**UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS**

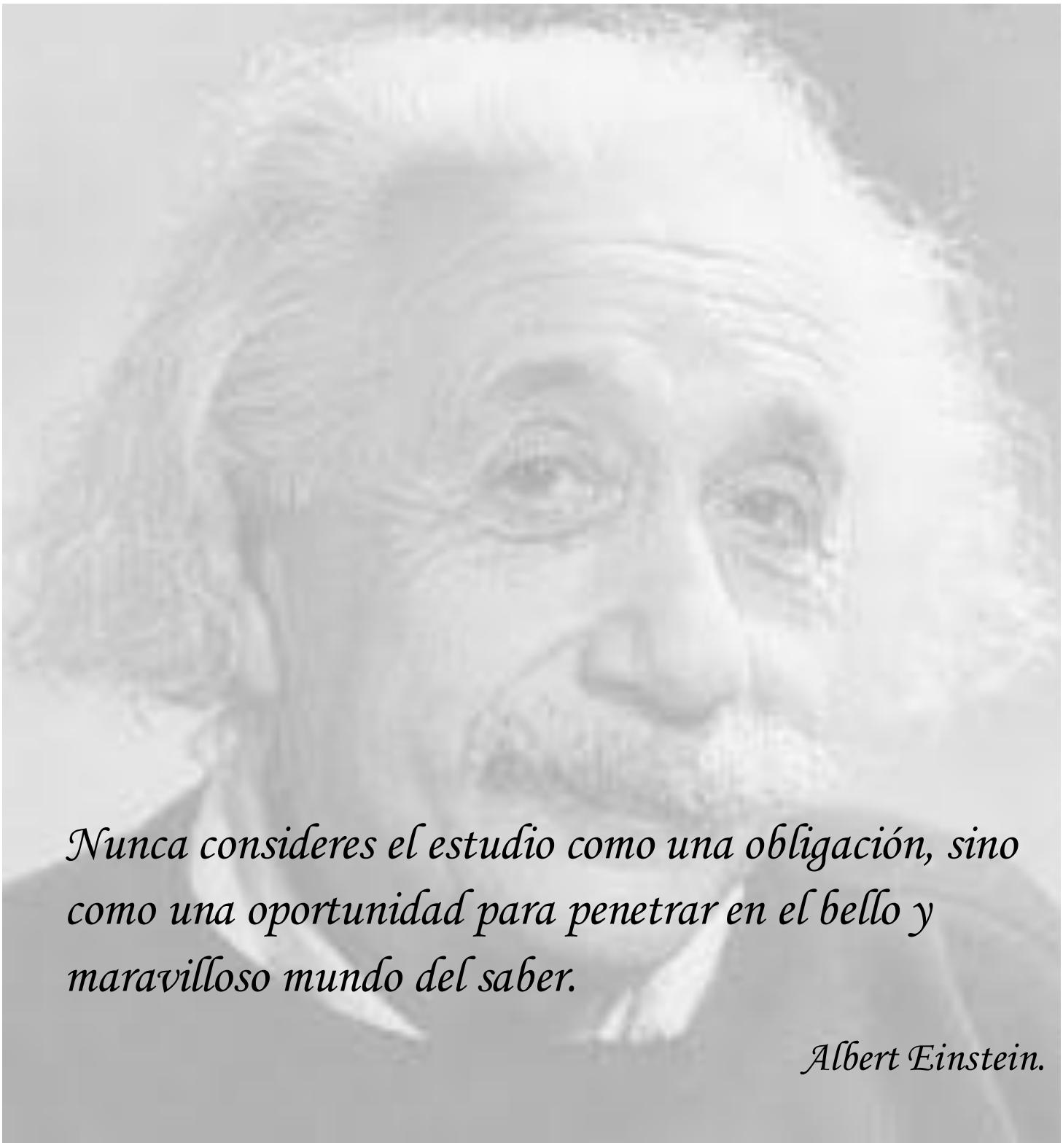
**FACULTAD 2**


**Trabajo de Diploma para optar por el título de**
**Ingeniero en Ciencias Informáticas**

# *Componente para la Validación de Algoritmos de Agrupamiento*

**Autor: Javier Sierra Ross**

**Tutor: MSc. Héctor Raúl González Diez**



*Nunca consideres el estudio como una obligación, sino como una oportunidad para penetrar en el bello y maravilloso mundo del saber.*

*Albert Einstein.*

# Agradecimientos

*Agradecer a mis padres y mi hermano quienes fueron un baluarte y los principales protagonistas de todo mi esfuerzo en estos cinco años de universidad.*

*Quisiera agradecer a mi novia, cuya ayuda y comprensión fue un consuelo para poder terminar esta tesis. Agradecer también a Osmel, quien ha sido mi amigo y fiel compañero en todo momento de mi vida.*

*Debo agradecer de manera especial y sincera al Profesor Héctor Raúl González Diez por aceptarme para realizar esta tesis. Su apoyo y confianza en mi trabajo y su capacidad para guiar mis ideas ha sido un aporte invaluable.*

*A mis familiares, abuelos, a mis primos, gracias por formar parte de mi vida y de alguna manera aportar su pedacito, para que yo pudiera graduarme.*

*A todos mis amigos, que aportaron de una forma u otra para que tuviera una buena estancia en la universidad.*

***Esta tesis esta dedicada a las tres personas más importante en mi vida, todo lo que tengo y todo lo que soy, se lo debo a ellos…***

***Mi madre…***

***Mi padre…***

***Mi hermano…***

Declaración de autoría

Declaro que soy el único autor de este trabajo y autorizo a la Universidad de las Ciencias Informáticas a que haga uso del mismo.
Para que así conste, firmo la presente a los ___ días del mes de_____ del año _____.

**Javier Sierra Ross**

_____

Firma del autor

**MSc. Héctor Raúl González Diez**

_____

Firma del tutor

## RESUMEN

En el ámbito del creciente desarrollo del clustering en innumerables áreas de la sociedad, y debido a la inherente imprecisión de la definición de clúster, surge el Trabajo de Diploma: “Componente para la Validación de Algoritmos de Agrupamiento” con el objetivo de ayudar a un mejor entendimiento de los aspectos fundamentales de esta área, creando una aplicación que concentre los diferentes enfoques que se proponen en el área de la validación de agrupamiento y así ayude en el diseño de experimentos, para recopilar información sobre diferentes tipos de datos. La solución ofrecerá a las personas calificadas en esta área, la posibilidad de interactuar con esta aplicación y experimentar las diferentes soluciones, obtenidas a través de las pruebas realizadas a diversos problemas, tanto de la vida real como experimentos de laboratorio.

# Abstract

In the area of increasing development of clustering in countless areas of society, and because of the inherent vagueness of the definition of cluster, Diploma work arises: "Validation Component for Clustering Algorithms" in order to help a better understanding of the fundamental aspects of this area, creating an application that concentrates the different approaches proposed in the area of clustering validation and so help in designing experiments, to collect information on different types of data. The solution offered to those qualified in this area, the ability to interact with this application and experience the different solutions obtained through tests with various problems, both real life and laboratory experiments.

# Índice

# INTRODUCCION

El desarrollo de Internet, aparejado con las mejoras en las conexiones, el aumento de ancho de banda y las nuevas formas de compresión de datos digitales, han hecho posible que sea uno de los medios de interacción más utilizado por la sociedad. Por ello es la plataforma común donde todos los usuarios se relacionan para transmitir información, siendo hoy el repositorio mundial de contenido, donde más allá de información textual, se maneja cualquier tipo de archivo, incluyendo las imágenes y el video.

La proliferación a nivel mundial de dispositivos, como computadoras y teléfonos inteligentes, aumentó el acceso a Internet dentro de los mercados emergentes y el incremento de datos generados por estos dispositivos. Este crecimiento ha contribuido a la duplicación del Universo Digital en los últimos años, hasta alcanzar un tamaño descomunal de 2,837exabytes en el año 2012. International Data Corporation (IDC) proyecta, que para el 2020, el Universo Digital alcanzará 40,026exabytes, cifra que supera las proyecciones anteriores por 14%. Este aumento de la cantidad de datos y su variedad, requiere serios avances en las técnicas de análisis de datos, con lo cual se ha impulsado en los últimos años la minería de datos, como campo de aplicación de la inteligencia artificial.(AETECNO 2013).

En la década de los noventa, disciplinas como la minería de datos, el aprendizaje automático, la minería de textos y el reconocimiento de patrones, resultaron de gran ayuda para extraer información relevante de un conjunto de datos. La minería de datos o DataMining consiste en la extracción de la información, o el conocimiento, que reside de forma implícita en los datos(HERNÁNDEZ 2004; KAMBER 2006). El proceso completo consiste en varios pasos, los cuales algunos autores los tratan de manera independiente.

- Obtención de los datos: El primer paso de un proceso de minería de datos consiste en obtener los datos a analizar.
- Limpieza e integración: Los datos se limpian, en la medida de lo posible, de datos inconsistentes y ruido. Asimismo, si los datos proceden de distintas fuentes se integran de forma adecuada.
- Selección y transformación de los datos: En este paso se seleccionan los datos más adecuados para la extracción del conocimiento y se transforman de forma apropiada. Por ejemplo, para

ciertos métodos de aprendizaje es necesario normalizar las características para que todas estén en un rango similar.

- Extracción del conocimiento: En este paso, es donde se aplican los métodos estadísticos de aprendizaje automático de reconocimiento de patrones capaces de extraer conocimiento de los datos. Estos métodos generalmente extraen modelos que representan los datos en forma de reglas, modelos estadísticos, redes neuronales.
- Validación del modelo: En este paso se valida el modelo obtenido en el paso anterior. Para ello es posible que haya sido necesario reservar parte de los datos. De esta manera se puede validar el modelo obtenido utilizando datos ignorados en el proceso de aprendizaje, de forma que se evitan posibles problemas derivados de un sesgo excesivo hacia la muestra de entrenamiento.
- Presentación de los resultados: En este paso final se presentan los resultados de forma sencilla de interpretar.

La etapa de extracción del conocimiento ha estado centrada en el grueso de los esfuerzos de la comunidad científica, en el cual se han desarrollado múltiples enfoques y algoritmos que están agrupados en el área del aprendizaje automático. El aprendizaje automático se ha tratado desde dos grandes enfoques. Los métodos de clasificación supervisada y los métodos de clasificación no supervisada. En la clasificación supervisada se requiere de un proceso de entrenamiento en el cual es necesario conocer a priori la clase a la que pertenecen cada uno de los datos del conjunto de entrenamiento y los rasgos que identifican a cada elemento del conjunto de entrenamiento. En el caso de la clasificación no supervisada, no existe información a priori de las clases a la que pertenecen los datos, por lo que se basan fundamentalmente en determinar una estructura óptima que agrupe los datos tomando en cuenta su naturaleza en el espacio de características (o rasgos).

Los algoritmos de agrupamiento son las técnicas más empleadas en la clasificación no supervisadas. Su objetivo es agrupar un conjunto de objetos de una base de datos, atendiendo a la naturaleza de los mismos (rasgos). Se emplea un modelo que permite dividir los datos de forma que los casos que estén en el mismo grupo sean lo más similares posibles y los que estén en distintos grupos se encuentren bien separados. El número de algoritmos de clustering que sirve para agrupar los datos de una base de datos es incalculable.(GONZÁLEZ Marzo de 2010; JAIN 2009b).

El agrupamiento de datos se ha utilizado principalmente para tres propósitos fundamentales:

- Detectar la estructura subyacente en los datos: conocer mejor los datos, generar hipótesis, detectar anomalías e identificar las características principales.
- Clasificación natural: identificar el nivel de similitud de diferentes organismos (relaciones filogenéticas).
- Compresión: método para organizar los datos y resumirlos a través de prototipos o representantes de clúster.

Derivado de estos objetivos, el agrupamiento se ha utilizado en numerosos campos o dominios de aplicación:

- En la biología, en el agrupamiento de genes(YEUNG 2001); clustering de proteínas.(PACCANARO 2006).(P. A. VIJAYA 2006).
- Seguridad Informática. (MALOOF 2006.). (PERONA 2009).
- En la medicina, con la segmentación de imágenes cerebrales mediante resonancia magnética (GOIKOETXEA 2010).
- Procesado de señal. (MURTAGH. 2009).
- Se aplica además en áreas representativas de la computación, en la Recuperación de Información y Minería de Datos.(JAJODIA 2002).

Como se puede ver, el desarrollo del agrupamiento se ha realizado gracias a investigadores de diversas disciplinas: taxónomos, sociólogos, psicólogos, biólogos, estadísticos, matemáticos, ingenieros, informáticos, médicos, entre otros. Dependiendo del área en la que se ha aplicado el clustering ha recibido diversos nombres como, Q-analysis, tipology, clumping o numerical taxonomy.

Existen diferentes bases de datos empleadas para realizar los experimentos con el fin de evaluar los algoritmos de agrupamiento. La elección de las base de datos, se realiza en función de parámetros muy variados, tales como las diferentes tallas de los conjuntos, la dimensión del vector de características, el número de posibles clases y el grado de solapamiento entre las regiones de clases distintas.

Debido a la diversidad de técnicas de agrupamiento, se pueden obtener diferentes soluciones al intentar dividir en grupos un conjunto de datos, por lo que es preciso decidir cuál de las estrategias de agrupamiento se ajusta más a este conjunto de datos. Para esto existen metodologías que permiten realizar esta evaluación a cada algoritmo de agrupamiento, verificando así, la calidad de las estructuras resultantes de los algoritmos de agrupamiento. Estas se conocen bajo el nombre de validación de clúster.(GONZÁLEZ Marzo de 2010).

Uno de los temas más importantes en el análisis de clúster, es la evaluación de los resultados del agrupamiento. Un método de validación de Clúster se refiere a aquellos criterios que nos sirven para la evaluación cuantitativa y objetiva del resultado que se obtiene de un algoritmo de agrupamiento.

La utilización de los métodos de validación de Clúster puede ser interesante, no solo como herramienta para determinar el mejor agrupamiento de la base de datos, sino también para determinar parámetros del algoritmo de agrupamiento realizado, como puede ser el numero óptimo de grupos para un conjunto de datos, o la mejor estructuración a partir de la aplicación iterativa de un algoritmo de agrupamiento.

Algunos métodos de validación de clúster basan su desarrollo en la comparación de dos estructuras de agrupamiento, es decir, se tiene una estructura que es conocida a priori (resultado ideal esperado para un conjunto de datos), y se asume que es un agrupamiento ideal, entonces se realiza una comparación con el agrupamiento que se deriva de un algoritmo de clúster y nos define si el algoritmo realizado es bueno o no [Criterio Externo]. Por su parte, el Criterio Interno es una complementación del Criterio Externo, ya que este último es más utilizado en entornos experimentales con bases de datos de prueba, debido a que el conocimiento de una partición a priori en datos reales resulta imposible conocerla. El criterio interno realiza su validación sobre una sola estructura de agrupamiento, derivada de un algoritmo de clúster, el cual puede tener diferentes parámetros de entrada.

Los índices de validación interna evalúan los resultados de un algoritmo de agrupamiento empleando dos conceptos o criterios de medida: cohesión y separación. La compacidad mide la cercanía entre los elementos de un mismo grupo. Mientras que la separación indica cuan diferentes son aquellos elementos que pertenecen a grupos diferentes. La figura 1 muestra gráficamente estos dos conceptos. (GONZÁLEZ Marzo de 2010).

a) b)

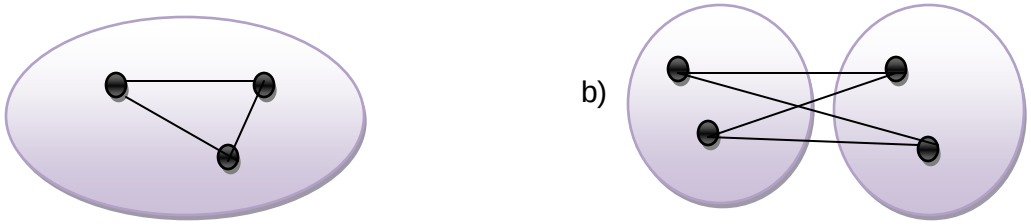


**Figura 1. Conceptos fundamentales en la validación de clúster. a). Cohesión b). Separación**

El análisis de los algoritmos de agrupamiento es una herramienta importante para investigar e interpretar datos, de ahí que los métodos de validación de clúster sirvan para evaluar la calidad de los agrupamientos, lo que permite identificar si se realizó, o no, un buen algoritmo a un conjunto de datos.

