Métodos de Inicialización para *k-means* Trabajo Final - Procesamiento

Cipolatti Edgardo, Rosales Mario y Santellán Franco
edgardocipolatti@hotmail.com - mariorosales941@gmail.com - fransantellan@gmail.com

***Resumen*— En el siguiente informe se comparan diferentes métodos de inicialización de centroides para el algoritmo *k-means*. Los métodos propuestos para tal fin son: BallHall, Etiquetado, Forgy, k-means++, McQueen y McRae. Luego su desempeño se observará con la ayuda de las siguientes medidas e índices: Iteraciones, Tiempo, Intra e Inter cluster, Davies Bouldin y Dunn.**

Se utilizarán diferentes bases de datos que varían en complejidad, según la cantidad de datos y dimensiones, como así también variando la cantidad de agrupamientos K solicitados.

Hacia el final de este informe se presentan gráficas que plasman los resultados. Los que fueron obtenidos realizando un número fijo de iteraciones para diferentes valores K de clusters solicitados.

***Palabras clave*— K-means, Clusters, McQueen, Etiquetado, McRae, Forgy, k-means++, Dunn, Davies-Bouldin,**

I. Introducción

El propósito de cualquier técnica de agrupamiento (*clustering*) es encontrar una matriz de partición $U(X)$ de Kxn de un conjunto de datos $X = \{x_1, x_2, \cdots, x_n\}$ en R^n, representando su partición en un número, digamos K, de racimos (*clusters*) $(C_1, C_2, \cdots, C_K)$. La matriz de partición $U(X)$ puede representarse como $U = [u_{kj}], k = 1, 2, \cdots, K$, y $j = 1, 2, \cdots n$, donde u_{kj} es la pertenencia del patrón x_j al cluster C_K. En la división de los datos, se cumple que: $u_{kj} = 1$ si $x_j \in C_K$; en otro caso, $u_{kj} = 0$. Las técnicas de clustering se dividen en dos clases, *Particionales* y *Jerárquicas*. *K-means* es una de las más usadas en los dominios de agrupación jerárquica.

Las dos preguntas fundamentales que deben abordarse en cualquier sistema de agrupamiento son: 1) ¿Cuántos grupos están realmente presentes en los datos? y 2) ¿Cuán real o bueno es el clustering en sí? Es decir, cualquiera sea el método de agrupamiento utilizado, uno tiene que determinar el número de grupos a formar y la bondad o validez de los conjuntos formados. La medida de validez de los clusters debe ser tal que sea capaz de imponer un ordenamiento de los clusters en términos de su bondad.

El clustering obtenido por el algoritmo *k-means* tiene la desventaja de que depende fuertemente de la inicialización de sus semillas. Es por esto que en este trabajo se pretende comparar diferentes métodos de inicialización utilizando las siguientes medidas e índices de validez: **Iteraciones**, **Tiempo**, **Intra** e **Inter-Cluster** y su relación **Intra/Inter**, el índice de **Davies-Bouldin** [5] y el índice de **Dunn** [6].

Los métodos de inicialización a comparar son: **Mc-Queen**, **Etiquetado**, **McRae**, **Forgy**, **Ball Hall** y **k-means++**. [7], [8], [9].

II. Métodos de Inicialización

A. BallHall

Ball y Hall proponen tomar el vector de medias de los datos como el primer punto semilla; posteriormente se seleccionan los restantes examinando los individuos sucesivamente, aceptando uno de ellos como siguiente punto semilla siempre y cuando esté, por lo menos, a alguna distancia, d, de todos los puntos elegidos anteriormente. Se continúa de esta forma hasta completar los K puntos deseados o el conjunto de datos se agota.

Aquí, d es tomado como la longitud entre el mínimo y máximo punto, dividida por la cantidad de clusters solicitados. Considerando como punto mínimo y máximo aquellos cuyas coordenadas se obtienen de buscar el mínimo y el máximo valor en cada dimensión.

Algoritmo 1: BallHall

```
Entradas: k, Data
Retorna : Matriz Seed de semillas
[n, m] = size(Data);
d = max(distancia entre puntos) / k;
Seed[1] = mean(Data);
ind = 2;
while ind < k+1 do
    for i ← 2 to n do
        if distancia(Data(i)) > d then
            Seed[ind] = Data(i);
            ind + 1;
        else
            continue;
        end
    end
end
```

B. Mc Queen

El método de McQueen se basa en elegir como semillas a los primeros K patrones del conjunto de datos. Se considera que la secuenciación en que se introducen los datos a la base de datos no influye en el resultado final.

Algoritmo 2: Mc Queen

```
Entradas: k, Data
Retorna : Matriz Seed de semillas
[n, m] = size(Data);
Seed[1:k] = Data(1:k);
```

C. Etiquetado

En este algoritmo se crean K índices donde cada uno de estos hace referencia a un patrón en la matriz *Data* original de n elementos. Dichos índices se conforman de la siguiente manera:

$$\left[\frac{\alpha \cdot m}{K}\right]$$

donde: $\alpha = 1, 2, \ldots, (K-1), K$; $m = 1, 2, \ldots, (n-1), n$ y $[x]$ representa la parte entera de x.

Luego, obtenidos los K índices, se toman como *Seed* aquellos patrones que se referencian en Data.

