virginica


Clustered Instances

0      50 ( 33%)
1      50 ( 33%)
2      50 ( 33%)
```

Figura 9 Resultados del algoritmo Simple K-Means sobre el archivo Iris.arff

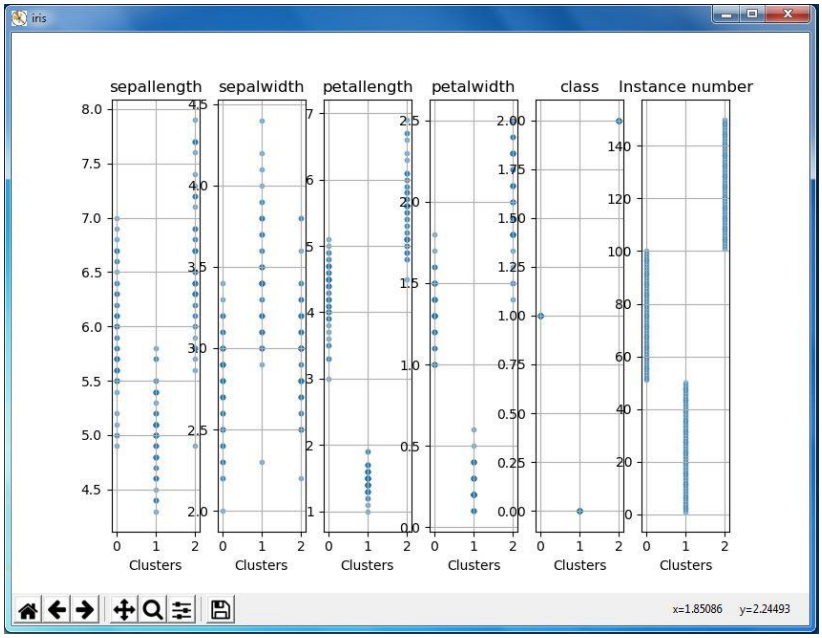

Figura 10 Gráfica de clasificación de clusters del algoritmo Simple K-Means

En las Figuras 11,12,13 se puede apreciar la codificación de la segunda iteración, la cual es el desarrollo de una función que sea capaz de graficar el resultado del algoritmo simple K-Means, todas las gráficas mostradas son