En este sentido, investigadores que trabajan en el dominio de las técnicas de agrupamiento, han desarrollado modelos y herramientas propias para dar solución a diversos problemas de agrupamiento. De conjunto con los métodos de agrupamiento se desarrollan metodologías, métricas y métodos para diseñar experimentos que permitan evaluar los resultados del agrupamiento. En la Universidad de las Ciencias Informáticas se desarrollan diversas investigaciones en este ámbito. Los resultados que se han obtenido se centran fundamentalmente en modelos y herramientas asociados a los métodos de agrupamiento. Estos resultados se obtienen de manera individual y aislada, en función de las diferentes aplicaciones y áreas de investigación que se desarrollan. Existen múltiples herramientas para el análisis de datos que implementan algoritmos de agrupamiento como es el caso de WEKA y Rapidminer. Estas herramientas son de propósito general, por lo que su uso en problemas específicos requiere que los investigadores implementen adecuaciones particulares a estas. Por tanto, la integración y el diseño experimental de aplicaciones específicas, requiere definir una plataforma experimental adaptada al problema específico que se desea tratar. Uno de los problemas específicos que se aborda en la universidad es el de la categorización de imágenes, utilizando un modelo de representación basado en algoritmos de agrupamiento, por lo que el diseño de experimento para evaluar los resultados del agrupamiento es crucial en el desarrollo de esta investigación.

Por la situación problemática anteriormente descrita, se tiene como **problema a resolver**,

¿Cómo facilitar a investigadores de la UCI, el diseño de diversos tipos de experimentos para la validación de algoritmos de agrupamiento?

El **objeto de estudio** de la presente investigación lo constituyen los métodos de validación de clúster.

De ahí que el **objetivo general** del presente trabajo sea desarrollar un componente de software que contenga un paquete de métricas y métodos, que ayude en el diseño de diferentes experimentos para la validación de algoritmos de agrupamiento.

Del objetivo general se derivan los siguientes **objetivos específicos**:

Caracterizar los diferentes enfoques para la validación de clúster, así como su aplicación en la evaluación del agrupamiento.

Definir un componente que implemente los diferentes enfoques para la validación de clúster, de modo que se pueda reutilizar en la evaluación de algoritmos de agrupamiento.

Validar el componente propuesto, así como los diferentes enfoques de validación de clúster desarrollados como parte del componente.

El **campo de acción** viene dado por el análisis de los métodos de validación de clúster como herramienta para validar la partición resultante de un algoritmo de agrupamiento.

La **idea a defender** es que si se desarrolla un componente de software, que contenga un paquete de métricas y métodos, que ayude en el diseño de diferentes experimentos para la validación de algoritmos de agrupamiento, se le facilitará el trabajo a los investigadores de la UCI, el diseño de estos experimentos para la validación de algoritmos de agrupamiento.

**Aporte práctico de la investigación.**

Se desarrolla un componente de software para la validación de algoritmos de agrupamiento que sirva de base experimental a los métodos de representación de imágenes basado en algoritmos de agrupamientos.

Durante el trayecto de la investigación del siguiente trabajo se utilizarán **métodos científicos** para lograr caracterizar a fondo el objeto de estudio, así como garantizar el conocimiento del estado del arte, su evolución y relación con otros fenómenos. Los métodos científicos a utilizar son:

**Analítico-Sintético:** a través de este método se analiza la bibliografía disponible para realizar una investigación lo más completa posible sobre los métodos de validación de clúster. Este método permite definir los principales conceptos, definiciones y otras soluciones ya existentes.

**Histórico-Lógico:** A través de este método se hace un estudio de la evolución de los métodos de validación de clúster, para así lograr un mejor entendimiento de estos enfoques y trabajar en su mejoramiento o actualización.

**Métodos empíricos:** se realizan experimentos a los algoritmos de agrupamiento usando los métodos de validación de clúster.

**Organización de la Tesis:**

A partir de los objetivos definidos en esta sección, la tesis se ha organizado en tres capítulos fundamentales.

**Capítulo 1:** La primera parte se dedica a la presentación de los fundamentos teóricos sobre los que se basará gran parte de la tesis. Se abordarán los conceptos asociados a los métodos de validación de clúster, así como una descripción de cada uno de estos enfoques.

**Capítulo 2:** Se hace referencia a las características que debe tener el sistema para su buen funcionamiento, haciendo hincapié fundamentalmente en los requisitos de software y las fases en las que está dividida la metodología utilizada, además del diseño de clases propuesto.

**Capítulo 3:** En este capítulo se realiza la propuesta de un Caso de Estudio, donde se haga uso de los métodos de validación de clúster, como enfoques fundamentales para validar los algoritmos de agrupamiento. Además de las pruebas para validar el componente.

# CAPITULO 1. Fundamentación Teórica.

## Introducción:

En este capítulo se abordarán los conceptos asociados a los métodos de validación de clúster, así como una descripción de cada uno de estos enfoques que se van a utilizar para darle solución al trabajo. Además se explicarán las metodologías y herramientas con las que se va a trabajar, justificando en cada caso su utilización.

## 1.1 Definiciones y notación

Antes de entrar en una descripción más detallada de algunos elementos, se definirán varios conceptos y notaciones que se emplean a lo largo del documento.

El aprendizaje no supervisado es muy importante cuando tenemos muestras sin etiquetas de clase, cuando el costo de etiquetarlas por un experto es alto, o cuando los patrones pueden variar con el tiempo, por lo que es necesario primero procesar los datos para luego clasificar. La principal ventaja que presenta la clasificación no supervisada es que se puede obtener un conjunto de entrenamiento empleando muestras no etiquetadas. Como se explicó con anterioridad los métodos más conocidos en la clasificación no supervisada son los algoritmos de agrupamientos los cuales serán tratados con mayor profundidad en la presente investigación.

En primer lugar es necesario definir las notaciones empleadas en la presente investigación relacionadas con el conjunto de datos que sirven como entrada al algoritmo de agrupamiento.

**Definición 1:** Una base de datos, $X = \{x_1, x_2, \dots x_n\} \subseteq \mathrm{X} \in \mathfrak{N}^J$, es un conjunto de $N$ objetos o puntos representados como vectores de características en un espacio $J - dimensional$. Denotamos el valor del atributo correspondiente a la dimensión $J$ del vector $Xi$ como $xij$.

Como elemento clave del presente trabajo, se debe definir de manera formal el resultado que se obtiene de la aplicación de un algoritmo de agrupamiento, el cual será tratado con el término de clúster.

**Definición 2:** Una partición, $C = \{C_1, C_2, \dots C_k\} \subseteq X$, resultado de un algoritmo de agrupamiento asigna cada elemento del conjunto de datos de entrada $X$ al clúster $C_i$. El resultado del algoritmo de agrupamiento cumple las siguientes propiedades: $\bigcup_{C_k \in P} C_k = X$, $C_k \cap C_l = \emptyset \; \forall \; k \neq l$.

**Definición 3:** Datos Continuos: Son los datos que pueden tomar cualquier valor, estos datos pueden tener tantos decimales como se desee y que entre cada dos de ellos, siempre puede haber otro. Al poder estar muy cerca unos de otros, no se pueden estudiar de uno en uno y se agrupan en intervalos. Los valores que asume son números reales.

Para determinar las relaciones entre los objetos en los métodos de agrupamiento, se emplea lo que se conoce por función de distancia, cuando se trabaja con datos continuos. Podemos usar la función distancia para definir las relaciones entre los datos del conjunto de entrada. (Espacios métricos 2011).

**Definición 4:** Distancias: La distancia entre dos objetos cualesquiera de un agrupamiento es la medida de similitud entre estos objetos de acuerdo a sus características, estas se escogen de forma tal que mientras más parecidos sean los objetos, menor debe ser la distancia entre ellos y viceversa.

En aplicaciones prácticas de algoritmos de agrupamiento, varias cuestiones deben ser resueltas, entre ellas, la determinación del número de clúster y la evaluación de la calidad de un clúster, para esto existen algunos índices de validación de clúster que proporcionan una medida objetivo del resultado de un algoritmo de agrupamiento. En esta investigación se tratarán los índices de validación de clúster desde el enfoque del análisis de la calidad de una partición.

**Definición 5:** Índices de validación de clúster: Un índice de validación de clúster es una función, $F(P)$, que mide la bondad o calidad de una partición. Salvo si expresamente se indica lo contrario, asumiremos que un valor menor de $F$ denota una partición mejor.

## 1.2 Tipos de Algoritmos de Agrupamiento

Debido al gran número de algoritmos de agrupamiento y a las diversas estrategias seguidas a la hora de agrupar los datos, no es sencillo realizar una clasificación clara. La clasificación tradicional de los algoritmos los divide en dos categorías principales: jerárquicos y particionales.

Los algoritmos jerárquicos construyen una serie de particiones anidadas de forma aglomerativa y divisiva. Una partición anidada de forma aglomerativa se refiere a aquellos clúster que se van uniendo

formando particiones con cada vez menos clúster. En el segundo caso o sea una partición divisiva es el caso en que los clúster se van dividiendo de forma que las particiones tienen cada vez más clúster.

La ventaja principal de este tipo de algoritmo es que es capaz de capturar la posible estructura jerárquica de los datos. La parte izquierda de la Figura 2 muestra unos datos con claro carácter jerárquico, donde una solución de 3 clúster es tan buena como una de 6. Un algoritmo jerárquico puede crear una jerarquía de clúster donde se representen los distintos niveles de detalle en los que podemos estructurar los datos.

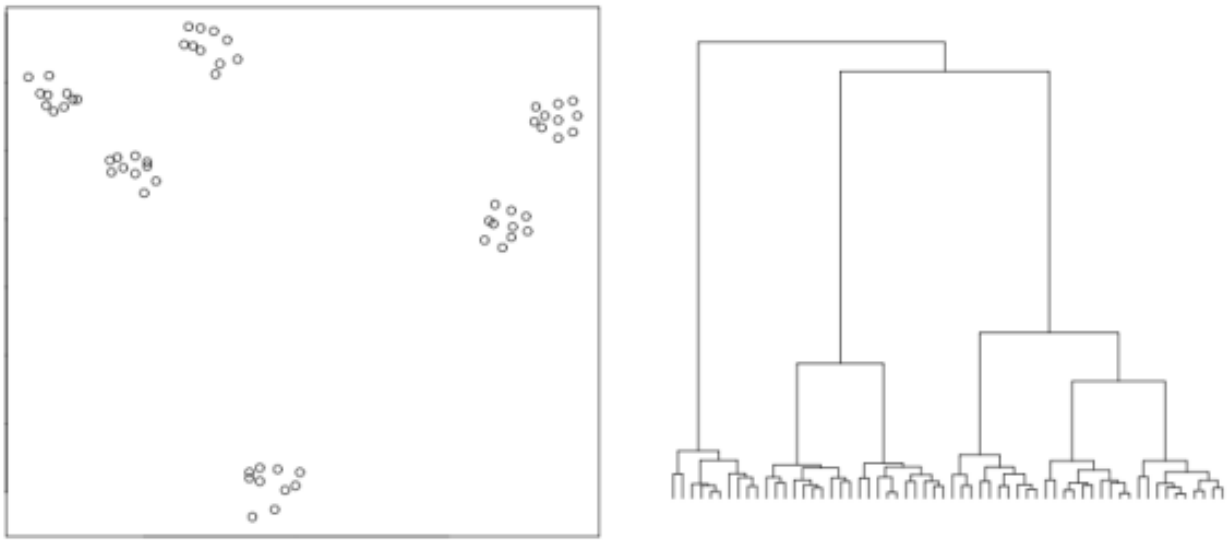


**Figura 2: Una base de datos con estructura jerárquica y un posible dendrograma.**

Típicamente una jerarquía de clúster se representa mediante un árbol denominado dendrograma. Los nodos hoja, representan la primera partición de un proceso aglomerativo (o la última de uno divisivo), mientras que los nodos internos representan la unión de varios clúster. La parte derecha de la Figura 2 muestra el dendrograma correspondiente a una posible jerarquía de los datos de la parte izquierda.

A diferencia del algoritmo jerárquico, el particional genera una única partición de los datos con lo que en un caso como el de la Figura 2 deberá optar por un único nivel de detalle. Generalmente este tipo de algoritmo es menos costoso y más utilizado.

### 1.2.1 Algoritmos de Agrupamiento

A continuación se describen dos algoritmos de agrupamiento que son los más usados en este tema. El primero de ellos es un método general en el que se basan los algoritmos jerárquicos más utilizados. El segundo es el algoritmo de clustering por excelencia: K-means.

#### 1.2.1.1 Método jerárquico

Muchos de los algoritmos jerárquicos más utilizados como el single-linkage, average-linkage, complete-linkage o Ward se basan en un mismo método. Este es el método jerárquico aglomerativo por excelencia.

Este método es sencillo e intuitivo y se describe en el Algoritmo 1. En primer lugar se crea una partición donde todos los clúster están compuestos por un único caso. Posteriormente, y de forma iterativa, se van uniendo los dos clúster más cercanos hasta que la partición obtenida tenga un único clúster, con todos los casos de la base de datos. Como consecuencia se crea una jerarquía con tantas particiones como casos, donde en cada partición se unen exactamente dos clúster de la partición anterior. Por lo tanto, el dendrograma resultante es un árbol binario.

| Algoritmo 1 Clúster Jerárquico |
|---|
| 1. Calcular la matriz de distancia correspondiente a los datos<br>2. Crear un clúster con cada caso<br>3. ***for*** *i*= 2 ***to*** N ***do***<br>4. Unir los dos clúster más cercanos en un nuevo clúster<br>5. Actualizar la matriz de distancias<br>6. ***end for*** |

Existen diferentes medidas utilizadas en el clustering para calcular la distancia o proximidad entre clúster y dependiendo de la medida utilizada este método permite obtener distintos algoritmos. Por ejemplo, el algoritmo single-linkage considera que la distancia entre clúster es la distancia entre los objetos más cercanos de cada clúster. Sin embargo el complete-linkage define la distancia entre los clúster como la distancia entre sus dos casos más lejanos.

Aparte de estas dos, existen otras muchas medidas de proximidad entre clúster, y por lo tanto, algoritmos basados en Clúster Jerárquico. Uno de ellos es el algoritmo Ward que, al igual que el algoritmo K-means, trata de minimizar el error cuadrático medio. Otros cuatro algoritmos comunes denominados UPGMA (o average-linkage), WPGMA, UPGMC y WPGMC (DUBES. 1988; SOKAL 1973)surgen de la combinación de dos criterios. Elprefijo "U" (Unweighted) significa que todos los casos pesan igual, mientras que "W" (Weighted) significa que todos los clúster pesan igual, de forma que los casos en los clúster pequeños pesan más. El sufijo "A" (Average) significa que la distancia se calcula mediante la media entre todos los casos de cada clúster, mientras que "C" (Centroid) significa que la distancia entre clúster se define como la distancia entre centroides. "PGM" es común a los cuatro y significa Pair Group Method.