Algoritmo 3: Etiquetado

```
   Entradas: k, Data
   Retorna : Matriz Seed de semillas
1  [n, m] = size(Data);
2  for alpha ← 1 to k do
3  |  ind = ParteEntera(alpha*n/k);
4  |  Seed[alpha] = Data(ind);
5  end
```

D. Forgy

En el algoritmo de Forgy se forman K grupos mutuamente excluyentes de patrones seleccionados al azar de la matriz *Data*. Una vez conformados estos grupos, se le toma a cada uno el promedio y dicho resultado se utiliza como semilla.

Algoritmo 4: Forgy

```
   Entradas: k, Data
   Retorna : Matriz Seed de semillas
1  [n, m] = size(Data);
2  mezclar(Data);
3  p = round(n/k); ← patrones por cluster
4  c = 1; ind = 1;
5  while c < k+1 do
6  |  if (i+p-1) > n then
7  |  |  Seed[c,:] = mean(Data(i:end));
8  |  else
9  |  |  Seed[c,:] = mean(Data(i:(i+p-1)));
10 |  end
11 |  i = i+p;
12 |  c = c + 1;
13 end
```

E. k-means ++

El algoritmo de k-mean ++ consiste en encontrar semillas tal que se minimice la varianza entre grupos, es decir, minimizar la suma de las distancias al cuadrado de cada punto al centro mas cercano a él. Para realizar esto lo que se hace es seleccionar la primer semilla al azar entre todos los patrones de *Data*. Luego se le asigna una probabilidad a cada patrón, donde esta probabilidad consiste en el cuadrado de la menor distancia a una semilla sobre la suma de las distancias:

$$P(x) = \frac{D(x)^2}{\sum_{x \epsilon X} D(x)^2}$$

Dadas estas probabilidades por patrón, se selecciona un ganador y se repite hasta encontrar las K semillas.

Algoritmo 5: k-means ++

```
   Entradas: k, Data
   Retorna : Matriz Seed de semillas
1  [n, m] = size(Data);
2  Seed[1] = Data(rand(),:);
3  for i ← 2 to k do
4  |  for j ← 1 to n do
5  |  |  Prob(j) = min(distancia(Data(j),Seed));
6  |  end
7  |  Seed[i] = Data(ganador);
8  end
```

F. Mc Rae

Se etiquetan los patrones de 1 a m. Para obtener la primer semilla, se genera un número al azar entre 1 y m, dicho número indica el patrón seleccionado. Para la siguiente semilla, se repite el mismo procedimiento pero esta vez, generando un valor al azar entre 1 y $(m-1)$, debido a que ya se ha obtenido una semilla y la cantidad de patrones disminuye una unidad.

Algoritmo 6: Mc Rae

```
   Entradas: k, Data
   Retorna : Matriz Seed de semillas
1  [n, m] = size(Data);
2  c = 1;
3  while c < k+1 do
4  |  ind = rand();
5  |  Seed[c] = Data(ind);
6  |  Data = elimino(Data(ind));
7  |  c = c + 1;
8  end
```

III. Medidas e Índices de Validación de Clusters

Sea K el número de clusters que se quieren obtener de un conjunto de datos. Sea n la cantidad de patrones agrupados en un cluster. Sea $d()$ la medida de distancia utilizada (norma 2 en nuestro caso) y sea z el centroide de un dado cluster. Entonces, con el fin de establecer la bondad de los clusters encontrados por *k-means*, dependiendo de las distintas inicializaciones, se utiliza una serie de índices o parámetros que se describen a continuación:

A. Distancia Intra-Cluster

La distancia *Intra-Cluster* es una medida de cuán compacto es el cluster en cuestión. En otras palabras, este índice mide qué tan alejados están los patrones del cluster con

respecto a su centroide. La forma de obtener este valor es de la siguiente manera:

$$\sum_{i=1}^{n} \frac{d(x_i - z)}{n}$$

B. Distacia Inter-Cluster

La distancia *Inter-Cluster* es una medida de cuán dispersos están los clusters unos de otros. Es decir, este índice mide qué tan alejado está un cluster del resto de los clusters encontrados. Su valor puede calcularse como sigue:

$$\frac{\sum_{i=1}^{K-1} \sum_{j=i+1}^{K} d(z_i - z_j)}{K}$$

C. Índice Davies-Bouldin

Este índice esta definido como la razón entre la distancia *Intra* e *Inter-Cluster*, de la siguiente manera:

$$DB = \frac{1}{K} \sum_{i=1}^{K} R_{i,qt}$$

donde $R_{i,qt} = max_{i,j \neq i} \left\{ \frac{S_{i,q} - S_{j,q}}{d(z_i, z_j)} \right\}$. Aquí, S_i y S_j son las distancias *Intra-cluster* de los clusters i, j, y el denominador representa la distancia *Inter-Cluster*. El objetivo es minimizar este índice, ya que cuanto más compactos sean los clusters y más alejados estén, este índice es más chico.