evaluadas en el primer atributo encontrado sobre el archivo, en este caso es el atributo “Sepallength”.

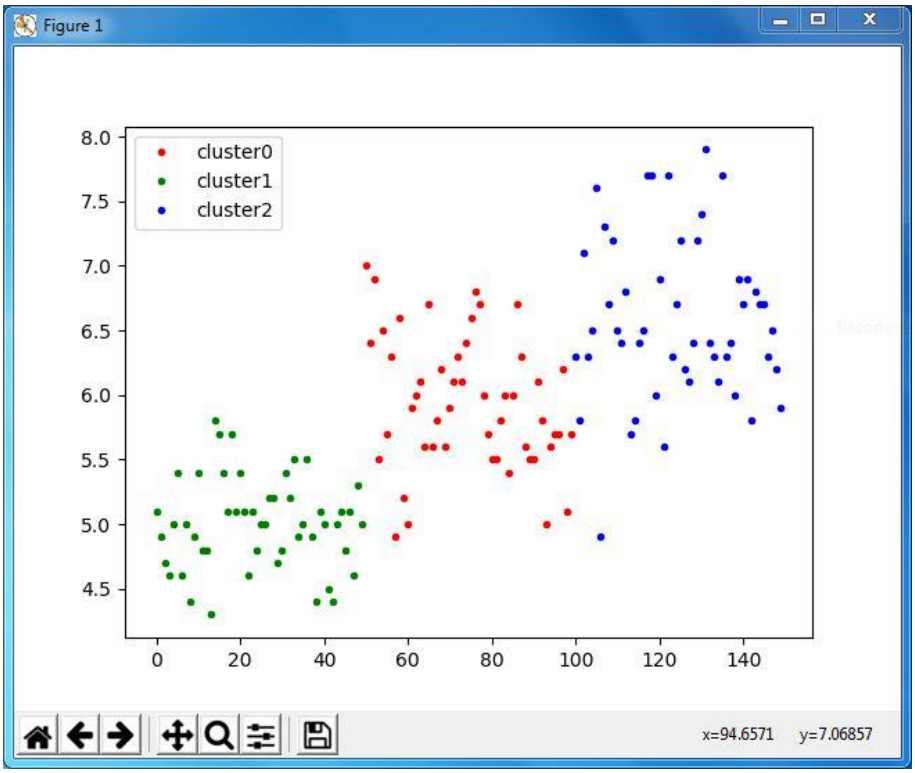

Figura 11 Gráfica de Instancias X Atributo

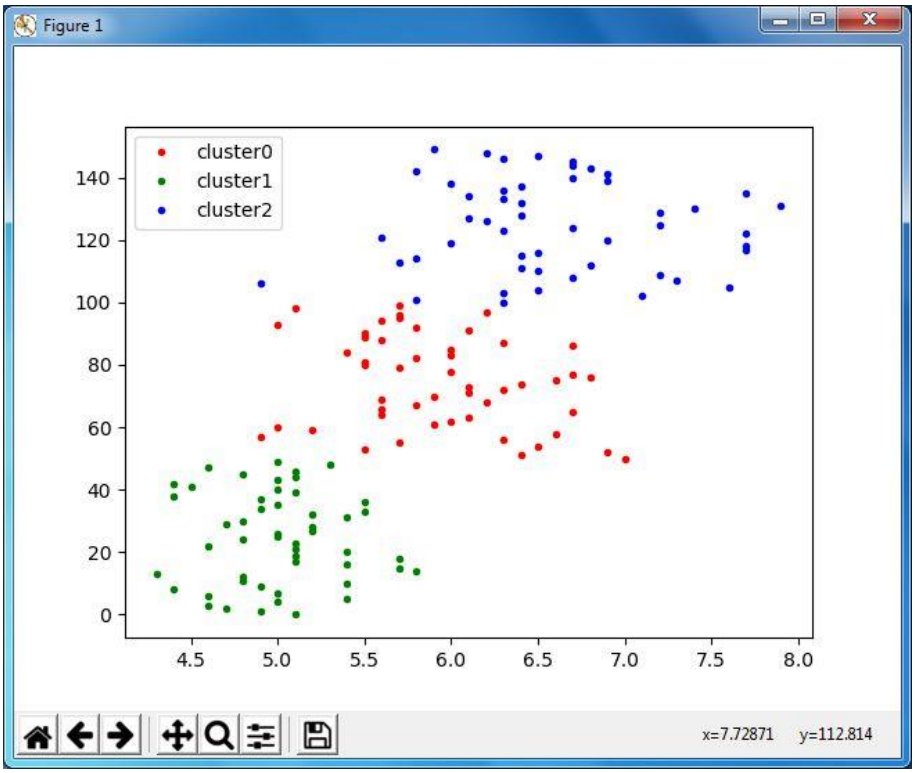

Figura 12 Gráfica de Atributo X Instancias

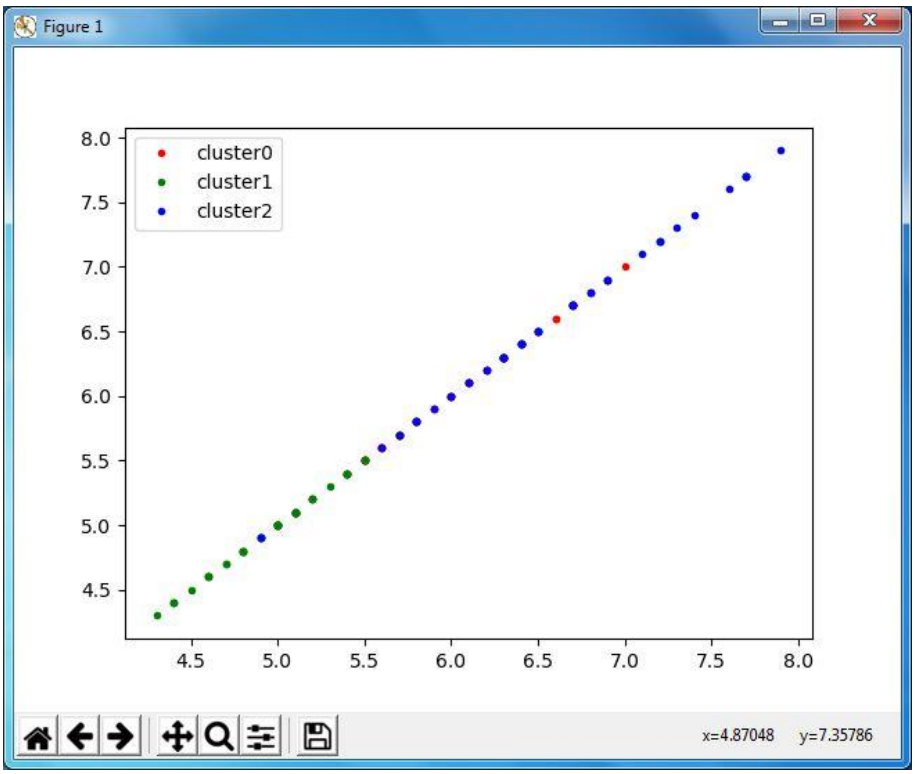

Figura 13 Gráfica Atributo X Atributo

En las Figuras 14, 15, 16 y 17 se muestran los resultados de la codificación de los bocetos de las interfaces gráficas, la cual pertenece a la tercera iteración.