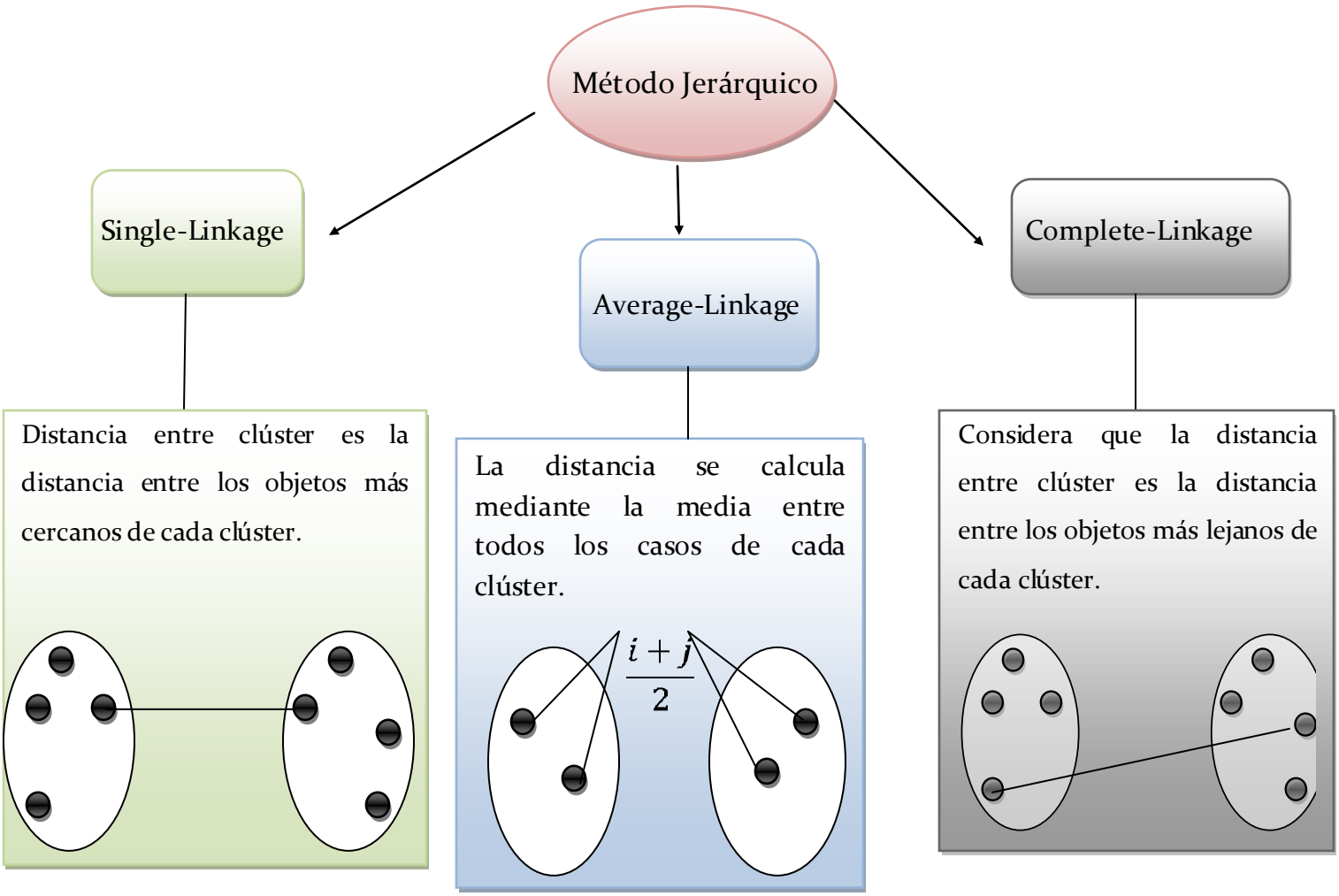



**Figura 3. Medidas para calcular la distancia o proximidad entre clúster.**

El enfoque de agrupamiento jerárquico se apoya de una matriz de distancia que establece las relaciones entre los vectores de características que componen la base de datos. Esta matriz de distancia cuenta con $\frac{N\,(N-1)}{2}$ elementos.

Para otras medidas de proximidad se pueden definir otro tipo de fórmulas matemáticas. Sin embargo (WILLIAMS. 1967), Lance y Williams propusieron una fórmula con cuatro coeficientes que sirven para actualizar la matriz de distancias y se adecúa a la mayor parte de medidas de proximidad conocidas. Todas estas funciones de cercanía, que emplean los métodos jerárquicos anteriormente mencionados, se pueden ver como una selección de parámetros en la fórmula que se muestra a continuación.

Supongamos que hemos unido los clúster $Ci\ y\ Cj$ para crear el nuevo clúster $Cij$. Entonces la distancia entre el nuevo clúster $Cij$ y un clúster $C_k$se puede calcular como:

$$d(Cij, C_k) = \alpha_i d(C_k, Ci) + \alpha_j d(C_k, Cj) + \beta\, d(Ci, Cj) + \square\, |\, d(C_k, Ci) - d(C_k, Cj)\, |$$

Donde $d$ es la función de proximidad entre los clúster.

Esta es una forma de simplificar los métodos combinatorios en una única fórmula, pero también permite definir con sencillez, nuevas medidas de proximidad. Además, permite utilizar técnicas de optimización para instanciar los cuatro coeficientes de forma que se pueda definir una medida de proximidad adaptada a cada problema.

En la siguiente tabla se indican los valores de los coeficientes para las medidas de proximidad mencionadas anteriormente.

| Algoritmo | $\alpha_i$ | $\alpha_j$ | $\beta$ | $\square$ |
|---|---|---|---|---|
| Single-Linkage | $1/2$ | $1/2$ | $0$ | $-1/2$ |
| Complete-Linkage | $1/2$ | $1/2$ | $0$ | $1/2$ |
| UPGMA | $\frac{\lvert Ci\rvert}{\lvert Cij\rvert}$ | $\frac{\lvert Ci\rvert}{\lvert Cij\rvert}$ | $0$ | $0$ |
| WPGMA | $1/2$ | $1/2$ | $0$ | $0$ |
| UPGMC | $\frac{\lvert Ci\rvert}{\lvert Cij\rvert}$ | $\frac{\lvert Ci\rvert}{\lvert Cij\rvert}$ | $\frac{-\lvert C_i\rvert\lvert C_j\rvert}{\lvert Cij\rvert^2}$ | $0$ |

| | | | | |
|---|---|---|---|---|
| WPGMC | $1/2$ | $1/2$ | $-1/4$ | 0 |
| Ward | $\frac{\lvert C_i\rvert + \lvert C_k\rvert}{\lvert Cij\rvert + C_k}$ | $\frac{\lvert C_j\rvert + \lvert C_k\rvert}{\lvert Cij\rvert + C_k}$ | $\frac{-\lvert C_k\rvert}{\lvert Cij\rvert + C_k}$ | 0 |

### 1.2.1.2 Métodos particionales. K-means

Otro de los algoritmos de agrupamiento más utilizados es el K-means, es conocido desde hace más de cinco décadas, y sigue siendo uno de los algoritmos de aprendizaje automático mejor valorados.(KUO-LUNG WU Marzo, 2009). Dado un conjunto de datos y un valor$K$, el algoritmo genera una partición de los datos con $K$ clúster. El objetivo de K-means es minimizar el error cuadrático medio de todos los clúster. El error cuadrático de un clúster, se define como la media de los cuadrados de las diferencias de los objetos de un clúster al centroide del clúster. El centroide de un clúster es el punto medio de todos los objetos del clúster.

El Algoritmo 2 describe los pasos seguidos por K-means. En primer lugar se seleccionan $K$ casos como centroides iniciales. Posteriormente, en un segundo paso, se asigna cada caso al clúster, representado por el centroide más cercano. Finalmente, en el tercer paso, se vuelven a calcular los centroides como la media de todos los casos asignados a cada clúster. Si los centroides calculados difieren de los anteriores, se vuelve al paso 2, si no, el algoritmo devuelve como resultado la última partición calculada.

| Algoritmo 2 K-means |
|---|
| 1. Seleccionar $K$ casos como centroides iniciales<br>2. ***repeat***<br>3. Asignar cada caso al clúster con centroide más cercano<br>4. Recalcular los centroides de cada clúster<br>5. ***until*** los centroides no se han modificado |

El algoritmo K-means requiere tres parámetros de usuario. El primero y el principal es $K$, el número de clúster de la partición.Otro parámetro a definir es la selección inicial de centroides. El tercer parámetro a definir es la función de distancia entre los casos de la base de datos.

## 1.3 Retos de los algoritmos de agrupamiento

El clustering ha demostrado ser un problema de difícil solución. Aunque otras áreas del aprendizaje automático, la estadística o la minería de datos están teóricamente bien analizadas y se basan en sólidos y claros análisis matemáticos, no ocurre lo mismo con el agrupamiento(KETTENRING 2006). Jain (2009) atribuye la dificultad del agrupamiento a la inherente imprecisión de la definición de clúster y a la dificultad de definir funciones de distancia y funciones objetivo adecuadas.

Como consecuencia, existen multitud de retos en el área del agrupamiento, como por ejemplo el autoescalado de las variables, el agrupamiento semisupervisado, el agrupamiento múltiple, el agrupamiento de variables, la reducción de dimensiones, el agrupamiento de datos heterogéneos, la extracción de particiones a partir de jerarquías, el agrupamiento de bases de datos de gran tamaño, la validación e interpretación de los resultados. (JAIN 2009a; KETTENRING 2006).

A pesar de que el agrupamiento lleva años de estudio y gran cantidad de algoritmos diseñados, existen cuestiones básicas del agrupamiento que siguen sin tener una solución satisfactoria.

## 1.4 Validación de Clúster

El objetivo de la validación del agrupamiento, es medir de forma objetiva la calidad de la partición obtenida como resultado de un algoritmo de agrupamiento, este resultado puede ser una partición o una jerarquía de particiones. Otra campo de aplicación de los índices de validación de clúster está en determinar el número correcto de grupos sobre un conjunto de datos. Aunque este enfoque no fue tratado en la presente investigación, la necesidad de algunos métodos de agrupamiento de determinar el número de grupo ha permitido el desarrollo de los índices de validación de clúster.

La validación de clúster se ha clasificado tradicionalmente en tres categorías: externa, interna y relativa. Sin embargo, aunque las dos primeras categorías parecen estar claramente definidas, existe cierta confusión respecto a la tercera.(DUBES. 1988; MARIA HALKIDI 2001).(K. Y. YEUNG 2001)

La validación externa asume el conocimiento de la partición correcta de los datos. Es decir, la verdadera partición es conocida. Evidentemente, en un contexto real de aplicación de agrupamiento, la partición correcta no es conocida, pero puede que en un contexto experimental de evaluación de algoritmos de agrupamiento sí lo sea. El problema principal de esta categoría es que no siempre es

posible conocer de antemano una partición ideal sobre todo cuando se trabaja con datos reales.(DOM 2001).

En cambio la validación interna, no necesita el conocimiento de la partición correcta, para realizar la evaluación. Ya que esta realiza su validación sobre la partición que resulta de un algoritmo de agrupamiento, esta evalúa la partición en base a los datos y las distancias entre ellos. La ventaja principal es que no requiere un conocimiento priori de la partición correcta, lo que permite que se utilice en aplicaciones reales y no sólo en experimentos de laboratorio. Tiene como desventaja que evalúa la partición sobre los mismos datos que se utilizaron para construirla. Desafortunadamente, evaluar la adecuación de una partición sobre los datos de la misma partición evaluada, es igual a evaluar los clúster bajo una diferente función objetivo. Por ejemplo, si utilizamos como medida de calidad de una partición el error cuadrático medio, es de suponer que un algoritmo como K-means obtenga mejores resultados que el single-linkage.

La validación tanto externa como interna, se realiza a través de índices de validación, que dada una partición, calculan una medida de la calidad de dicha partición.

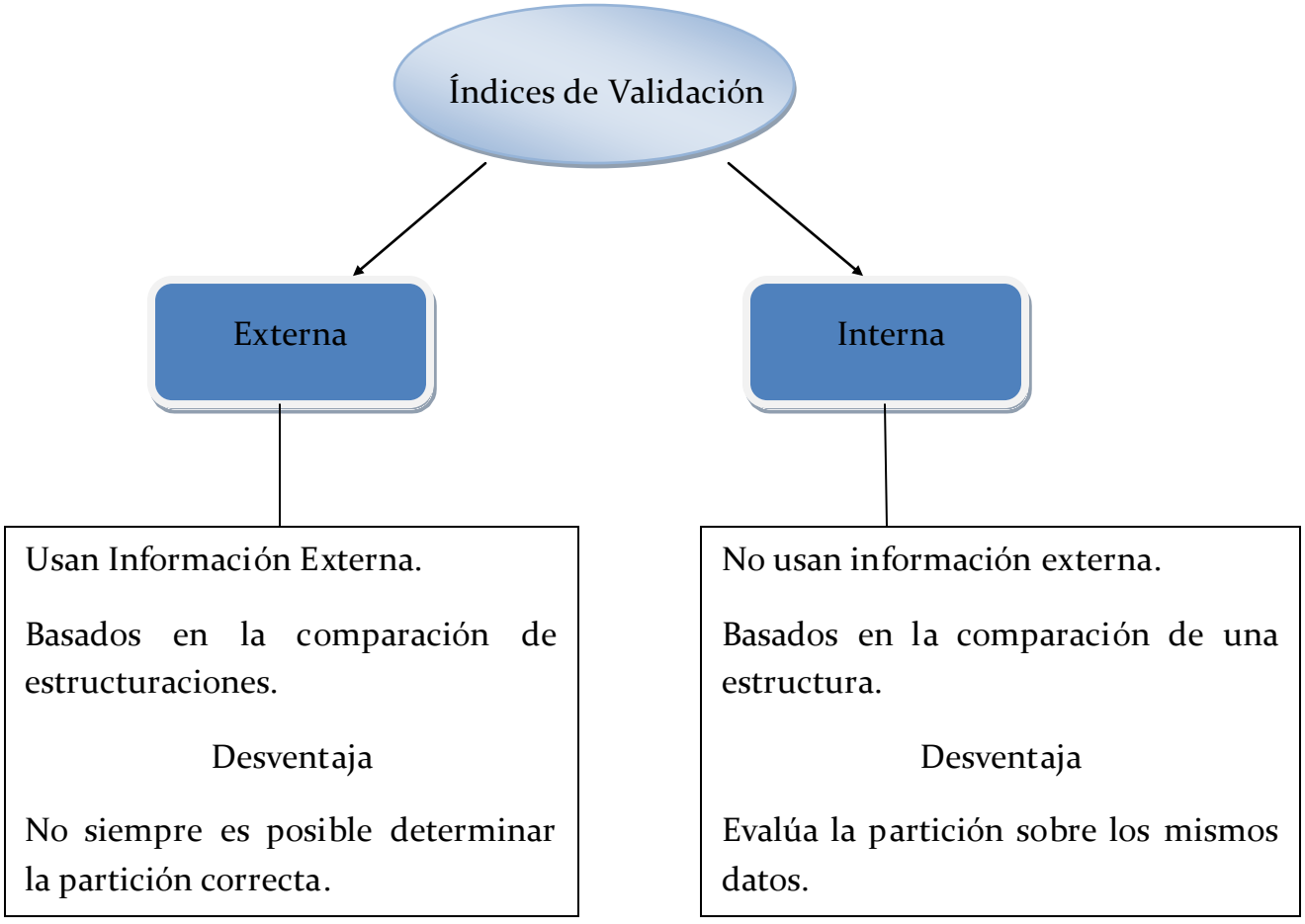



**Figura 4. Diferentes tipos de índices de validación y características generales.**

En este trabajo se realiza el estudio de solamente dos validaciones, la externa y la interna, debido a que el término de validación relativa no parece estar claramente definido. Jain y Dubes (1988) lo utilizan para la validación mediante la comparación de varias particiones, pero solo mediante índices de validación interna. En cambio, Halkidi et al. (2001) limitan la validación relativa a la comparación de particiones obtenidas mediante la variación de los parámetros de un mismo algoritmo. Otros, como Dom (2001) o Yeung et al. (2001), ni siquiera mencionan la validación relativa.

Dado que la práctica generalizada es la de analizar las particiones de manera comparativa, tanto para la validación externa como la interna, parece que la confusión puede surgir al situar la validación relativa al mismo nivel que las otras dos. Parece más adecuado limitar la validación a dos categorías: externa e interna.

### 1.4.1Validación de Particiones

La validación de particiones es la validación más común en el agrupamiento. Para ello existen varios índices que tratan de estimar la calidad de la partición evaluada. Una forma común para evaluar una partición es convertirla en una matriz,$M$, donde $M(i,j) = 1$ si los pares $Xi\ y\ Xj$ están en el mismo clúster y $M(i,j) = 0$ en caso contrario. De este modo se pueden utilizar medidas de similitud de matrices como índices de validación de particiones. Los mismos índices sirven para la validación interna (comparar con la matriz de similitud) como para la externa (comparar con la matriz correspondiente a la partición correcta). (DUBES. 1988).