D. Indice de Dunn

Sean S y T dos clusters no vacíos. El diámetro de un cluster se puede expresar como $\triangle(S) = max_{x,y \in S} d(x, y)$ y la distancia δ entre S y T es $\delta(S, T) = min_{x \in S, y \in T} d(x, y)$. Entonces el índice *Dunn* se define como sigue:

$$\nu_D = min_{1 \leq i \leq K} \left\{ min_{1 \leq j \leq K, j \neq i} \left\{ \frac{\delta(C_i, C_j)}{max_{1 \leq k \leq K}(\triangle(C_k))} \right\} \right\}$$

Valores grandes de ν_D corresponden a buenos clusters, y los valores de K que ayuden a maximizar este índice son la cantidad óptima de clusters para la base de datos.

E. Tiempo de Ejecución y Cantidad de Iteraciones

Si bien el Tiempo y las Iteraciones no describen la bondad de un agrupamiento, se pueden utilizar como otro parámetro de calificación de los métodos de inicialización. Estos parámetros son útiles ya que algunos pueden facilitar la tarea del algoritmo *k-menas*, o bien, dificultar la llegada del mismo a su convergencia. En otras palabras, cuanto mejor sean las semillas que recibe *k-means* menor trabajo le costará terminar su proceso y devolver los agrupamientos solicitados.

El tiempo de ejecución es tomado como el tiempo que lleva ejecutar el método de inicialización y *k-means*, mientras que para las iteraciones solo son tenidas en cuenta las que realiza este último.

IV. Desarrollo

A. Bases de Datos utilizadas

Para probar los métodos de inicialización en *k-means* se utilizaron los siguientes conjuntos de datos:

- **Iris**: 4 atributos, 3 clases y 150 elementos.
- **Nubes-10**: 2 atributos, 10 clases y 500 elementos.
- **Glass**: 10 atributos, 6 clases y 214 elementos.
- **Ionosphere**: 34 atributos, 2 clases y 351 elementos.
- **Doughnut**: 12 atributos, 2 clases y 500 elementos.
- **White Wine**: 11 atributos, 7 clases y 500 - 4897 elementos.

B. Obtención de Resultados

Para evaluar el desempeño de los métodos de inicialización y la bondad de los agrupamientos encontrados, se realizaron 30 iteraciones con cada método para valores de K que varían ente $K_{min} = 2$ y $K_{max} = 15$. A excepción de *White Wine* que por costo computacional se ejecutó con 10 iteraciones y $K_{min} = 5$ y $K_{max} = 10$.

A partir de los datos obtenidos, se realizó el promedio para cada índice (entre todas las iteraciones) para cada uno de los posibles valores de K. Debemos recordar que no solo evaluamos un método por su promedio en un índice, sino que también consideramos su desvío y su evolución. En cuanto a evolución, nos referimos al comportamiento del algoritmo a medida que la cantidad de clusters solicitados aumenta.

Luego de correr los algoritmos, los resultados obtenidos se muestran en el *Apéndice* (VI). El mismo contiene imágenes que ilustran el desempeño de los métodos de inicialización en relación a las medidas e índices anteriormente mencionados.

C. Resultados

En cuanto a las gráficas de los resultados sobre *Iris*, podemos ver que en la figura 1 los métodos arrojan en todos los índices resultados similares dado que el número de clusters solicitado es $K = 2$ y esto hace que *k.means*, sin importar las semillas que reciba, encuentre dos grupos bien definidos. Es por esto que las medidas de tiempo e iteración juegan un papel importante para la selección del método adecuado.

En la figura 2 el valor de $K = 3$ coincide con las clases de *Iris* y al mejor método lo define el índice *Dunn*, siendo este *BallHall*.

Al incrementar el número de clusters solicitados (K entre 4 y 15), nuevamente el índice *Dunn* refleja que el mejor método resulta ser *k.means++*. Como sustento de lo antedicho, se puede observar en la figura 4 la evolución de este índice en relación a K.

La siguiente base de datos analizada es *Nube10*. Dada la simplicidad de la estructura de los datos que la componen, todos los métodos devuelven buenos resultados. El índice *Dunn* es el único que brinda una distinción significativa

arrojando a *k-means++* como sobresaliente. En representación a los resultados para distintos valores de K, se muestra sólo la figura 5, con $K = 10$. En la figura 6 se muestra la relación *Intra/Inter* que ilustra que para cualquier K los métodos arrojan resultados igualmente aceptables.

Al analizar la base de datos *Glass* podemos ver que en las figuras 7 y 8 las medidas e índices a considerar son Iteraciones, Tiempo e índice *Dunn*, donde el peor método de inicialización es *BallHall* y el mejor es *k-means++*. Al observar la figura 9 notamos que *McQueen*, *McRae* y *k-means++* obtienen los mejores resultados en contraposición a *BallHall* y *Etiquetado*.

Para la base de datos *Ionosphere*, en la figura 12 podemos observar que hasta $K = 12$ el mejor método es *k-means++* con algunos sobrepasos de *Etiquetado*. Sin embargo, con $K = 14$ y $K = 15$ el método que devuelve mejores resultados es *BallHall*. En las figuras 10 y 11 se puede observar lo antedicho, pero se debe tener en cuenta que *BallHall* es el algoritmo que requiere más iteraciones.

Dada la disposición de los patrones en la base de datos *Doughnut* el algoritmo *k-means* falla. Esto ocurre debido a que las dos clases son concéntricas. En las figuras 13, 14 y 15 observamos que ningún índice es determinante para la selección de un método favorable, la única opción a tener en cuenta es el Tiempo y las Iteraciones.