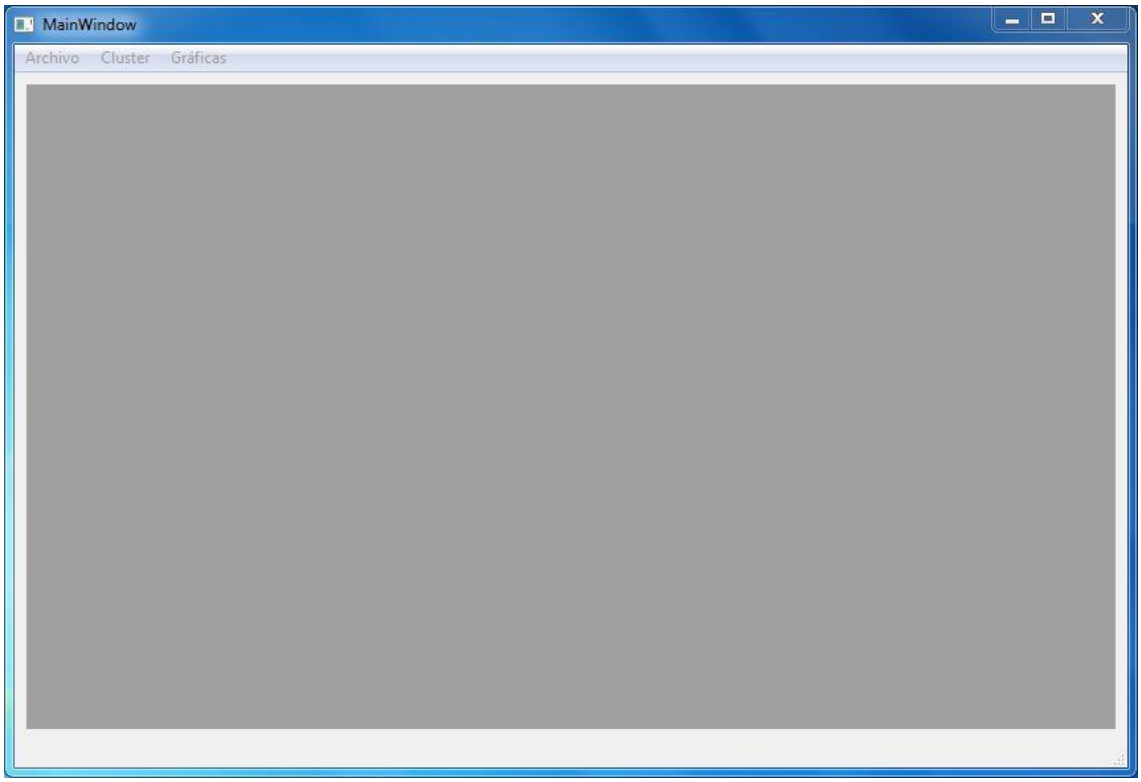

Figura 14 Codificación de la ventana principal

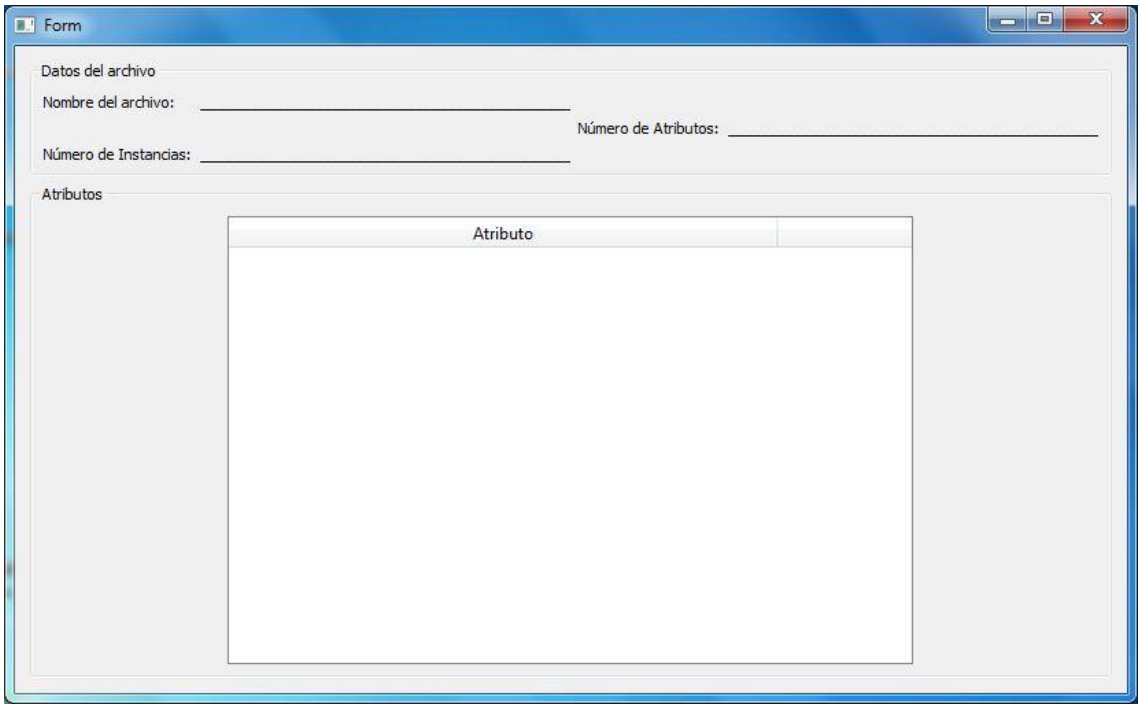

Figura 15 Codificación de la ventana Archivo

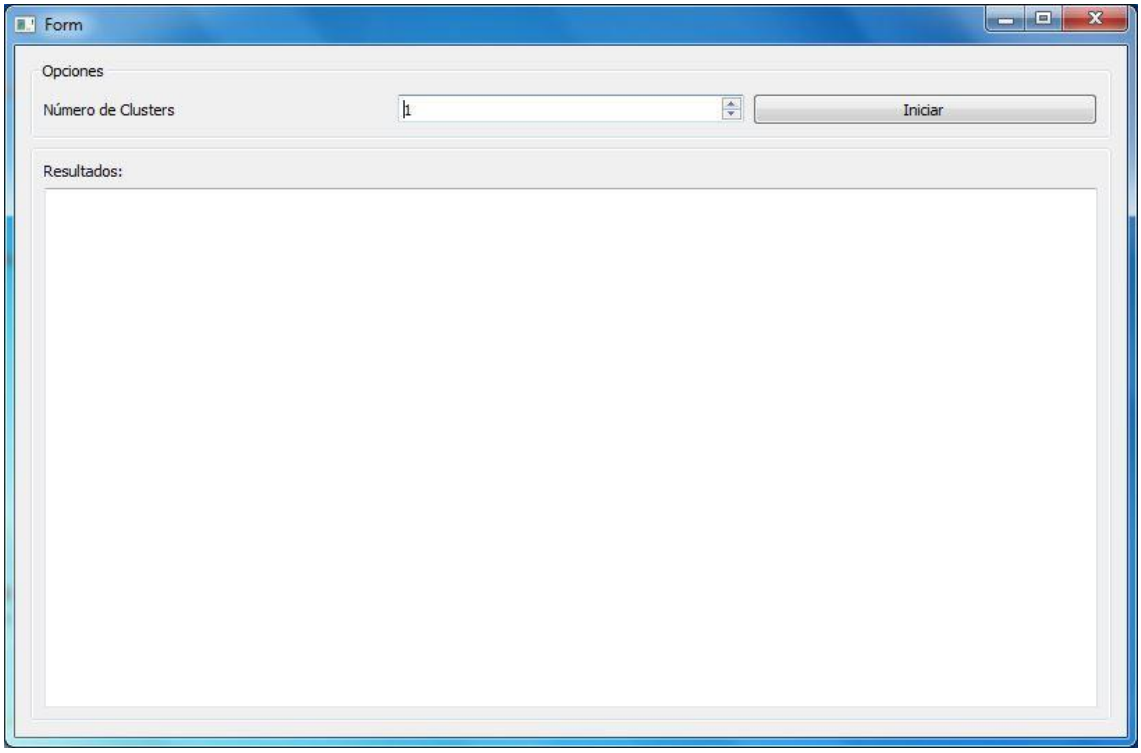