Sin embargo, la mayoría de los índices de validación, son específicos del tipo de validación. Típicamente la validación externa se reduce a la comparación de dos particiones: la evaluada y la correcta. Por ello, los índices de validación externa suelen ser medidas de similitud de particiones. Los índices de validación interna, en cambio, suelen estimar la cohesión y la separación de los clúster y aunque la terminología no está unificada generalmente, son denominados como índices de validación de clúster o clúster validity index (CVI).

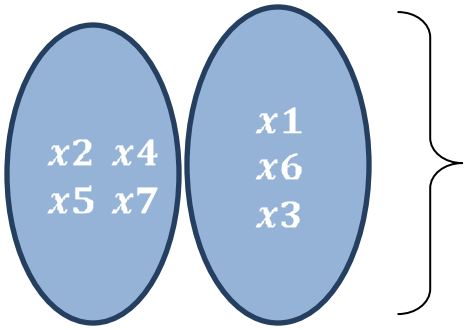
#### 1.4.1.1Validación Externa

La Validación Externa, se basa fundamentalmente en la comparación de dos estructuras de agrupamiento. O sea, este intenta lograr que la estructura que se deriva de un algoritmo de clúster, sea lo más parecida a este agrupamiento ideal. Como se acaba de mencionar, los índices de validación externa suelen ser medidas de similitud de particiones. Existen diversos tipos de medidas de similitud, dependiendo de la información en que se basan:

Recuento de pares: este tipo de medida de similitud se basa fundamentalmente, en la comparación de dos matrices, de las cuales se recuentan las distintas situaciones en las que se encuentran los posibles pares pertenecientes a cada una de las matrices.

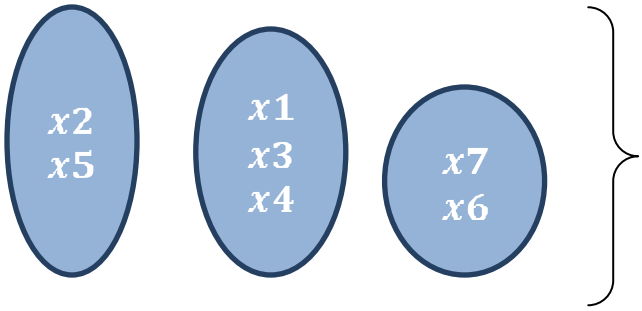
A continuación se realiza una descripción del funcionamiento de Validación Externa:

Tenemos un clúster$P$, que por lo general es conocido de antemano, de un conjunto de datos $X$.




Estructura de Agrupamiento $P = \{(x2, x4, x5, x7) ; (x1, x6, x3)\}$

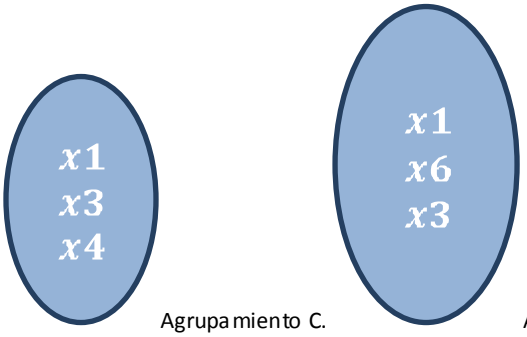
Este es independiente de la estructura de agrupación C, que es el resultado de un algoritmo de agrupamiento. La validación de C, por el Criterio Externo, se consigue comparando C y P.(RUI XIU AND DONALD C. WUNSCH 2009).




Estructura de Agrupamiento $C = \{(x2, x5) ; (x1, x3, x4) ; (x7, x6)\}$

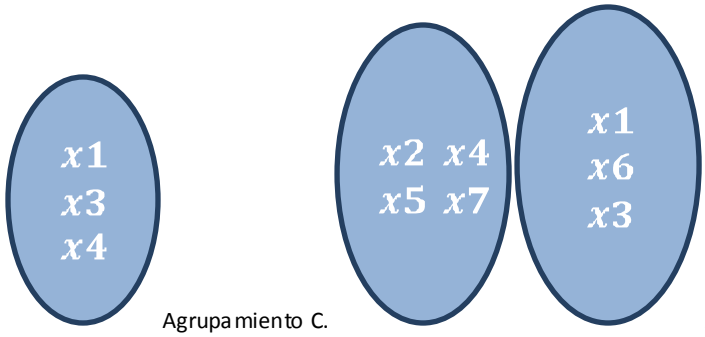
Teniendo en cuenta un par de puntos de datos $xi$ y $xj$ del conjunto de datos $X$, existen cuatro casos diferentes sobre la base de la forma $xi$ y $xj$ que se encuentran en $C$ y $P$, que se utilizarán para llevar a cabo el Criterio Externo.

- **Caso 1:** $xi$ y $xj$ pertenecen al mismo clúster de $C$ y de $P$.



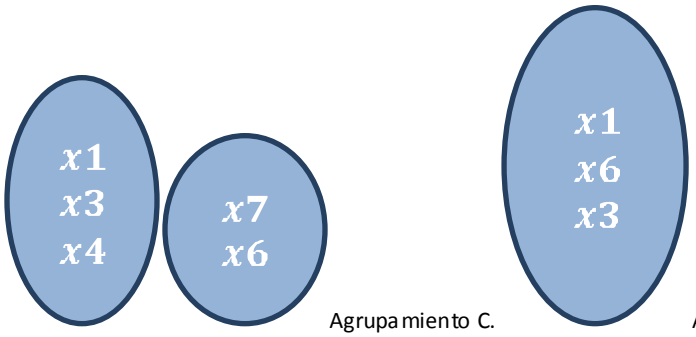

Agrupamiento C. Agrupamiento P.

- **Caso 2:** $xi$ y $xj$ pertenecen al mismo clúster de $C$ y diferentes clúster de $P$.



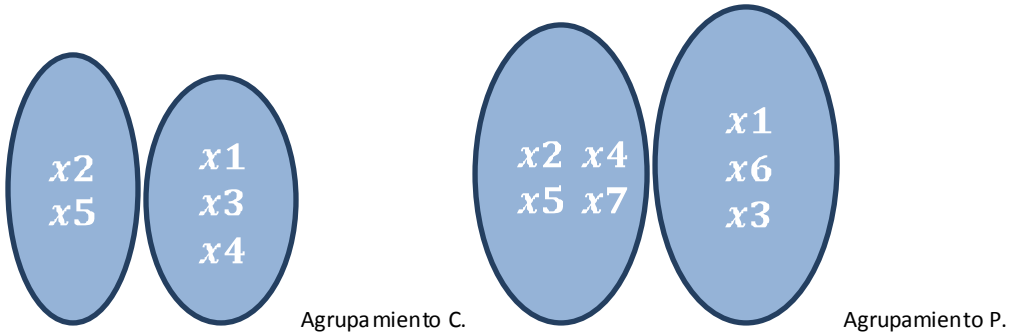

Agrupamiento C. Agrupamiento P.

- **Caso 3:** $xi$ y $xj$ pertenecen a diferentes clúster de $C$ y el mismo clúster de $P$.




Agrupamiento C. Agrupamiento P.

- **Caso 4:** $xi$ y $xj$ pertenecen a diferentes clúster de $C$ y de $P$.




Agrupamiento C. Agrupamiento P.

A continuación se definen algunas variables que nos permitirán aplicar algunos índices para la evaluación del algoritmo de agrupamiento:

- $a$: es el número de pares de objetos que pertenecen al mismo clúster de $C$ y de $P$.

- $b$ : es el número de pares de objetos que pertenecen al mismo clúster de $C$ y diferentes clúster de $P$.
- $c$ : es el número de pares de objetos que pertenecen a diferentes clúster de $C$ y el mismo clúster de $P$.
- $d$ : es el número de pares de objetos que pertenecen a diferentes clúster de $C$ y de $P$.

Ahora se muestran algunos índices para la validación:

1. **Rand index (Rand, 1971)**

$$R = \frac{(a+d)}{M}$$

**2. Jaccard coefficient (Coeficiente de Jaccard)**

$$J = \frac{a}{(a+b+c)}$$

**Fowlkes and Mallows index (Fowlkes and Mallows, 1983)**

$$\frac{a}{\sqrt{(a+b)(a+c)}}$$

**¶ Statistics (estadísticas)**

$$¶ = \frac{Ma - m_1 m_2}{\sqrt{m_1 m_2 (M-m_1)(M-m_2)}}$$

Donde $m1 = a + b \; y \; m2 = a + c$

$M = a + b + c + d.$

Un valor de Rand, Jaccard y FM y estadísticas cercano a 1 indica un buen agrupamiento de los datos.

### 1.4.1.2Validación Interna

A diferencia de la validación externa, los índices de validación interna se concentran en el trabajo sobre una sola estructura de agrupamiento. Existe un gran número de índices para la validación interna.(BOUGUESSA 2006; KUO-LUNG WU 2009) Estos se basan en los conceptos de cohesión y separación mencionados anteriormente en la introducción de la tesis. En el caso de los índices de validación interna, es un problema no tratado, el hecho de que índice funciona mejor en un contexto determinado, por lo que la mayoría de los autores han empleado estos, sin profundizar en los criterios a seguir para su elección. En el trabajo (RAMAKRISHNA 2005) se define una taxonomía para separar estos índices en dos grandes familias, los basados en la suma y los basados en el radio (centroide). Bajo esta definición, aquellos índices basados en la suma, ponderan mejor los modelos de agrupamientos que no son basados en el centroide, mientras que los índices tipo radio, dan ventajas a los índices basados en el centroide. Este es un criterio importante a la hora de seleccionar un índice de validación de clúster. A continuación se muestra una tabla con los principales índices de validación interno. Es importante resaltar que existen más de 30 (COOPER. 1985) de estos índices definidos en la literatura, por lo que resulta muy difícil detallar todos estos.

| Índice de validación de clúster Interno | Tipo | Referencia |
|---|---|---|
| Davies Bouldin | Radio | (DAVIES Y BOULDIN 1979) |
| DUNN | Suma | (DUNN. 1973) |
| Calinski-Harabasz | Radio | (HARABASZ. 1974) |
| C-Index | Suma | (L. J. HUBERT Y J. R. LEVIN 1976) |
| COP | Radio | (RAMAKRISHNA 2005) |

En el caso de estos índices, las relaciones intra-grupos de aquellos índices que son de tipo *radio,* están determinados por la distancia de los elementos de un grupo al centroide, en cambio aquellos de tipo *suma,* su relación intra-grupo, la determina la suma de las distancias de los elementos de un grupo.

Para determinar las relaciones inter-grupos en algunos de estos índices, se utiliza la distancia entre sus centroides, como es el caso de Davies Boulding y Calinski-Harabasz. Por otra parte, en el caso del índice COP y DUNN, se utilizan relaciones de máximo o mínimo de la distancia entre elementos de grupos diferentes.

## 1.5 Metodologías, Herramientas y tecnologías utilizadas

En un proyecto de desarrollo de software la metodología define quién, qué, cuándo y cómo debe hacerlo. No existe una metodología de software universal. Cada equipo de desarrollo escoge la metodología según las características de su proyecto, por lo que es importante determinar el alcance del proyecto antes de escoger la metodología que se va a usar en el desarrollo del mismo. A continuación se realiza una breve descripción de las metodologías y herramientas utilizadas en el desarrollo de la aplicación.

### 1.5.1Metodologías ágiles

Las metodologías ágiles, proporcionan un marco de trabajo dentro de la ingeniería. Existen diversos métodos de desarrollo ágil, la mayoría minimizan riesgos construyendo software en cortos períodos de tiempo, entre ellos se encuentran: Adaptive Software Development, Crystal Clear, Open Unified Process y Programación Extrema.(FOWLER 2006).

#### 1.5.1.1 Programación Extrema (XP)

Esta metodología ágil se utiliza para proyectos sencillos, de ahí que el objetivo principal sea lograr la satisfacción del cliente, es decir, trata de entregarle al cliente el software que necesita y cuando lo necesita. Además persigue potenciar al máximo el trabajo en grupo(BARBONE 2004). Se basa en la simplicidad, comunicación y el reciclado continuo de código. Entre las características más significativas de XP se encuentran:

- Orientado a la persona que produce y usa el software.
- Reduce el costo del cambio en las etapas de vida del sistema. (SIERRA 2007).

La metodología XP se basa en cuatro valores imprescindibles para el desarrollo de software:

- Simplicidad: enfocado en un diseño sencillo del código generado.
- Comunicación: potenciada por el desarrollo en pares, la presencia del cliente y la simplicidad en cuanto al código.
- Retroalimentación: propiciada por el protagonismo del cliente que participa activamente y por el trabajo en ciclos cortos.
- Coraje: enfrentando decisiones, en ocasiones complejas, que pudieran afectar el tiempo de desarrollo y la calidad del producto. (SIERRA 2007)

El ciclo de vida ideal consta de 5 fases:

- Exploración: los clientes plantean a grandes rasgos las Historias de usuario que son de interés para la primera entrega del producto.
- Planificación de Entregas: se establece la prioridad de cada Historia de usuario y los programadores realizan una estimación del esfuerzo necesario de cada una de ellas.
- Iteraciones: incluye varias iteraciones sobre el sistema antes de ser entregado. El plan de entrega está compuesto por iteraciones de no más de tres semanas.
- Prueba: requiere de pruebas adicionales y revisiones del rendimiento antes de que el sistema sea trasladado al entorno del cliente.
- Mantenimiento: mientras la primera versión se encuentra en producción, el proyecto XP debe mantener el sistema en funcionamiento al mismo tiempo que desarrolla nuevas iteraciones.

### 1.5.2Herramientas utilizadas para el desarrollo de la aplicación

Microsoft Visual Studio es un entorno de desarrollo integrado (IDE, por sus siglas en inglés) para sistemas operativos Windows. Soporta varios lenguajes de programación tales como Visual C++, Visual C#, Visual J#, y Visual Basic .NET, al igual que entornos de desarrollo web como ASP.NET. Aunque actualmente se han desarrollado las extensiones para muchos otros. Visual Studio permite a los desarrolladores crear aplicaciones, sitios y aplicaciones web, así como servicios web en cualquier

entorno que soporte la plataforma .NET (a partir de la versión .NET 2002). Así se pueden crear aplicaciones que se intercomuniquen entre estaciones de trabajo, páginas web y dispositivos móviles.

### 1.4.2.1 Visual Studio 2010

Visual Studio 2010, acompañada por .NET Framework 3.5, lanzó su última versión el 12 de abril de 2010. Entre sus más destacables características, se encuentran la capacidad para utilizar múltiples monitores, así como la posibilidad de desacoplar las ventanas de su sitio original y acoplarlas en otros sitios de la interfaz de trabajo. Además ofrece la posibilidad de crear aplicaciones para muchas plataformas de Microsoft, como Windows, Azure, Windows Phone 7 o Sharepoint. Microsoft ha sido sensible a la nueva tendencia de las pantallas táctiles y con este Visual Studio 2010 también es posible desarrollar aplicativos para pantallas multitáctiles.

### 1.5.2.2 Lenguaje de Programación C++

C++ es un lenguaje de programación diseñado a mediados de los años 1980 por Bjarne Stroustrup. La intención de su creación fue el extender al exitoso lenguaje de programación C con mecanismos que permitan la manipulación de objetos. En ese sentido, desde el punto de vista de los lenguajes orientados a objetos, el C++ es un lenguaje híbrido. Posteriormente se añadieron facilidades de programación genérica, que se sumó a los otros dos paradigmas que ya estaban admitidos en el lenguaje (programación estructurada y la programación orientada a objetos). Por esto se suele decir que el C++ es un lenguaje de programación multiparadigma. Una particularidad del C++ es que brinda la posibilidad de redefinir los operadores. El nombre C++ fue propuesto por Rick Mascitti en el año 1983, cuando el lenguaje fue utilizado por primera vez fuera de un laboratorio científico. Antes se había usado el nombre "C con clases". En C++, la expresión "C++" significa "incremento de C" y se refiere a que C++ es una extensión de C.1.3.3**.**

## 1.5.3 Herramienta de modelado

### 1.5.3.1 Visual Paradigm

“Es una herramienta CASE (ComputerAided Software Engineering, Ingeniería de Software Asistida por Computadora), que utiliza el lenguaje UML”. Soporta el ciclo de vida completo de desarrollo de software y tiene la ventaja de que puede ejecutarse por diferentes sistemas operativos debido a que es multiplataforma. Ayuda a la construcción de aplicaciones con calidad y menor costo; además posee

una interfaz gráfica amigable. Facilita el modelado de base de datos, requerimientos, los procesos de negocio, la interoperabilidad y la generación de documentación.