La última base de datos analizada es *White Wine*, donde se ejecutaron los métodos con distinta cantidad de patrones con el propósito de analizar el tiempo de ejecución de los mismos en relación al crecimiento de la base de datos. En la figura 16 se puede observar cómo el tiempo crece considerablemente para la base de datos Wine de 4897 patrones y se logra ver que aquellos métodos que necesitan de cálculos sobre todos los patrones, BallHall y k-means ++, son fuertemente influenciados por el volumen de datos.

V. Conclusión

Los resultados expuestos en la sección anterior nos dejan ver que la elección del método para inicialización de *k-means* es totalmente dependiente de la cantidad K de clusters solicitados y de la base de datos bajo análisis. En bases de datos sencillas, como *Nubes10* o para valores de K bajos en cualquier base de datos, los índices de validación devuelven valores similares, quedando como únicas medidas significativas el Tiempo e Iteraciones. Por otro lado, en bases de datos complejas con mayor volumen de datos y dimensiones, los valores de los índices son inconclusos, y hasta poco intuitivos.

En conclusión, decimos que no es posible establecer una regla o un criterio que permita decidir estrictamente un método de inicialización o el valor optimo de K. En cada caso, la elección de la método de inicialización y el valor de K dependerá de la base de datos con la que se trabaje. Sin embargo, a lo largo del trabajo pudimos notar que aquellos métodos que realizan un análisis basado en distancias para la elección de las semillas, brindan mejores inicializaciones.

Referencias

[1] G. Milligan y C. Cooper, “An examination of procedures for determining the number of clusters in a data set,” *Psychometrika*, vol. 50, pp. 159–179, 1985.

[2] M. Meila y D. Heckerman, “An experimental comparison of several clustering and initialization methods,” *Proc. 14th Conf. Uncertainty in Artificial Intelligence*, pp. 386–395, 1998.

[3] C. Fraley y A. Raftery, “How many clusters? which clustering method? answers via model-based cluster analysis,” *The Computer J.*, vol. 41, pp. 578–588, 1998.

[4] L. Hall, I. Ozyurt, y J. C. Bezdek, “Clustering with a genetically optimized approach,” *IEEE Trans. Evolutionary Computation*, vol. 3, pp. 103–112, 1999.

[5] D. Davies y D. Bouldin, “A cluster separation measure,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 1, pp. 224–227, 1979.

[6] J. Dunn, “A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters,” *J. Cybernetics*, vol. 3, pp. 32–57, 1973.

[7] *Metodos no Jerarquicos de Analisis de Cluster*. Universidad de Granada, 2014, capÃtulo 4.

[8] S. Vassilvitski, *k-means++: The Advantages of Careful Seeding*. Standar Theory Groups, 2007.

[9] P. L. anda I. Inza y A. Moujahid, *Tema 14. Clustering*. Departamento de Ciencias de la Computacion e Inteligencia Artificial - Universidad del PaÃs Vasco, 2014.