Figura 16 Codificación de la ventana Algoritmo

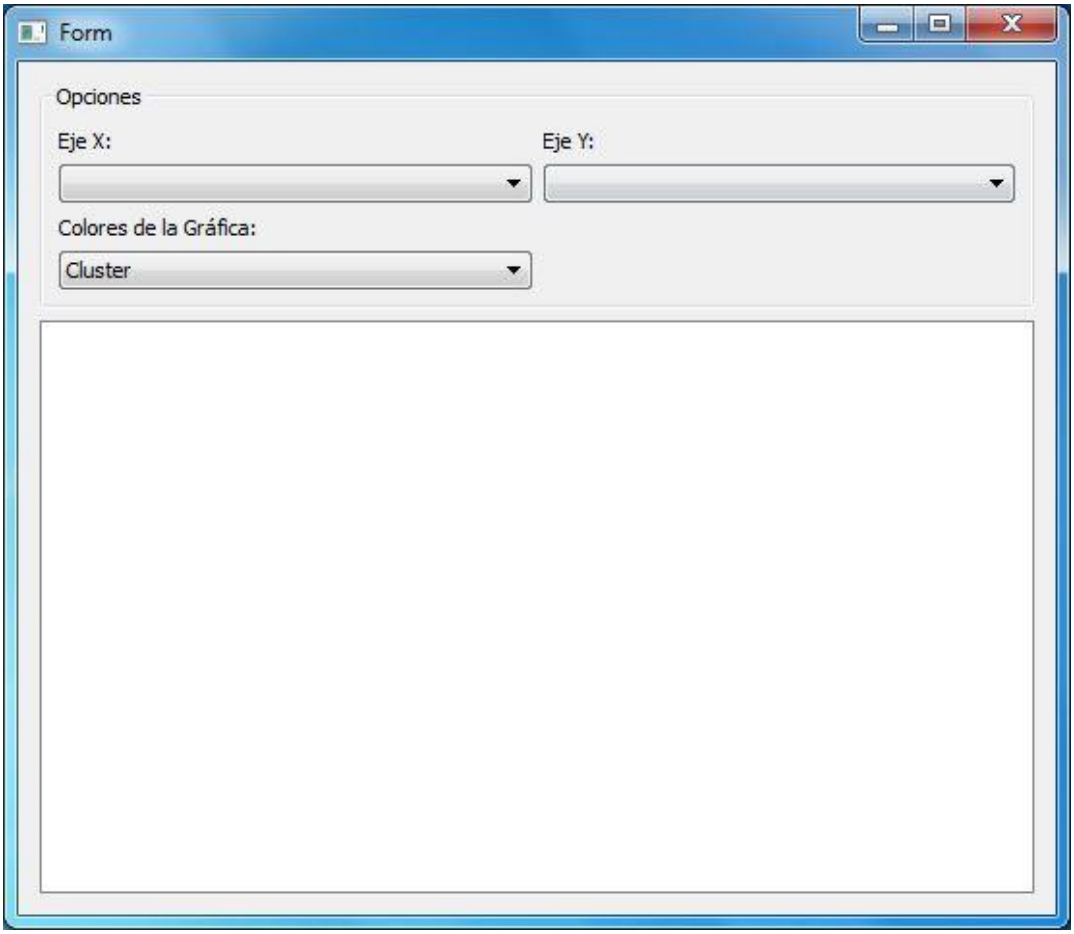

Figura 17 Codificación de la ventana Gráfica

A continuación se muestra la codificación de la cuarta iteración, la cual consiste en la codificación e integración del sistema completo, la ventana principal puede generar “n” sub ventanas del diseño anterior, como se puede ver en las figuras 18, 19, 20, 21 y 22.