## 1.6Conclusiones del Capítulo

En la definición de los métodos de validación de clúster, se identificaron dos enfoques fundamentales, los métodos de validación interna y los métodos de validación externa.

En el caso de los métodos de validación externa, tienen su base en la determinación de cuan buena es una partición con respecto a una partición ideal conocida. Para la construcción de los índices de validación externa, se identifican 4 parámetros esenciales, a partir de los cuales se modelan las diferentes métricas.

En el caso de los métodos de validación interna, se identifican dos tipos de métodos, los basados en el radio y en la suma. La evaluación de varios de estos índices, permite identificar elementos comunes para la construcción de cada tipo de índice, lo cual se debe tomar en cuenta como parte de la propuesta.

Tomando en cuenta las características con la que se desarrollará la investigación, en la cual el cliente mantiene una participación directa en la construcción del sistema y por no contar con un equipo de amplio trabajo, se define como metodología de desarrollo de software XP.

# CAPITULO 2. Características del Sistema.

## Introducción

En este capítulo se describe la propuesta de solución del problema, además de describir los requisitos funcionales y no funcionales que debe tener el sistema para cumplir con las condiciones que propone el cliente, así como los procesos que se van a automatizar. Se definen las historias de usuarios, las cuales describirán lo que el sistema debe realizar.

## 2.1 Propuesta de Solución

Para darle solución a este trabajo, se implementará una aplicación capaz de agrupar todos los requisitos necesarios que le permitan a cualquier persona calificada en nuestra universidad, validar los algoritmos de agrupamiento apoyándose de una herramienta lo suficientemente genérica, que implemente los enfoques propuestos en la investigación, Validación Externa y Validación Interna. De forma tal, que el cliente se apoye de esta aplicación para evaluar la calidad de la estructura resultante de un algoritmo de agrupamiento y así poder determinar si el algoritmo de clúster realizado a un conjunto de datos es bueno o no.

En la Figura 5 se muestra la interrelación existente entre un conjunto de objetos, los algoritmos de agrupamiento y los índices de validación de clúster, en el proceso de agrupamiento y validación de un conjunto de objetos. Esta describe la representación de un conjunto de datos y la selección de un algoritmo de agrupamiento, capaz de obtener una estructuración de los objetos, que cumpla en cierto grado, con un conjunto de requisitos necesarios para un problema (medidos a partir de uno o más índices de validación), esto debe aportar la información necesaria para la solución del problema en cuestión.

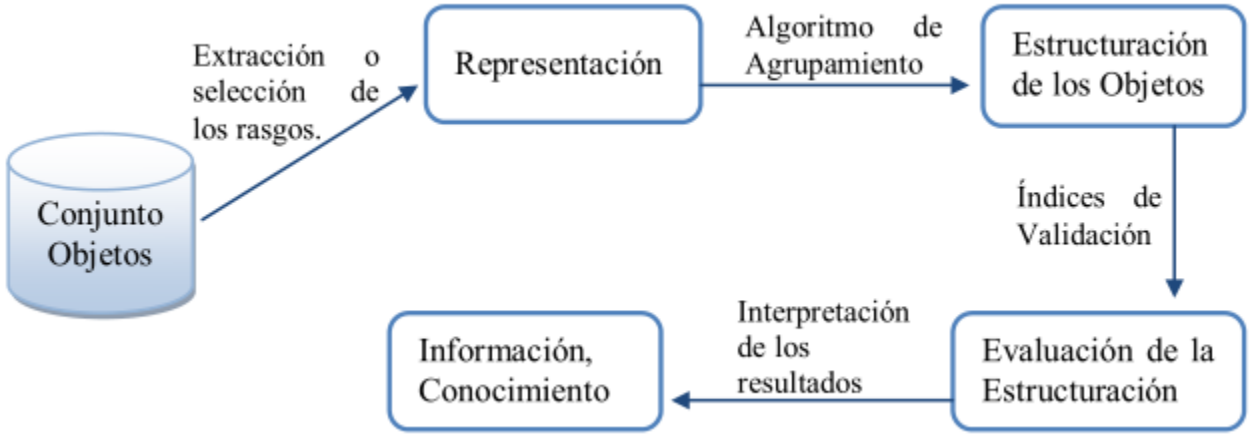



**Figura 5. Proceso de agrupamiento y validación de un conjunto de objetos.**

La distancia más habitual usada en este tema del análisis de clúster, es la euclidiana, con lo que se generan clúster de forma esférica. Como propuesta para calcular la distancia entre los casos de una base de datos, se decidió utilizar la distancia euclidiana cuadrado.

### 2.1.1 Índices de Validación Interna

Como parte de la solución para dar respuesta a un problema de agrupamiento y validar el mismo por medio de la Validación Interna, se propone el índice de validación de clúster David Bouldin en la implementación de este enfoque. Este índice está basado en la relación de los datos con su centroide, para establecer las relaciones intra-clúster. La incorporación de nuevos índices de validación de clúster resulta muy sencilla, pues existen funcionalidades en el componente que se han definido, que pueden ser reutilizadas para la construcción de los mismos. Estas funcionalidades cubren múltiples heurísticas, que se emplean en la modelación de los índices de validación de clúster. A continuación se mencionan estas funcionalidades:

| Funcionalidades | Expresión |
|---|---|
| Mínima distancia entre dos objetos de un mismo grupo | $d = \min_{x_i, x_j \in C_k} d(x_i, x_j)$ |
| Mínima distancia entre dos objetos de grupos | $d = \min_{x_i \in C_k, x_j \in C_l} d(x_i, x_j)$ |

| | |
|---|---|
| diferentes | |
| Máxima distancia entre dos objetos de un mismo grupo | $d = \max_{x_i, x_j \in C_k} d(x_i, x_j)$ |
| Máxima distancia entre dos objetos de grupos diferentes | $d = \max_{x_i \in C_k, x_j \in C_l} d(x_i, x_j)$ |
| Media de la distancia dentro de un grupo. | $d = \frac{1}{\lvert C_k \rvert} \sum_{x_i, x_j \in C_k} d(x_i, x_j)$ |
| Centroide de un grupo | $\overline{X_k} = \frac{1}{\lvert C_k \rvert} \sum_{i=1}^{K} x_i^k$ |
| Distancia de un elemento a su centroide | $d = d_{x_i \in C}(x_i, \bar{C})$ |

Para lograr una implementación que ocupe el espacio de memoria adecuado de cada una de estas funciones, se emplea en el caso de las relaciones inter-grupos, un arreglo bidimensional, que forma una matriz triangular simétrica, donde solo se almacenará información por debajo de la diagonal principal. En el caso de los cálculos relacionados con las distancias intra-grupos se utiliza un arreglo unidimensional.

Para la implementación del índice Davis Bouldin se siguen los pasos descritos en el algoritmo 3.

**Algoritmo 3.** Davies Bouldin.
**Entrada:** Partición.
**Salida:** Índice de validación de clúster

1. Calcular los centroides de cada grupo
2. Determinar la distancia de cada elemento a su centroide
3. ***for*** k= 1***to*** Numero de grupos ***do***
4. $S(C_k) = \frac{1}{\lvert C_k \rvert} \sum_{x_i \in C_k} d(x_i, \overline{C_k})$
5. ***end for***
6. Calcular Davies Bouldin $DB = \frac{1}{\lvert P \rvert} \sum_{C_k \in P} \max_{C_l \in P, k \neq l} \frac{S(C_k) + S(C_l)}{d(\overline{C_k}, \overline{C_l})}$

## 2.1.2 Índices de Validación Externa

Como se explicó con anterioridad, los métodos de validación externos de clúster requieren conocer a priori la partición ideal. Por lo que dada una partición obtenida de un algoritmo de agrupamiento, se determina cuánto esta se parece a la partición ideal. La determinación de una medida externa de clúster, está basada en determinar los valores de los parámetros a, b, c, d. Para obtener los valores de estos parámetros nos basamos en el enfoque del conteo de pares, para lo cual se definen los siguientes elementos:

Sean $P = \{C_1, C_2, \dots, C_k\}$ y $P^* = \{C_1, C_2, \dots, C_l\}$ dos particiones se define a $n_p$como el número de pares de elementos $x_i, x_j \in C_k$ para la partición $P$ y sea$n'_p$ el números de pares de elementos $x_i \in C_l, x_j \in C_k$. De igual manera se definen $n_{p^*}$ y $n'_{p^*}$para la partición $P^*$.

Luego se determinan los parámetros a, b, c, d de la siguiente manera.

$$a = |n_p \cap n_{p^*}| \qquad c = |n'_p \cap n_{p^*}|$$

$$b = |n_p \cap n'_{p^*}| \qquad d = |n'_p \cap n'_{p^*}|$$

Para lograr una implementación de las relaciones entre pares de objetos, se utiliza un arreglo bidimensional binario, que indica si el elemento $(i, j)$ pertenece a un mismo grupo. Luego con esta matriz diagonal, se determinan cada uno de los parámetros, que son la base para la construcción de los índices de validación externa. El algoritmo 4 muestra de manera detallada la implementación de los métodos de validación externa.

| **Algoritmo 3.** Método de validación externa<br>**Entrada:** Partición ideal, Partición obtenida.<br>**Salida:** Índice de validación de clúster |
|---|
| 1. Calcular la matriz binaria para cada partición.<br>2. Determinar los parámetros a, b, c, d<br>3. Calcular el índice de validación externa. |

## 2.2 Especificación de los requisitos del software

A continuación se muestran los requisitos del sistema, los cuales determinan lo que se desea hacer o implementar en el sistema. Estos requisitos se dividen en dos categorías, la Lista de Reserva del Producto (LRP) y las funcionalidades del sistema o historias de usuario. Las LRP son todas aquellas que especifican los criterios que pueden usarse para juzgar la operación de un sistema. Son requisitos que imponen restricciones en el diseño o la implementación. Son propiedades o cualidades que el producto debe tener. Existen diferentes categorías de las LRP, algunas de ellas son:

- De Usabilidad.
- De Rendimiento.
- De Software.

### 2.2.1 Lista de Reserva del Producto (LRP)

**Usabilidad:**

1. El sistema debe ser desarrollado de forma tal que los investigadores puedan reutilizar en sus implementaciones los métodos que se encuentran en el componente.

**Rendimiento:**

2. Los métodos propuestos como parte de la solución, deben estar completamente enfocados al ahorro de espacio en memoria y optimización.

**Software:**

3. El sistema podrá ser utilizado en el sistema operativo Windows, conjuntamente con esto debe tener en la computadora el .Net, con el framework .net 3.5.

### 2.2.2 Funcionalidades del sistema

Las funcionalidades del sistema son declaraciones de los servicios que debe proporcionar el sistema. Estas mostraran lo que debe hacer el sistema para el cumplimiento del objetivo trazado, para ello se enumerarán a través de funcionalidades, las funciones que el sistema deberá ser capaz de realizar. En el caso de la metodología XP, estas funcionalidades coinciden con las Historias de Usuario.

**1.**Validar un agrupamiento mediante la Validación Externa.

**2.**Validar un agrupamiento mediante la Validación Interna.

## 2.3 Fase Exploración

La fase Exploración es la fase inicial del ciclo de vida de la metodología XP, en esta fase se define el alcance general del proyecto, aquí el cliente define lo que necesita mediante la redacción de Historias de Usuario y a partir de ahí, los programadores estiman el tiempo de desarrollo, basándose de esta información.

### 2.3.1 Historias de Usuarios

Las historias de usuarios son realizadas por el cliente a su manera o lenguaje, lo que brindaría una idea de lo que ese sistema debe realizar. En esta técnica se debe tener un nivel de especificación muy alto, para que los programadores puedan desarrollarlas en el menor tiempo posible. Estas formas de especificar los requisitos del software deben ser implementadas en un corto período de tiempo, si excede este período de estimación planificado, la historia de usuario debe ser dividida en dos o más historias.(JOSKOWICZ 2008).

| **Historia de Usuario** | |
|---|---|
| **Número**:1 | Nombre de Historia de Usuario: Validar un agrupamiento mediante la validación externa. |
| **Modificación de historia de Usuario** | |
| **Usuario**: Cliente | **Iteración Asignada**:1 |
| **Prioridad en negocio**: Alta | |
| **Riesgo en Desarrollo**: Alto | |
| **Descripción**: Brinda la posibilidad de evaluar la calidad de un algoritmo de agrupamiento realizado, por medio de los índices de validación externa. | |

| **Historia de Usuario** | |
|---|---|
| **Número**:1 | Nombre de Historia de Usuario: Validar un agrupamiento mediante la validación interna. |
| **Modificación de historia de Usuario** | |
| **Usuario**: Cliente | **Iteración Asignada**:1 |
| **Prioridad en negocio**: Alta | |
| **Riesgo en Desarrollo**: Alto | |
| **Descripción**: Brinda la posibilidad de evaluar la calidad de un algoritmo de agrupamiento realizado, por medio de los índices de validación interna. | |

## 2.4 Fase Planificación

La planificación es una fase corta, en la que el cliente, los gerentes y el grupo de desarrolladores acuerdan el orden en que deberán implementarse las historias de usuario y asociadas a éstas las entregas. Típicamente esta fase consiste en una o varias reuniones grupales de planificación.

### 2.4.1 Estimación de esfuerzo por Historia de Usuario

La tabla que se muestra a continuación contiene la estimación de esfuerzo por Historia de Usuario según el orden a realizar, para ello se utilizan puntos. Un punto equivale a una semana ideal de programación. Para el desarrollo del sistema se decidió asignarle a cada historia de usuario el tiempo de estimación siguiente.

| **Historia de Usuario** | **Estimación** |
|---|---|
| Validar un agrupamiento mediante la validación | 1 |

| | |
|---|---|
| externa. | |
| Validar un agrupamiento mediante la validación interna. | 1 |

## 2.4.2 Plan de Iteraciones

Una vez identificadas las Historias de Usuario y la estimación de esfuerzo de cada una de ellas, se procede a realizar el plan de iteraciones, el cual consiste en determinar cuál será la iteración correspondiente a cada HU para ser implementada. Debido a que el riesgo para desarrollar cada una de las HU es el mismo, se decidió realizar una sola iteración:

**Iteración 1:** En esta iteración se implementarán las historias de usuarios de prioridad alta, aquí están las funcionalidades que son indispensables para el cliente y que son claves en el funcionamiento de la aplicación.

## 2.4.3 Plan de duración de Iteraciones

A continuación se muestra el plan de duración de cada una de las Historias de Usuario, para esto se tuvo en cuenta la estimación de esfuerzo de cada HU.

| Iteraciones | Historias de Usuario por iteración | Duración total de la iteración en semanas |
|---|---|---|
| **Iteración 1** | Validar un agrupamiento mediante la validación externa. | 2 |
| | Validar un agrupamiento mediante la validación interna. | |

## 2.4.4 Plan de Entregas

El cronograma de entregas establece qué historias de usuario serán agrupadas para conformar una entrega y el orden de las mismas, asumiendo que la implementación empieza en el mes de abril:

| Herramienta | Final 1ra iteración |
|---|---|
| Componente para la validación de algoritmos de agrupamiento. | 05/06/2013<br>Versión 0.1 |

## 2.5 Diseño

La metodología X.P sugiere que hay que conseguir diseños simples y sencillos. Hay que procurar hacerlo todo lo menos complicado posible para conseguir un diseño entendible y de fácil implementación, que a la larga costará menos tiempo y esfuerzo desarrollar. Logrando así, que cambios futuros, puedan realizarse sin mucha dificultad. La metodología XP no define una técnica específica de modelado, pueden utilizarse indistintamente sencillos esquemas, tarjetas CRC (clase, responsabilidad, colaboración) o diagramas de clases utilizando UML.