VI. GRÁFICAS

A. *Gráficas para la base de datos Iris*

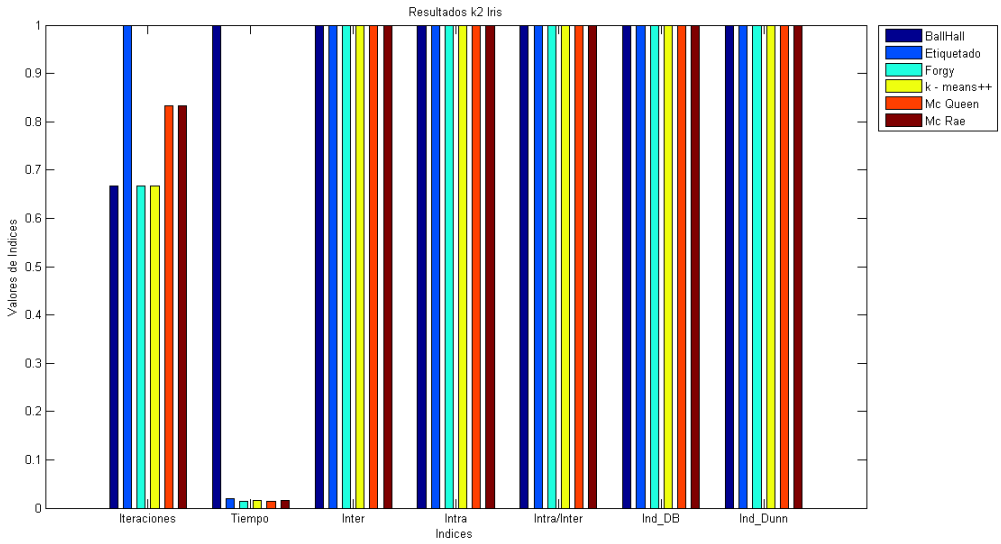

Fig. 1
IRIS CON $K = 2$

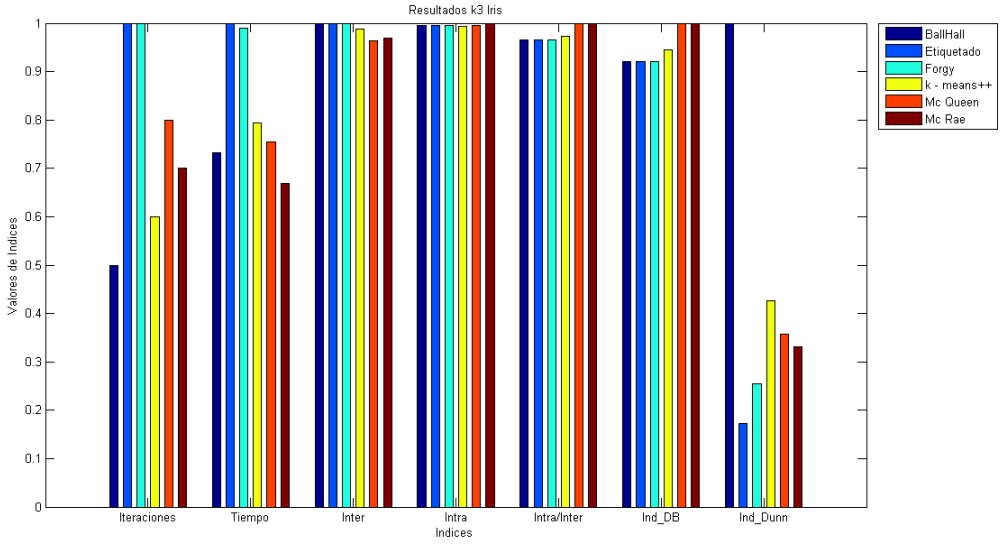

Fig. 2
IRIS CON $K = 3$

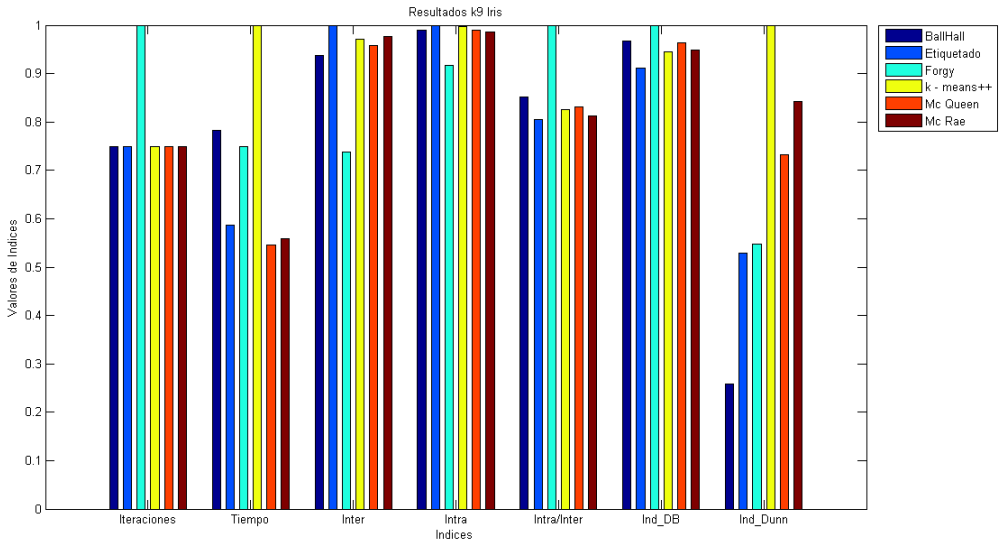

Fig. 3
IRIS CON $K = 9$

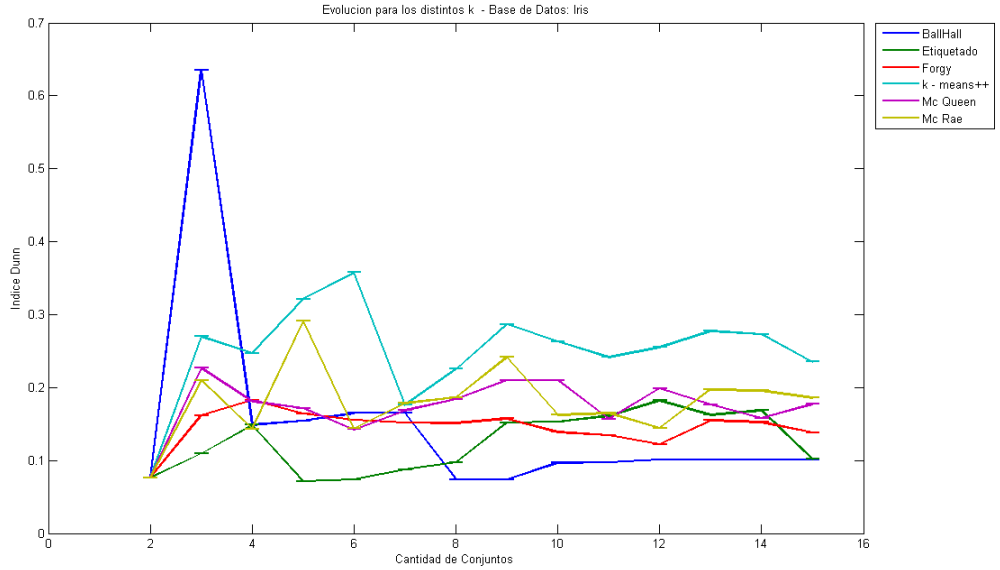

Fig. 4