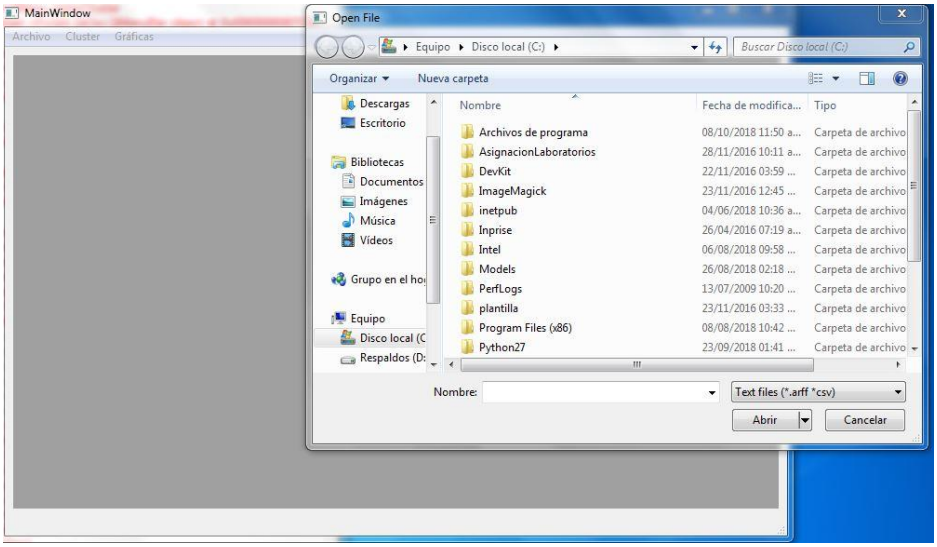

Figura 18 Ventana que permite seleccionar un archivo

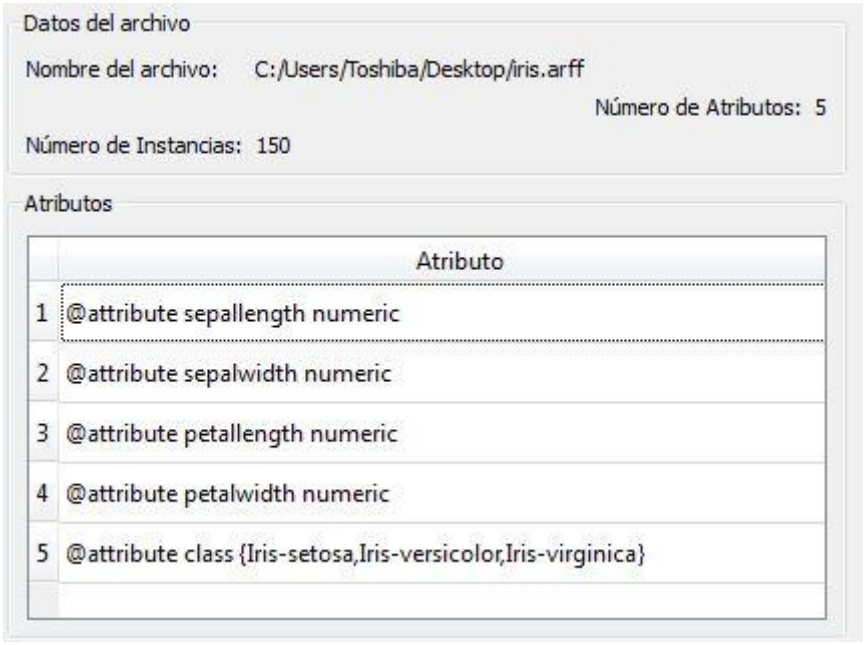

Figura 19 Codificación de la sub ventana Archivo

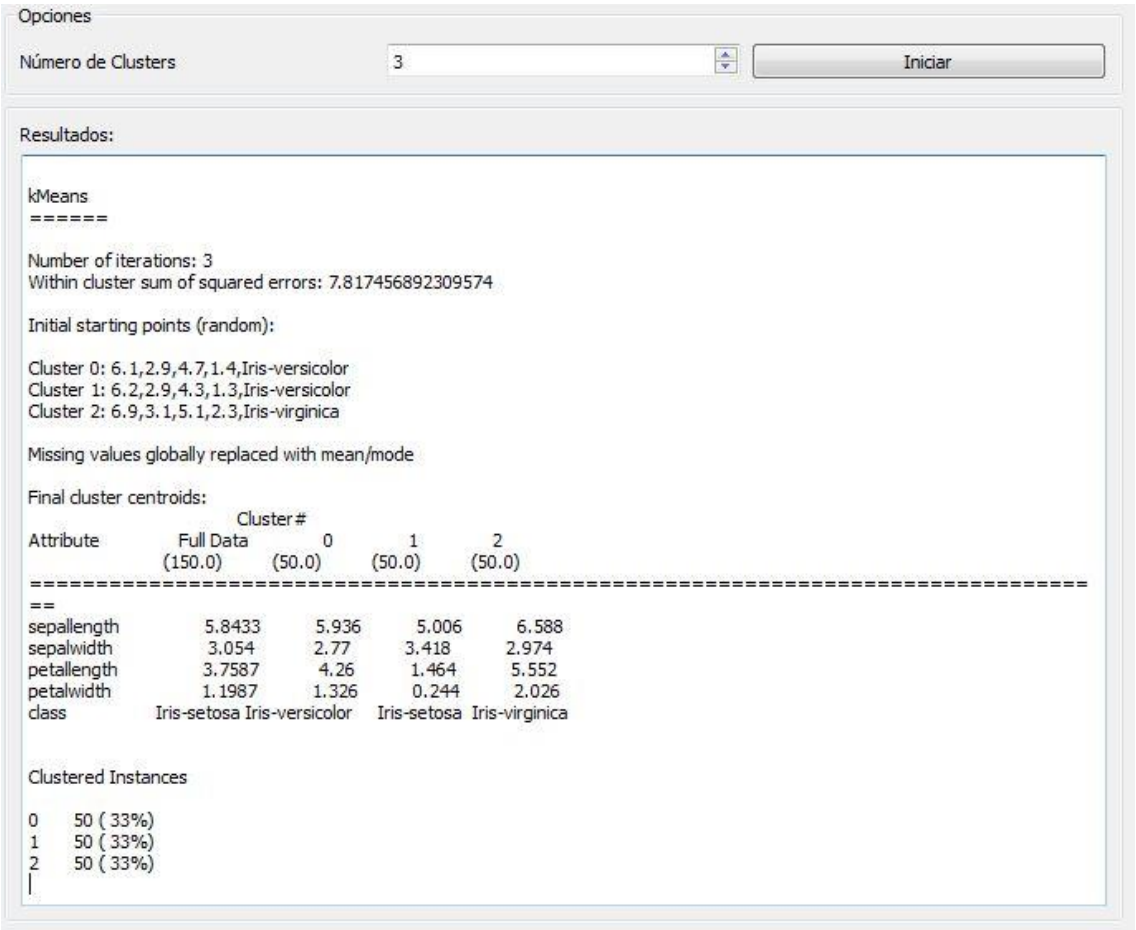

Figura 20 Codificación de la sub ventana Algoritmo

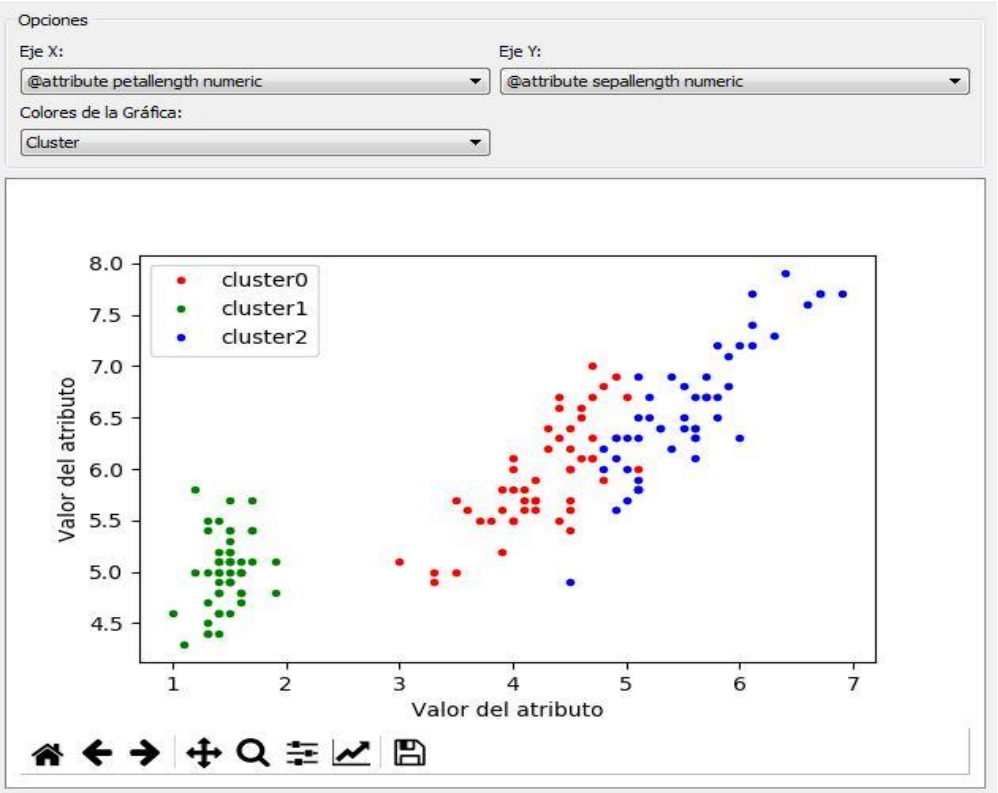

Figura 21 Codificación de la sub ventana Gráficas

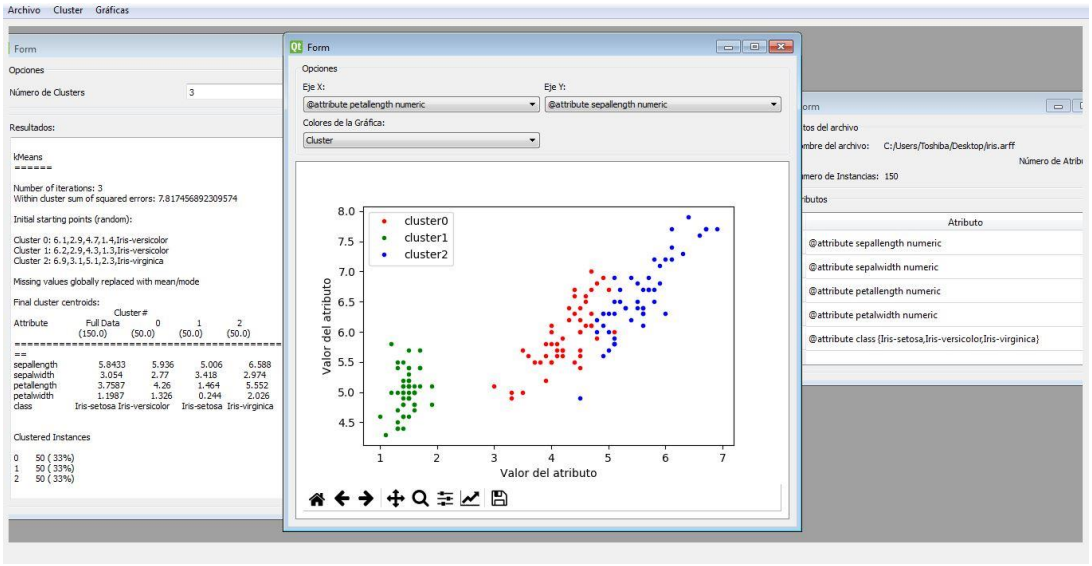

Figura 22 Vista del sistema actual

3.5. Despliegue

El sistema fue entregado al asesor técnico, el cual verificó si cumplía con todos los requerimientos establecidos y finaliza el prototipo.

4. Conclusiones y recomendaciones

Considero que el proyecto realizado me sirvió para conocer y tener una idea más clara sobre la minería de datos y al aprendizaje automático, dado a que son temas que en la ingeniería son difíciles de abarcar pero altamente demandados por las empresas actualmente, por que ofrecen una mejor proyección en la toma de decisiones.