Con el objetivo de tener detalles concretos de la implementación del sistema, se decidió utilizar como técnica de modelado diagramas de clases.

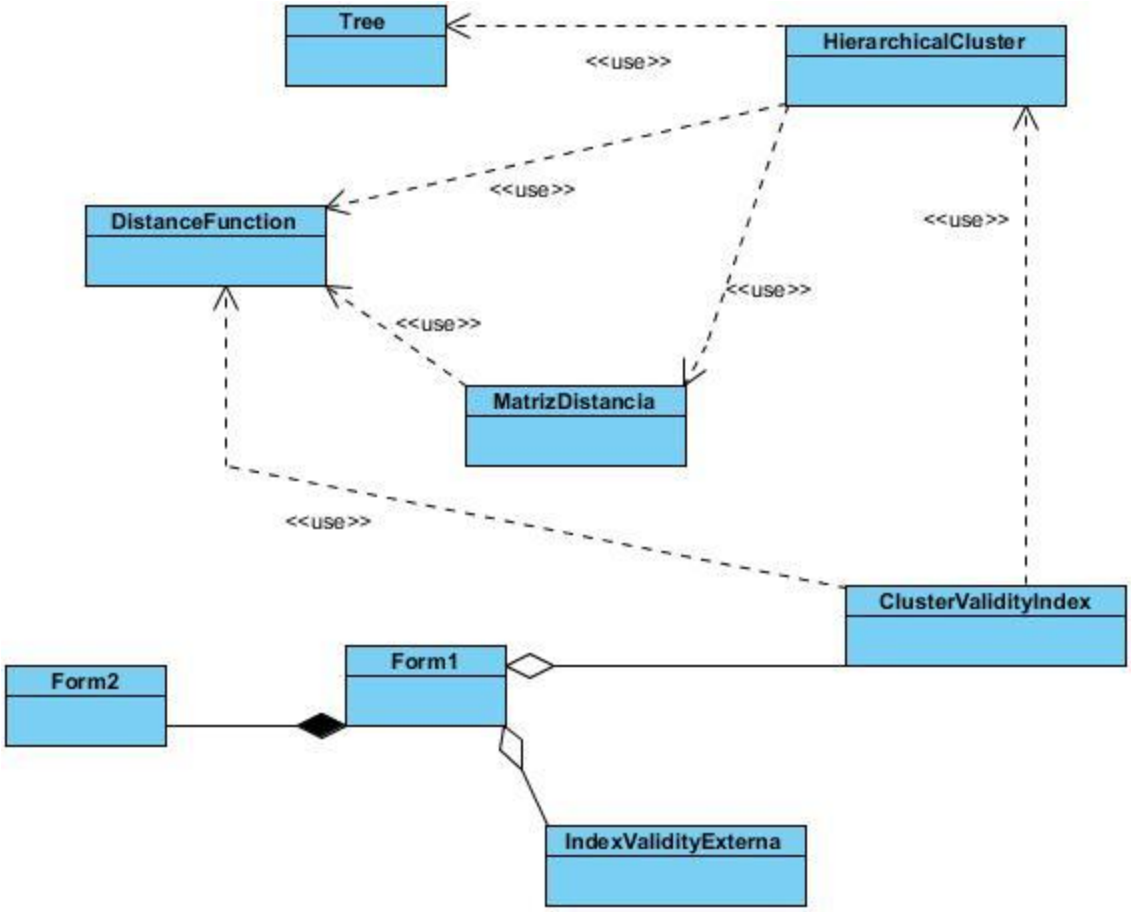



**Figura 6. Diseño del diagrama de Clases.**

El siguiente diagrama representa el diseño de clases propuesto para la implementación del componente. En él, se representan las clases propuestas para dar solución al problema planteado, estas clases son, Cluster Validity Index, la cual es la encargada de contener los índices de validación interna, ejemplo de uno de estos índices es Davies Bouldin. Matriz Distancia, esta clase contiene la matriz de distancia de cada una de las estructuraciones que se forma al realizar un algoritmo de agrupamiento a un conjunto de datos. Distance Function, aquí se encuentra el método que calcula la distancia entre dos casos de un agrupamiento a través de la euclidiana cuadrado. Hierarchical Cluster, en esta clase están los algoritmos de agrupamiento jerárquicos, utilizados para realizar el experimento para probar cada uno de los índices de validación de clúster. Además una clase de tipo estructura llamada Tree, todas estas clases presentan una relación de uso. Además está la clase Index Validity Externa, la cual contiene las funcionalidades necesarias para validar un agrupamiento por medio de los

índices de validación externa, por último están los dos componentes visuales, empleados para que el cliente interactúe con la aplicación.

### 2.5.1 Arquitectura del Sistema

La arquitectura de software es un conjunto de patrones que proporcionan un marco de referencia necesario para guiar la construcción de un software, permitiendo a los programadores, analistas y todo el conjunto de desarrolladores del software compartir una misma línea de trabajo y cubrir todos los objetivos y restricciones de la aplicación. Es considerada el nivel más alto en el diseño de la arquitectura de un sistema puesto que establecen la estructura, funcionamiento e interacción entre las partes del software. Una arquitectura de software define la estructura del sistema.(ERIKA CAMACHO Abril, 2004).

Para la solución de este sistema se utilizó la arquitectura en capas, la cual es la vista conceptual de la estructura de la aplicación. Estas capas son: la de Presentación, esta es la encargada de agrupar las vistas con las que va a interactuar el usuario, y la capa de Negocio que contiene las entidades que manejan la lógica del negocio.

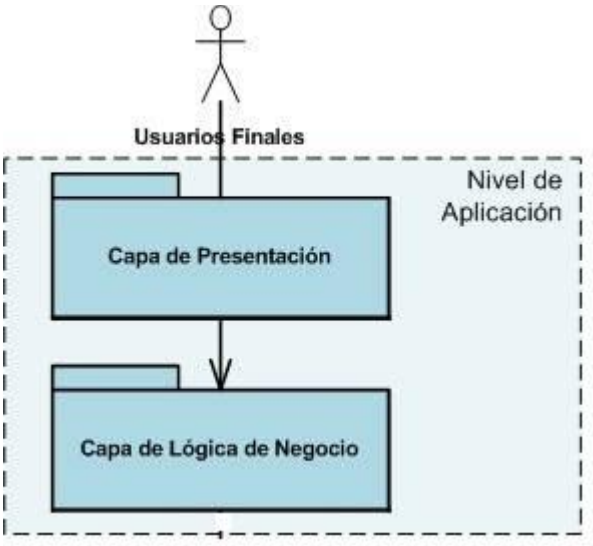



**Figura 7. Arquitectura en capas.**

## 2.5.2 Patrones GRASP

Los patrones GRASP, describen los principios fundamentales del Diseño Orientado a Objetos (DOO) y la asignación de responsabilidades expresados como patrones. En diseño orientado a objetos, **GRASP** son patrones generales de software para la asignación de responsabilidades, es el acrónimo de "GRASP (object-oriented design General Responsibility Assignment Software Patterns)". Aunque se considera que más que patrones propiamente dichos, son una serie de "buenas prácticas" de aplicación recomendable en el diseño de software.

**Experto en información:** El GRASP de experto en información es el principio básico de asignación de responsabilidades. Nos indica, por ejemplo, que la responsabilidad de la creación de un objeto o la implementación de un método, debe recaer sobre la clase que conoce toda la información necesaria para crearlo. De este modo obtendremos un diseño con mayor cohesión y así la información se mantiene encapsulada (disminución del acoplamiento).

Un ejemplo donde se evidencia este patrón es en la clase Cluster Validity Index, esta clase contiene la información necesaria para implementar cada uno de los índices de validación de clúster interna, de igual manera sucede con la clase Index Validity Externa.

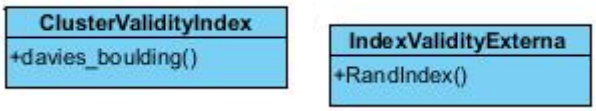



**Figura 8. Ejemplo del patrón GRASP experto en información.**

**Creador:** El patrón creador nos ayuda a identificar quién debe ser el responsable de la creación (o instanciación) de nuevos objetos o clases.

La nueva instancia deberá ser creada por la clase que:

- Tiene la información necesaria para realizar la creación del objeto.
- Usa directamente las instancias creadas del objeto.
- Almacena o maneja varias instancias de la clase
- Contiene o agrega la clase.

Una de las consecuencias de usar este patrón es la visibilidad entre la clase creada y la clase creador. Una ventaja es el bajo acoplamiento, lo cual supone facilidad de mantenimiento y reutilización. La creación de instancias es una de las actividades más comunes en un sistema orientado a objetos. En consecuencia es útil contar con un principio general para la asignación de las responsabilidades de creación. Si se asignan bien, el diseño puede soportar un bajo acoplamiento, mayor claridad, encapsulación y reutilización.

Este patrón se pone de manifiesto en la clase Matriz Distancia, la cual crea una instancia de la clase Distance Function, para poder utilizar su función de distancia.

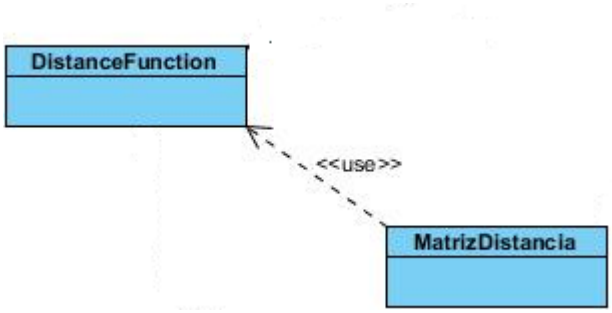



**Figura 9. Ejemplo del patrón GRASP creador.**

**Controlador:** El patrón controlador es un patrón que sirve como intermediario entre una determinada interfaz y el algoritmo que la implementa, de tal forma que es la que recibe los datos del usuario y la que los envía a las distintas clases según el método llamado. Este patrón sugiere que la lógica de negocios debe estar separada de la capa de presentación, esto para aumentar la reutilización de código y a la vez tener un mayor control. Se recomienda dividir los eventos del sistema en el mayor número de controladores para poder aumentar la cohesión y disminuir el acoplamiento.

**Alta cohesión y bajo acoplamiento:** Los conceptos de cohesión y acoplamiento no están íntimamente relacionados, sin embargo se recomienda tener un mayor grado de cohesión con un menor grado de acoplamiento. De esta forma se tiene menor dependencia y se especifican los propósitos de cada objeto en el sistema.

**Alta cohesión:** Se refiere a que la información que almacena una clase debe de ser coherente y debe estar (en la medida de lo posible) relacionada con ella misma.

**Bajo acoplamiento:** Es la idea de tener las clases lo menos ligadas entre sí. De tal forma que en caso de producirse una modificación en alguna de ellas, esto tenga la mínima repercusión posible en el resto de clases, potenciando la reutilización, y disminuyendo la dependencia entre las clases.

Como se ve representado en el diagrama de clases, en la mayoría de estas entidades se trató de que quedaran lo menos ligadas entre sí, vinculadas solamente por una relación de uso, la cual asegura que en caso de realizarse una modificación en una de estas clases, esto no afectara a las restantes, asegurando así el bajo acoplamiento. Ejemplo de esto se muestra en la siguiente figura.

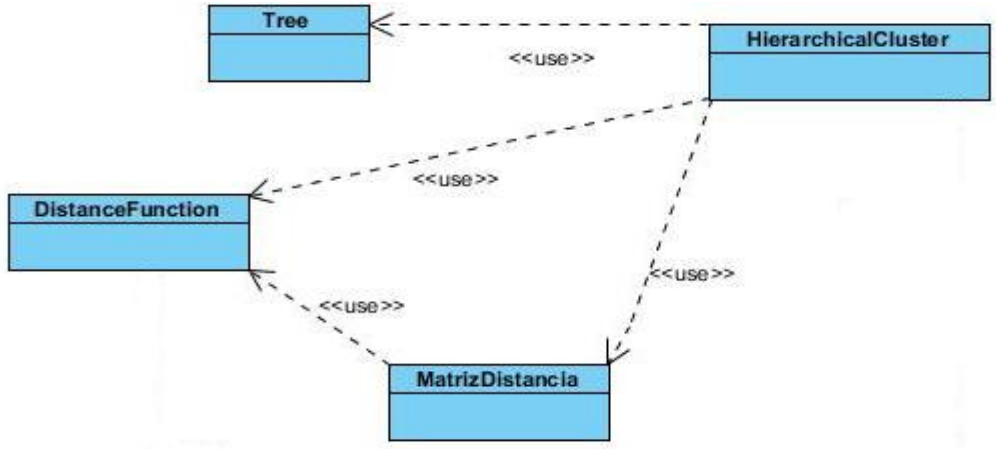



**Figura 10. Ejemplo del patrón GRASP de bajo acoplamiento.**

## 2.6 Implementación

XP propone tener en cuenta algunos aspectos a la hora de la implementación, como ya se dijo en la introducción, el cliente es una parte más del equipo de desarrollo; su presencia es indispensable en las distintas fases de XP. A la hora de implementar una historia de usuario su presencia es aún más necesaria. Un concepto clave durante la actividad de implementación es la programación en pareja. XP recomienda que dos personas trabajen juntas en una estación de trabajo para crear el código de una Historia de usuario.

Para dar solución a este trabajo se tuvo en cuenta algunos elementos de optimización, entre los que cabe destacar, el ahorro de espacio en memoria a la hora de realizar la implementación. Debido a que la matriz de distancia de los agrupamientos es una matriz simétrica, o sea, que en las coordenadas, (x, j) y (j, x), se tiene el mismo valor, se decidió implementar una funcionalidad que creara una matriz

triangular y así trabajar solamente con la parte inferior izquierda, logrando un menor uso de memoria durante la ejecución del programa.

### 2.6.1 Tareas de ingeniería

Para llevar a cabo la correcta implementación de las HU descritas por el cliente, se deben definir por parte del equipo de desarrollo las tareas de ingeniería, para realizar estas tareas se cuenta con una plantilla, la cual permite definir cada una de las actividades que estarán asociadas a las historias de usuario y que permitirán su implementación.