ÍNDICE DUNN EN IRIS

B. Gráficas para la base de datos Nubes 10

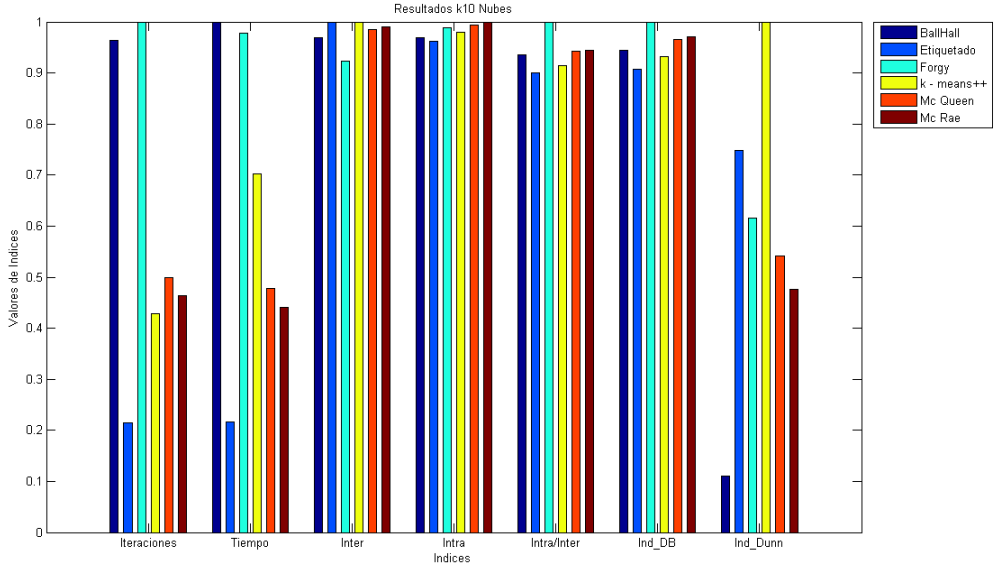

Fig. 5
NUBES 10 CON $K = 10$.

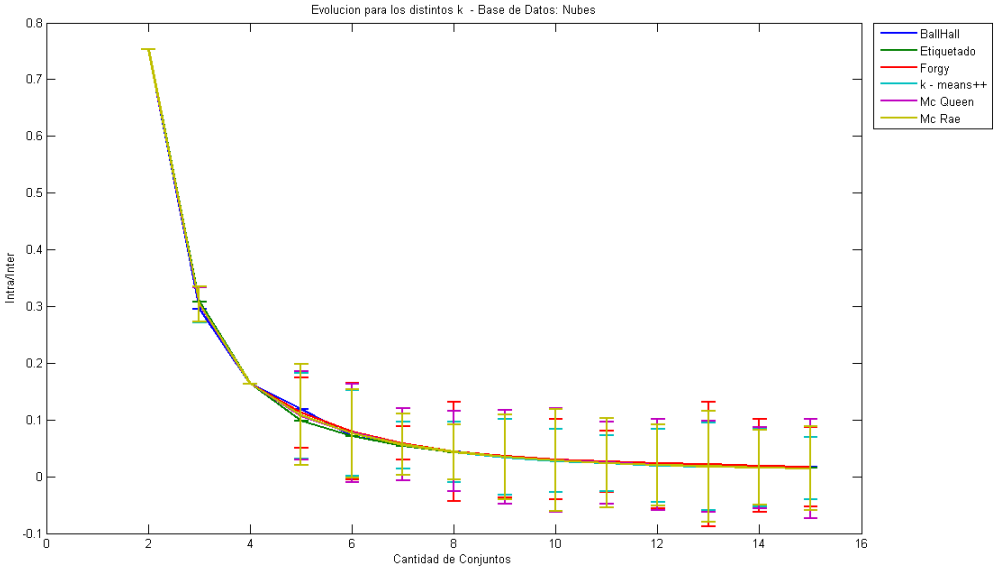

Fig. 6
ÍNDICE INTRA/INTER.

C. *Gráficas para la base de datos Glass*

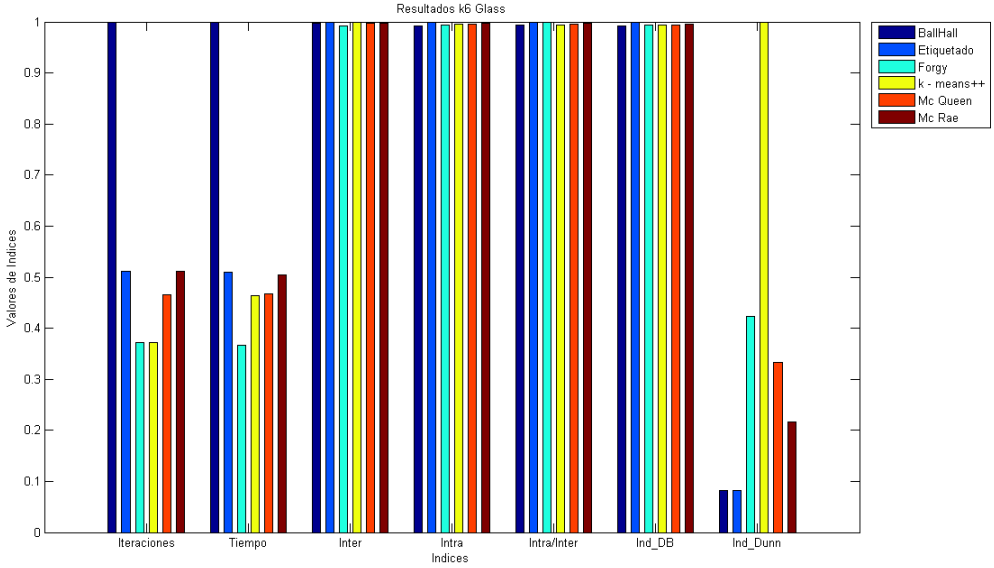

Fig. 7
GLASS CON $K = 6$.