En un principio existieron muchas dificultades para realizar el proyecto: empezando por el lenguaje de programación el cual no tenía la experiencia necesaria para realizar un sistema completo, además el manejo entre versiones es más notable y existen muchas incompatibilidades en librerías si no se cuenta con la versión adecuada y con respecto con las librerías de Weka solo contenían una pequeña parte a comparación de las librerías originales de Java. En este proyecto no sólo se tuvo que realizar la parte funcional del sistema, sino que también se investigó que se podía realizar, tomando en cuenta las incompatibilidades de algunas librerías y herramientas; algo muy importante en todo esto fue la investigación de los conceptos de minería de datos e inteligencia artificial, porque sería algo incongruente realizar un sistema sin saber que realmente hace o para qué sirve y que beneficios puede dar, por esa razón se tomó una parte del tiempo en conocer más sobre esta área el cual es muy extensa y complicada.

Para todas aquellas personas que deseen continuar con este proyecto o que trabajen con algo similar les recomiendo tener mucha paciencia y conocer el lenguaje en el cual trabajarán, además de investigar acerca de la minería de datos y de la inteligencia artificial.

Para concluir este proyecto me fue útil para conocer un nuevo lenguaje de programación, conocer dos áreas sumamente importantes actualmente, para tener un criterio más claro en la diferencia de los lenguajes de programación y para reafirmar conceptos vistos en clase acerca del desarrollo de software.

5. Referencias bibliográficas

[1] Roger S. Pressman. “Ingeniería del software un enfoque práctico” Editorial McGraw-Hill, Estados Unidos, 2010.

[2] URL: https://docs.python.org/3/faq/general.html Página oficial de la documentación de Python, en ella se puede consultar las preguntas frecuentes y el por qué utilizar este lenguaje de programación. Fecha de consulta: 10/octubre/2018

[3] Ian H. Witten. ”Data Mining” Editorial Elsevier, Estados Unidos, 2011

Universidad Politécnica de Puebla
Ingeniería en Informática

Obed Zeferino Ureiro Ruiz
Jorge De la Calleja Mora
Rebeca Rodríguez Huesca