Las tareas de ingeniería también conocidas como tareas de implementación, permiten a los desarrolladores obtener un nivel de detalle más avanzado que el que propician las HU.
Para la implementación del software se planificó una sola iteración de trabajo. En esta única iteración para implementar cada HU se trazaron las siguientes tareas de ingeniería:

| **Historias de Usuario** | **Tareas de Ingeniería** |
|---|---|
| Validar un agrupamiento mediante la validación externa. | ➢ Implementar la funcionalidad MatrixArreglo<br>➢ Implementar la funcionalidad LlenarMatriz<br>➢ Implementar la funcionalidad CompararMatrices<br>➢ ImplementarRandIndex<br>➢ ImplementarJaccardcoefficient<br>➢ ImplementarFowlkesandMallowsindex<br>➢ ImplementarΓ_statistics |
| Validar un agrupamiento mediante la validación Interna. | ➢ Implementar la funcionalidad Centroides<br>➢ Implementar la funcionalidad min_value<br>➢ Implementar la funcionalidad max_value<br>➢ Implementar la funcionalidad suma_value<br>➢ Implementar la funcionalidad |

| | |
|---|---|
| | davies_bouldin |

A continuación se detallan cada una de las tareas relacionadas con la HU Validar un agrupamiento mediante el Criterio Externo:

| **Tarea de Ingeniería** | |
|---|---|
| **Número de la tarea:** 1 | **Número de Historia de Usuario**: 1 |
| **Nombre de la tarea**: Implementar la funcionalidad Matriz Arreglo. | |
| **Tipo de tarea:** Desarrollo | **Puntos Estimados**: 0.1 |
| **Fecha inicio**: 19/4/2013 | **Fecha fin**: 20/4/2013 |
| **Programador responsable:** Javier Sierra Ross | |
| **Descripción:** Se implementa una matriz triangular, para trabajar solamente con la parte inferior izquierda, por cuestiones de ahorro de espacio en memoria. | |

| **Tarea de Ingeniería** | |
|---|---|
| **Número de la tarea:** 2 | **Número de Historia de Usuario**: 1 |
| **Nombre de la tarea**: Implementar la funcionalidad Llenar Matriz. | |
| **Tipo de tarea:** Desarrollo | **Puntos Estimados**: 0.1 |
| **Fecha inicio**: 20/4/2013 | **Fecha fin**: 21/4/2013 |
| **Programador responsable:** Javier Sierra Ross | |
| **Descripción:** Se implementa una funcionalidad para llenar la matriz triangular con datos numéricos reales, los cuales provienen de un algoritmo de agrupamiento o de un conjunto de datos que se conozcan de antemano. | |

| **Tarea de Ingeniería** | |
|---|---|
| **Número de la tarea:** 3 | **Número de Historia de Usuario**: 1 |
| **Nombre de la tarea**: Implementar la funcionalidad Comparar Matrices. | |
| **Tipo de tarea:** Desarrollo | **Puntos Estimados**: 0.1 |
| **Fecha inicio**: 21/4/2013 | **Fecha fin**: 22/4/2013 |
| **Programador responsable:** Javier Sierra Ross | |

| |
|---|
| **Descripción:** Se implementa una funcionalidad que compara las matrices que resultan de un conjunto de datos que se conocen de antemano y otro que es el resultado de un algoritmo de agrupamiento y con la comparación se definen variables que servirán para calcular cada uno de los índices de validación. |

| **Tarea de Ingeniería** | |
|---|---|
| **Número de la tarea:** 4 | **Número de Historia de Usuario**: 1 |
| **Nombre de la tarea**: Implementar la funcionalidad índice de Rand. | |
| **Tipo de tarea:** Desarrollo | **Puntos Estimados**: 0.1 |
| **Fecha inicio**: 22/4/2013 | **Fecha fin**: 23/4/2013 |
| **Programador responsable:** Javier Sierra Ross | |
| **Descripción:** Se implementa una funcionalidad que calcula el índice de Rand el cual define en dependencia de su valor la calidad del agrupamiento realizado. | |

| **Tarea de Ingeniería** | |
|---|---|
| **Número de la tarea:** 5 | **Número de Historia de Usuario**: 1 |
| **Nombre de la tarea**: Implementar la funcionalidad coeficiente de Jaccard. | |
| **Tipo de tarea:** Desarrollo | **Puntos Estimados**: 0.1 |
| **Fecha inicio**: 23/4/2013 | **Fecha fin**: 23/4/2013 |
| **Programador responsable:** Javier Sierra Ross | |
| **Descripción:** Se implementa una funcionalidad que calcula el coeficiente de Jaccard, el cual define en dependencia de su valor la calidad del agrupamiento realizado. | |

| **Tarea de Ingeniería** | |
|---|---|
| **Número de la tarea:** 6 | **Número de Historia de Usuario**: 1 |
| **Nombre de la tarea**: Implementar la funcionalidad índice de Fowlkes. | |
| **Tipo de tarea:** Desarrollo | **Puntos Estimados**: 0.1 |
| **Fecha inicio**: 24/4/2013 | **Fecha fin**: 25/4/2013 |
| **Programador responsable:** Javier Sierra Ross | |
| **Descripción:** Se implementa una funcionalidad que calcula el índice de Fowlkes, el cual define en dependencia de su valor la calidad del agrupamiento realizado. | |

| Tarea de Ingeniería | |
|---|---|
| **Número de la tarea:** 7 | **Número de Historia de Usuario**: 1 |
| **Nombre de la tarea**: Implementar la funcionalidad índice de estadísticas. | |
| **Tipo de tarea:** Desarrollo | **Puntos Estimados**: 0.1 |
| **Fecha inicio**: 25/4/2013 | **Fecha fin**: 26/4/2013 |
| **Programador responsable:** Javier Sierra Ross | |
| **Descripción:** Se implementa una funcionalidad que calcula el índice de estadísticas, el cual define en dependencia de su valor la calidad del agrupamiento realizado. | |

A continuación se detallan cada una de las tareas relacionadas con la HU Validar un agrupamiento mediante el Criterio Relativo:

| Tarea de Ingeniería | |
|---|---|
| **Número de la tarea:** 1 | **Número de Historia de Usuario**: 2 |
| **Nombre de la tarea**: Implementar la funcionalidad centroides. | |
| **Tipo de tarea:** Desarrollo | **Puntos Estimados**: 0.1 |
| **Fecha inicio**: 26/4/2013 | **Fecha fin**: 27/4/2013 |
| **Programador responsable:** Javier Sierra Ross | |
| **Descripción:** Esta funcionalidad devuelve la distancia entre un objeto de un clúster y su centroide. | |

| Tarea de Ingeniería | |
|---|---|
| **Número de la tarea:** 2 | **Número de Historia de Usuario**: 2 |
| **Nombre de la tarea**: Implementar la funcionalidad min_value. | |
| **Tipo de tarea:** Desarrollo | **Puntos Estimados**: 0.1 |
| **Fecha inicio**: 27/4/2013 | **Fecha fin**: 28/4/2013 |
| **Programador responsable:** Javier Sierra Ross | |
| **Descripción:** Esta funcionalidad devuelve la distancia mínima entre dos puntos que se encuentren tanto en un mismo clúster como en clúster diferentes. | |

| Tarea de Ingeniería | |
|---|---|
| **Número de la tarea:** 3 | **Número de Historia de Usuario**: 2 |

| | |
|---|---|
| **Nombre de la tarea**: Implementar la funcionalidad max_value. | |
| **Tipo de tarea:** Desarrollo | **Puntos Estimados**: 0.1 |
| **Fecha inicio**: 28/4/2013 | **Fecha fin**: 29/4/2013 |
| **Programador responsable:** Javier Sierra Ross | |
| **Descripción:** Esta funcionalidad devuelve la distancia máxima entre dos puntos que se encuentren tanto en un mismo clúster como en clúster diferentes. | |

| **Tarea de Ingeniería** | |
|---|---|
| **Número de la tarea:** 4 | **Número de Historia de Usuario**: 2 |
| **Nombre de la tarea**: Implementar la funcionalidad suma_value. | |
| **Tipo de tarea:** Desarrollo | **Puntos Estimados**: 0.1 |
| **Fecha inicio**: 29/4/2013 | **Fecha fin**: 30/4/2013 |
| **Programador responsable:** Javier Sierra Ross | |
| **Descripción:** Esta funcionalidad devuelve la suma de todas las distancias entre los objetos de un clúster. | |

| **Tarea de Ingeniería** | |
|---|---|
| **Número de la tarea:** 7 | **Número de Historia de Usuario**: 2 |
| **Nombre de la tarea**: Implementar la funcionalidad davies_bouldin. | |
| **Tipo de tarea:** Desarrollo | **Puntos Estimados**: 0.1 |
| **Fecha inicio**: 02/5/2013 | **Fecha fin**: 03/5/2013 |
| **Programador responsable:** Javier Sierra Ross | |
| **Descripción:** Se implementa una funcionalidad que calcula el índice de David Bouldin, el cual define en dependencia de su valor la calidad del agrupamiento realizado. | |

## 2.7 Conclusiones del Capítulo

Como parte de la propuesta de solución, en este capítulo se describió el procedimiento llevado a cabo en la presente investigación para dar solución al problema de agrupamiento.

En el caso de la validación interna, se proponen algunas funcionalidades, que servirán para futuros cambios en el componente propuesto, estas brindan la posibilidad entre otras cosas de agregar un nuevo índice de validación de clúster interno, el cual será implementado sobre la base de estas funcionalidades.

Por su parte, en la validación externa se definen cuatro parámetros los cuales servirán como base de futuras implementaciones de nuevos índices de validación externa. Para así poder abarcar todas las posibles características que puedan tener las agrupaciones que resulten de los algoritmos de agrupamiento.

Como parte de la fase principal de la metodología XP, la fase Exploración es uno de los aspectos tratados en este capítulo. En este apartado, se definieron las HU, validar un agrupamiento mediante la validación externa, la cual brinda la posibilidad de evaluar la calidad de un agrupamiento, por medio de los índices de validación externa, la otra historia de usuario es validar un agrupamiento mediante la validación interna, la cual verifica cuan bueno es un agrupamiento a través de los índices de validación interna.

En cuanto a la fase de planificación, donde se decidió que el componente estará compuesto por una sola iteración, debido a que las HU tienen el mismo nivel de complejidad. Logrando así definir la fecha de entrega de este producto.

# CAPITULO 3.Validación y discusión de los resultados

## Introducción

En el presente capítulo se diseñan un conjunto de experimentos que permiten validar la solución propuesta. Como parte de la validación del funcionamiento del sistema, se proponen estrategias de pruebas, compuesta por pruebas unitarias y pruebas de aceptación, con el objetivo de validar el funcionamiento del componente. Adicionalmente se realizan experimentos para validar el funcionamiento de los métodos de validación de clúster propuestos a través de un caso de estudio. Se utilizaron una familia de algoritmos de agrupamiento jerárquicos, para obtener diferentes particiones, con múltiples valores del número de clúster sobre 2 bases de datos estándares (Iris, Glass). Por último se discuten los resultados obtenidos producto de este caso de estudio.

## 3.1 Pruebas de funcionamiento del sistema

Las Pruebas son actividades que se les realiza al sistema bajo condiciones específicas. Uno de los pilares de la Extreme Programming (XP) es el proceso de pruebas. XP anima a probar constantemente tanto como sea posible. Esto permite aumentar la calidad de los sistemas reduciendo el número de errores no detectados y disminuyendo el tiempo transcurrido entre la aparición de un error y su detección. También permite aumentar la seguridad de evitar efectos colaterales no deseados a la hora de realizar modificaciones y refactorizaciones. Las pruebas del sistema tienen como objetivo verificar la funcionalidad del sistema a través de sus interfaces externas, comprobando que dicha funcionalidad sea la esperada en función de los requisitos del sistema. El objetivo de las Pruebas es encontrar errores, una buena prueba es en la que se tiene una alta probabilidad de encontrar un error. Por lo tanto, cuando un ingeniero de software diseñe e implemente un sistema o producto de cómputo, debe tener en mente la facilidad de prueba. Al mismo tiempo, las propias pruebas deben mostrar un conjunto de características para alcanzar el objetivo de encontrar la mayor cantidad de errores con un mínimo de esfuerzo. (PRESSMAN_6TA_EDICIÓN.).

### 3.1.1 Estrategias de Prueba

XP divide las pruebas del sistema en dos grupos : pruebas unitarias, encargadas de verificar el código y diseñada por los programadores, y pruebas de aceptación o pruebas funcionales destinadas a evaluar si al final de una iteración se consiguió la funcionalidad requerida diseñadas por el cliente final.(J. J. GUTIÉRREZ).

#### 3.1.1.1 Pruebas Unitarias

Al desarrollar un nuevo software o sistema de información, la primera etapa de pruebas a considerar es la etapa de pruebas unitarias o también llamada pruebas de caja blanca. Las pruebas unitarias están relacionadas con el código y la responsabilidad de cada clase y sus fragmentos de código más críticos. Es la mejor forma de detectar errores tempranamente en el desarrollo, ayuda a definir los requerimientos y responsabilidades de cada método en cada clase probada, permite incluso hacer pruebas de estrés tempranamente en el código. Las pruebas unitarias son establecidas antes de escribir el código y son ejecutadas constantemente ante cada modificación del sistema. Los clientes escriben las pruebas funcionales para cada historia de usuario que deba validarse. En este contexto de desarrollo evolutivo y de énfasis en pruebas constantes, la automatización para apoyar esta actividad es crucial.(PATRICIO LETELIER 2006).

#### 3.1.1.2 Pruebas de Aceptación

Las pruebas de aceptación son más importantes que las pruebas unitarias, dado que significan la satisfacción del cliente con el producto desarrollado y el final de una iteración y el comienzo de la siguiente. Por esto, el cliente es la persona adecuada para diseñar las pruebas de aceptación. Estas pruebas son de caja negra, que representa un resultado esperado después de interactuar con la aplicación.

A continuación se representan mediante tablas las pruebas de aceptación realizadas a las HU.

**Historia de Usuario número 1.**

| **Caso de prueba de aceptación 1** |
|---|
| Historia de usuario: Validar un agrupamiento mediante la validación externa. |

| |
|---|
| Nombre: Implementar la funcionalidad “llenar matriz”. |
| Responsable: Javier Sierra Ross. |
| Descripción: Esta prueba se realiza con el objetivo de comprobar que se llene correctamente la matriz. |
| Condiciones de ejecución: Debe estar ejecutada opción seleccione el clustering priori de la aplicación. |
| Entradas/pasos de ejecución:<br>➢ Seleccionar la opción Validación de Clúster.<br>➢ Seleccionar la opción seleccione el clustering priori. |
| Resultado esperado: Los resultados de llenar la matriz deben verse en pantalla. |
| Evaluación de la prueba: Prueba satisfactoria. |

| **Caso de prueba de aceptación 2** |
|---|
| Historia de usuario: Validar un agrupamiento mediante la validación externa. |
| Nombre: Implementar la funcionalidad “comparar matrices”. |
| Responsable: Javier Sierra Ross. |
| Descripción: Esta prueba se realiza con el objetivo de comprobar que se comparen correctamente las matrices seleccionadas. |
| Condiciones de ejecución: Debe estar ejecutada opción comparar matrices de la aplicación. |
| Entradas/pasos de ejecución:<br>➢ Seleccionar la opción Validación de Clúster.<br>➢ Seleccionar la opción seleccione el clustering priori.<br>➢ Seleccionar la opción cargar BD.<br>➢ Seleccionar la opción comparar matrices. |
| Resultado esperado: Los resultados de comparar matrices deben mostrase en la aplicación. |
| Evaluación de la prueba: Prueba satisfactoria. |

| **Caso de prueba de aceptación 3** |
|---|
| Historia de usuario: Validar un agrupamiento mediante la validación externa. |
| Nombre: Implementar la funcionalidad “Índice de Rand”. |
| Responsable: Javier Sierra Ross. |
| Descripción: Esta prueba se realiza con el objetivo de comprobar que se calcule correctamente el índice de rand. |
| Condiciones de ejecución: Debe estar ejecutada opción Rand de la aplicación. |
| Entradas/pasos de ejecución:<br>➢ Seleccionar la opción Validación de Clúster.<br>➢ Seleccionar la opción seleccione el clustering priori.<br>➢ Seleccionar la opción cargar BD.<br>➢ Seleccionar la opción comparar matrices.<br>➢ Seleccionar la opción Rand. |
| Resultado esperado: Los resultados de índice de rand deben mostrase en la aplicación. |
| Evaluación de la prueba: Prueba satisfactoria. |
| **Caso de prueba de aceptación 4** |
| Historia de usuario: Validar un agrupamiento mediante la validación externa. |
| Nombre: Implementar la funcionalidad “jaccardcoeficient”. |
| Responsable: Javier Sierra Ross. |
| Descripción: Esta prueba se realiza con el objetivo de comprobar que se calcule correctamente el coeficiente de jaccard. |
| Condiciones de ejecución: Debe estar ejecutada opción Jaccard de la aplicación. |
| Entradas/pasos de ejecución:<br>➢ Seleccionar la opción Validación de Clúster.<br>➢ Seleccionar la opción seleccione el clustering priori.<br>➢ Seleccionar la opción cargar BD.<br>➢ Seleccionar la opción comparar matrices. |

| |
|---|
| ➢ Seleccionar la opción Jaccard. |
| Resultado esperado: Los resultados del coeficiente de jaccard deben mostrase en la aplicación. |
| Evaluación de la prueba: Prueba satisfactoria. |

| **Caso de prueba de aceptación 5** |
|---|
| Historia de usuario: Validar un agrupamiento mediante la validación externa. |
| Nombre: Implementar la funcionalidad “Fowlkes”. |
| Responsable: Javier Sierra Ross. |
| Descripción: Esta prueba se realiza con el objetivo de comprobar que se calcule correctamente el coeficiente de Fowlkes. |
| Condiciones de ejecución: Debe estar ejecutada opción Fowlkes de la aplicación. |
| Entradas/pasos de ejecución:<br>➢ Seleccionar la opción Validación de Clúster.<br>➢ Seleccionar la opción seleccione el clustering priori.<br>➢ Seleccionar la opción cargar BD.<br>➢ Seleccionar la opción comparar matrices.<br>➢ Seleccionar la opción Fowlkes. |
| Resultado esperado: Los resultados del índice Fowlkes deben mostrase en la aplicación. |
| Evaluación de la prueba: Prueba satisfactoria. |

| **Caso de prueba de aceptación 6** |
|---|
| Historia de usuario: Validar un agrupamiento mediante la validación externa. |
| Nombre: Implementar la funcionalidad “Fowlkes”. |
| Responsable: Javier Sierra Ross. |
| Descripción: Esta prueba se realiza con el objetivo de comprobar que se calcule correctamente el coeficiente de Fowlkes. |

| |
|---|
| Condiciones de ejecución: Debe estar ejecutada opción Fowlkes de la aplicación. |
| Entradas/pasos de ejecución:<br>➢ Seleccionar la opción Validación de Clúster.<br>➢ Seleccionar la opción seleccione el clustering priori.<br>➢ Seleccionar la opción cargar BD.<br>➢ Seleccionar la opción comparar matrices.<br>➢ Seleccionar la opción Fowlkes. |
| Resultado esperado: Los resultados del índice Fowlkes deben mostrase en la aplicación. |
| Evaluación de la prueba: Prueba satisfactoria. |