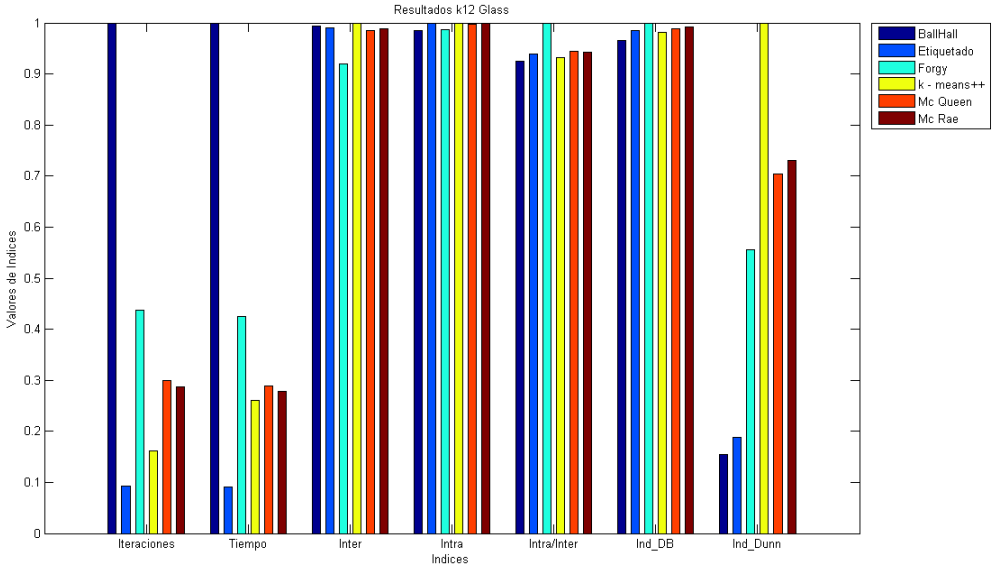

Fig. 8
GLASS CON $K = 12$.

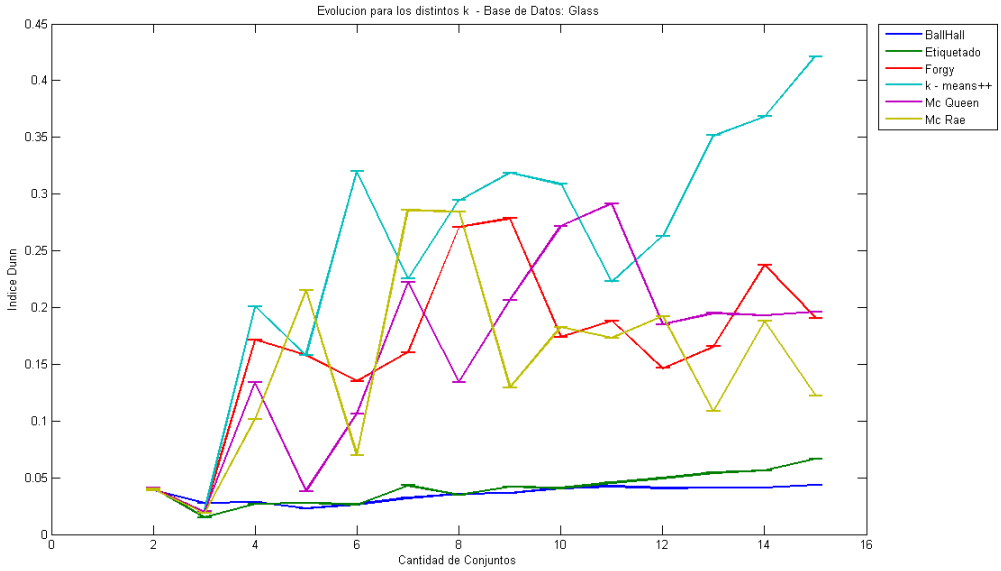

Fig. 9
ÍNDICE DUNN EN GLASS.