**Historia de Usuario número 2.**

| **Caso de prueba de aceptación 7** |
|---|
| Historia de usuario: Validar un agrupamiento mediante la validación interna. |
| Nombre: Implementar la funcionalidad “Davies Bouldin”. |
| Responsable: Javier Sierra Ross. |
| Descripción: Esta prueba se realiza con el objetivo de comprobar que se calcule correctamente el índice de Davies Bouldin. |
| Condiciones de ejecución: Debe estar ejecutada la opción cargar BD de la aplicación. |
| Entradas/pasos de ejecución:<br>➢ Seleccionar la opción Validación de Clúster.<br>➢ Seleccionar la opción cargar BD. |
| Resultado esperado: Los resultados del índice de Davies Bouldin deben mostrase en la aplicación. |
| Evaluación de la prueba: Prueba satisfactoria. |

| **Caso de prueba de aceptación 8** |
|---|
| Historia de usuario: |
| Nombre: Seleccionar el algoritmo de agrupamiento. |

| |
|---|
| Responsable: Javier Sierra Ross. |
| Descripción: Esta prueba se realiza con el objetivo de comprobar que se ejecute correctamente el algoritmo de agrupamiento seleccionado |
| Condiciones de ejecución: Debe estar ejecutada la opción Validación de Clúster. |
| Entradas/pasos de ejecución:<br>➢ Seleccionar la opción Validación de Clúster.<br>➢ Seleccionar el algoritmo de agrupamiento. |
| Resultado esperado: Debe realizarse la validación utilizando este algoritmo de agrupamiento. |
| Evaluación de la prueba: Prueba satisfactoria. |

## 3.2 Validación de los índices de validación de clúster.

Para validar el funcionamiento de los índices de validación de clúster, se propone la realización de experimentos que permitan ejecutar algoritmos de agrupamiento y validar el resultado de los mismos empleando los índices de validación de clúster propuesto. Se empleó la base de datos Iris, la cual ya se encuentra etiquetada y se conoce de antemano las clases a las que pertenece cada objeto. Esta es quizás la base de datos más conocida en la literatura de Reconocimiento de Patrones. El conjunto de datos consta de 3 clases de 50 ejemplos cada una, donde cada clase se refiere a un tipo de planta Iris. Una clase está separada linealmente de las otras dos, mientras que dos no son linealmente separables. Consta de cuatro atributos: longitud y ancho del sépalo de la flor y longitud y ancho de los pétalos en centímetros. Los tres tipos de planta Iris son: Iris Setosa, Iris Virginica e Iris Versicolor. La siguiente tabla detalla las características de esta base de dato.

| Base de datos | Cantidad de objetos | Cantidad de rasgos | Cantidad de clúster | Tipo de datos |
|---|---|---|---|---|
| IRIS | 150 | 4 | 3 | Real |

Como algoritmos de agrupamiento se emplean el single linkage (SL), Complete linkage (CL), Average linkage (AL) y Centroide (C). Para cada algoritmo de estos se calculan los valores de los índices de validación de clúster, tanto interno como externo. Estos valores son tomados para diferentes particiones (Número de clúster $k$), de modo que se pueda verificar cuál de las particiones es la mejor. Las siguientes tablas muestran los resultados obtenidos de estos experimentos.

| | k | SL | AL | CL | C |
|---|---|---|---|---|---|
| Davies Bouldin | 3 | **0.175** | **0.225** | **0.283** | 0.287 |
| | 4 | 0.389 | 0.290 | 0.286 | **0.264** |
| | 5 | 0.299 | 0.286 | 0.490 | 0.266 |
| | 6 | 0.302 | 0.395 | 0.555 | 0.299 |
| | 7 | 0.296 | 0.385 | 0.516 | 0.307 |

| | k | SL | AL | CL | C |
|---|---|---|---|---|---|
| RAND | 3 | 0.722 | 0.785 | **0.837** | **0.888** |
| | 4 | 0.723 | 0.818 | 0.821 | 0.877 |
| | 5 | **0.774** | 0.814 | 0.769 | 0.873 |
| | 6 | 0.774 | **0.832** | 0.785 | 0.847 |
| | **k** | **SL** | **AL** | **CL** | **C** |
| JACCARD | 3 | **0.578** | **0.565** | **0.616** | **0.710** |
| | 4 | 0.575 | 0.550 | 0.555 | 0.678 |
| | 5 | 0.563 | 0.539 | 0.424 | 0.667 |
| | 6 | 0.559 | 0.526 | 0.394 | 0.599 |
| | **k** | **SL** | **AL** | **CL** | **C** |
| FOWLKES | 3 | **0.756** | **0.733** | **0.764** | **0.831** |
| | 4 | 0.752 | 0.710 | 0.714 | 0.808 |
| | 5 | 0.736 | 0.701 | 0.560 | 0.800 |
| | 6 | 0.732 | 0.701 | 0.591 | 0.750 |
| | **k** | **SL** | **AL** | **CL** | **C** |
| ESTADISTICA | 3 | **0.608** | 0.577 | **0.642** | **0.748** |
| | 4 | 0.602 | 0.578 | 0.585 | 0.718 |
| | 5 | 0.578 | 0.566 | 0.433 | 0.707 |
| | 6 | 0.572 | **0.597** | 0.473 | 0.641 |

Como se aprecia en la tabla de resultados, la mayoría de los índices muestran la mejor partición para $k = 3$, lo cual coincide con el número de grupos de esta base de datos. Esta base de datos presenta solapamiento entre algunos objetos de las clases virginica y versicolor,y los métodos de agrupamiento jerárquicos no son capaces de separar completamente este solapamiento. Para el caso del método de agrupamiento del centroide, se obtuvo la mejor partición según Davies Bouldin para k = 4. Esto ocurre ya que para k =3 el método del centroide obtiene 35 y 63 elemento en los grupos de clases solapadas mientras que para k= 4 separa 4 elementos de la clase que tiene 63, de modo que la partición en general queda un poco más compacta. De igual manera para k=5 se separo 1 elemento de esta misma clase. En general se mantienen para este método 3 grandes grupos y elementos aislados que se pueden considerar como ruidos.

En el caso de los índices de validación de clúster externos se obtuvo la mejor partición para k = 6 en el caso del Average Linkage lo cual estuvo relacionado con el hecho de que no existen diferencias entre estas particiones, puesto que en todos los casos se obtienen 3 particiones grandes y pequeños grupos que pueden ser considerados como ruidos. En general se aprecia que la presencia de ruido no genera cambios significativos en los métodos de validación externa, pues no hay variaciones de estos índices que se consideren significativos. De los métodos de agrupamiento estudiados se obtiene, que los métodos CL y C determinan particiones más cercanas a la partición ideal, que los métodos SL y AL. Esto coincide con el hecho de que la base de datos Iris está ajustada al algoritmo de agrupamiento K-mean, el cual basa su funcionamiento en el uso del centroide. En el caso del algoritmo CL, tiende a generar grupos compactos, bastante similar a los que se obtienen del método del centroide. El algoritmos SL y en menor medida el AL, generan encadenamiento y los grupos que se forman en general tienden a ser más alargados, por lo que no están totalmente acorde con la naturaleza de los datos de esta base de datos.

## 3.3 Conclusiones del Capítulo

Con las pruebas de caja blanca se logró verificar que no quedaran errores, excepciones no manejadas, fuga de memoria, mala declaración de variables y llamadas repetidas a un método en el código fuente de la aplicación.

Se realizó un caso de estudio para evaluar el funcionamiento de los métodos de validación de clúster empleando una base de datos etiquetada. En general los resultados de este experimento mostraron como mejor partición para la base de datos Iris k = 3.

## CONCLUSIONES GENERALES:

El estudio de los métodos de validación de clúster permitió identificar dos enfoques esenciales para la validación de algoritmos de agrupamientos, validación interna y externa. De igual manera el estudio de los métodos de validación externa define varias métricas que pueden ser determinados a partir de la relación a pares de los elementos de una partición. Mientras que en los métodos de validación interna se identifican un grupo de funcionalidades que emplean la mayoría de estos índices para establecer las relaciones dentro de un grupo e inter grupos.

Se propone la implementación de un componente para la validación de clúster que tiene en cuenta los enfoques de validación interna y externa. Se logra una implementación para los índices de validación externa donde se definen los parámetros a, b, c, d que son la base para la incorporación de nuevas métricas. En el caso de los índices de validación interna se proponen un grupo de funcionalidades básicas que permitirán la construcción de nuevos índices de una manera sencilla. En general en la implementación se tuvo en cuenta estructuras de datos eficientes para el uso y manejo de la memoria.

El componente quedó validado tanto su funcionamiento como los métodos de validación de clúster implementados a través de una estrategia de pruebas y un caso de estudio respectivamente. De manera general se apreció que los métodos de validación interna y externa determinan como mejor partición para la base de datos Iris la partición k=3, lo cual coincide con las clases que posee esta base de datos. Se pudo apreciar además que los métodos agrupamientos propuestos así como los índices de validación de clúster presentan inestabilidad en los resultados cuando se tienen grupos considerados como ruidos.

## RECOMENDACIONES:

Se debe estudiar el problema de la estabilidad de los índices de validación de clúster en particiones con grupos que se consideren ruido. Como se puede apreciar en los resultados de la presente investigación este elemento vió afectado los resultados esperados de algunas particiones cuando la separa puntos aislados de los grupos.

Utilizar estos índices de validación de clúster para determinar automáticamente la mejor partición en los algoritmos de agrupamiento jerárquicos tomando en cuenta una estrategia de búsqueda eficiente sobre la jerarquía.

# BIBLIOGRAFÍA:


1. *AETECNO. El gran Universo Digital: la data crece más rápido de lo que podemos protegerla, 2013.*
2. *BARBONE. XP: Extreme Programming Montevideo Uruguay, 2004. [Disponible en: http://iie.fing.edu.uy/~nacho/blandos/seminario/XProg1.html.*
3. *BOUGUESSA, S. W. An objective approach to cluster validation., 2006. p.*
4. *COOPER., G. W. M. Y. M. C. An examination of procedures for determining the number of clusters in a data set., 1985. p.*
5. *DAVIES Y BOULDIN. A clustering separation measure. IEEE Transactions on Pattern Analysis and Machine Intelligence. 1979. p.*
6. *DOM, B. E. An information-theoretic external cluster-validity measure, 2001.*
7. *DUBES., J. Y. R. C. Algorithms for Clustering Data. Prentice-Hall. NJ, USA, 1988. p.*
8. *DUNN., J. C. A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters. Journal of Cybernetics, 1973.*
9. *ERIKA CAMACHO, F. C., GABRIEL NUÑEZ. ARQUITECTURAS DE SOFTWARE, Abril, 2004.*
10. *Espacios métricos, 2011.*
11. *FOWLER, M. La Nueva Metodología, 2006. [Disponible en: http://www.programacionextrema.org/articulos/newMethodology.es.html.*
12. *GOIKOETXEA, I. G. Aportaciones a la clasicación no supervisada y a su validación. Aplicación a la seguridad informática.*
13. *. Departamento de Arquitectura y Tecnología de Computadores. Donostia, 2010.220. p.*
14. *GONZÁLEZ, D. P. Algoritmos de Agrupamiento basados en densidad y Validación de clusters. Departament de Llenguatges I Sistemas Informátics Universitat Jaume I. Castellón, Marzo de 2010.183. p.*
15. *HARABASZ., T. C. Y. J. A dendrite method for cluster analysis. Communications in Statistics. 1974. p.*
16. *HERNÁNDEZ, J. Introducción a la Mineríade Datos., 2004. p.*
17. *J. J. GUTIÉRREZ, M. J. E., M. MEJÍAS, J. TORRES PRUEBAS DEL SISTEMA EN PROGRAMACIÓN EXTREMA*
18. *JAIN, A. K. Data clustering: 50 years beyond K-means. 2009a. p.*
19. *JAIN, A. K. Data clustering: 50 years beyond K-means., 2009b. p.*
20. *JAJODIA, D. B. Y. S. Applications of Data Mining in Computer Security., 2002.*

21. *JOSKOWICZ, J. Reglas y Prácticas en eXtreme Programming. España, Ingeniería Telemática de la Universidad de Vigo, España., 2008.22. p.*
22. *K. Y. YEUNG, D. R. H., Y W. L. RUZZO. Validating clustering for gene expression data. Bioinformatics. 2001. p.*
23. *KAMBER, J. H. Y. M. Data Mining: Concepts and Techniques. Morgan Kaufmann, segunda edición. 2006. p.*
24. *KETTENRING, J. R. The practice of cluster analysis. Journal of Classification, 2006.*
25. *KUO-LUNG WU, M.-S. Y., Y JUNE-NAN HSIEH. Robust cluster validity indexes., 2009. p.*
26. *KUO-LUNG WU, M.-S. Y., Y JUNE-NAN HSIEH. Robust cluster validity indexes., Marzo, 2009. p.*
27. *L. J. HUBERT Y J. R. LEVIN. A general statistical framework for assessing categorical clustering in free recall., 1976. p.*
28. *MALOOF, M. A. Machine Learning and Data Mining for Computer Security. London, 2006. p.*
29. *MARIA HALKIDI, Y. B., Y MICHALIS VAZIRGIANNIS. On clustering validation techniques. Journal of Intelligent Information Systems, 2001.*
30. *MURTAGH., F. The remarkable simplicity of very high dimensional data: Application of model-based clustering. Journal of Classification, 2009.*
31. *P. A. VIJAYA, M. N. M., Y D. K. SUBRAMANIAN. Efficient bottom up hybrid hierarchical clustering techniques for protein sequence classification., 2006. p.*
32. *PACCANARO, A. Spectral clustering of protein sequences., 2006. p.*
33. *PATRICIO LETELIER, M. C. P. Métodologías ágiles para el desarrollo de software: eXtreme Programming (XP), 2006.*
34. *PERONA, I. "SIHC: A Stable Incremental Hierarchical Clustering Algorithm". Milan, Italia, 2009. p.*
35. *PRESSMAN_6TA_EDICIÓN. Técnicas de Prueba del Sotware: 23.*
36. *RAMAKRISHNA, M. K. Y. R. S. New indices for cluster validity assessment., 2005. p.*
37. *RUI XIU AND DONALD C. WUNSCH, I. CLUSTERING. 2009. 364 p.*
38. *SIERRA, A. A. Programación Extrema y Software Libre, 2007. [Disponible en: http://www.seguridad.unam.mx/eventos/datos/ev11/semi18/mat.7.pon19.semi18.pdf.*
39. *SOKAL, P. H. A. S. Y. R. R. Numerical Taxonomy. Books in biology. W. H. Freeman and Company. San Francisco, 1973. p.*
40. *WILLIAMS., G. N. L. Y. W. T. A general theory of classificatory sorting strategies: II. Clustering systems. Computer Journal, 1967.*

41. *YEUNG, K. Y. H., D. R. AND RUZZO. Validating clustering for gene expression data. 2001. 309 – 318 p.*