D. Gráficas para la base de datos Ionosphere

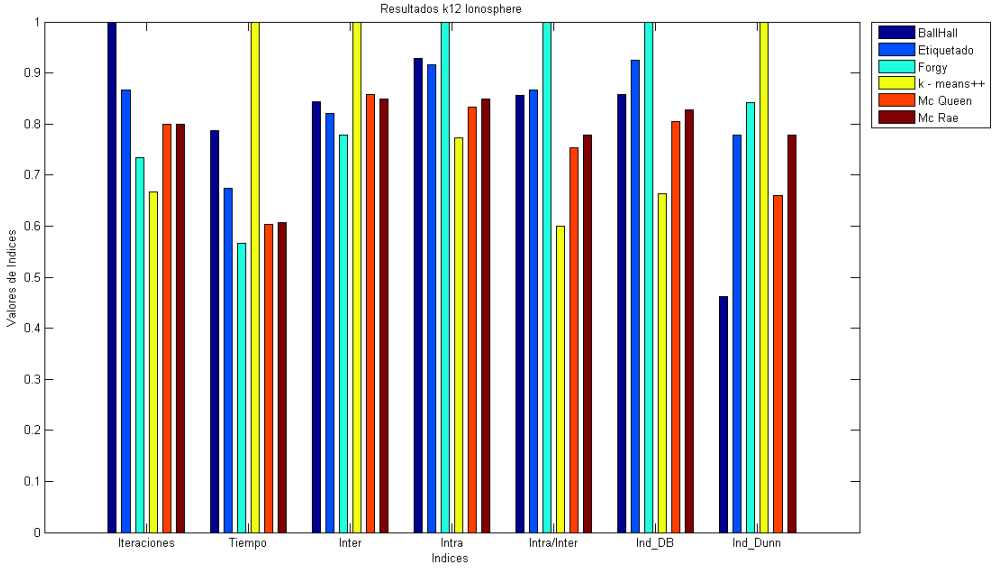

Fig. 10
IONOSPHERE CON $K = 12$.

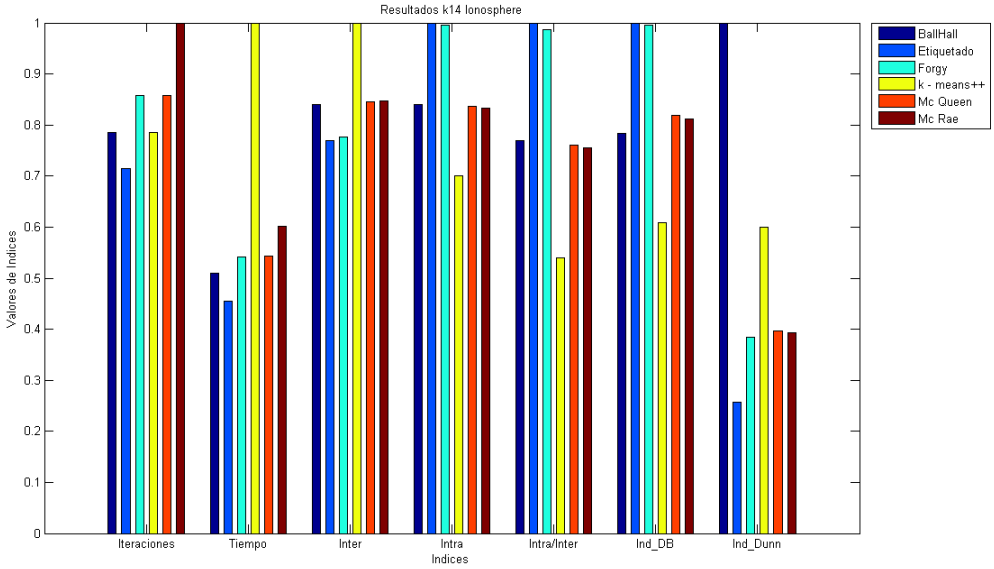

Fig. 11
IONOSPHERE CON $K = 14$.

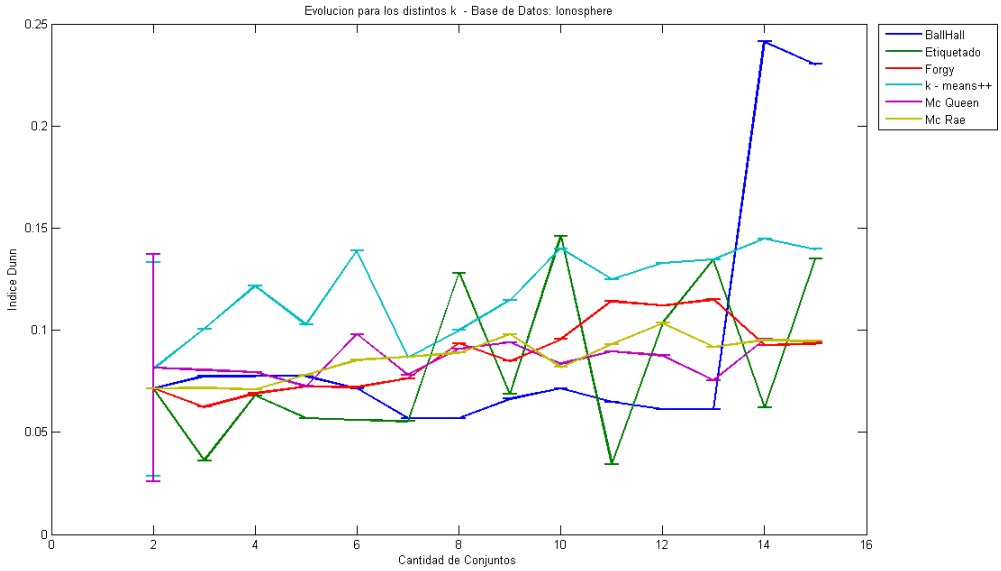

Fig. 12
ÍNDICE DUNN EN IONOSPHERE.

E. Gráficas para la base de datos Doughnut

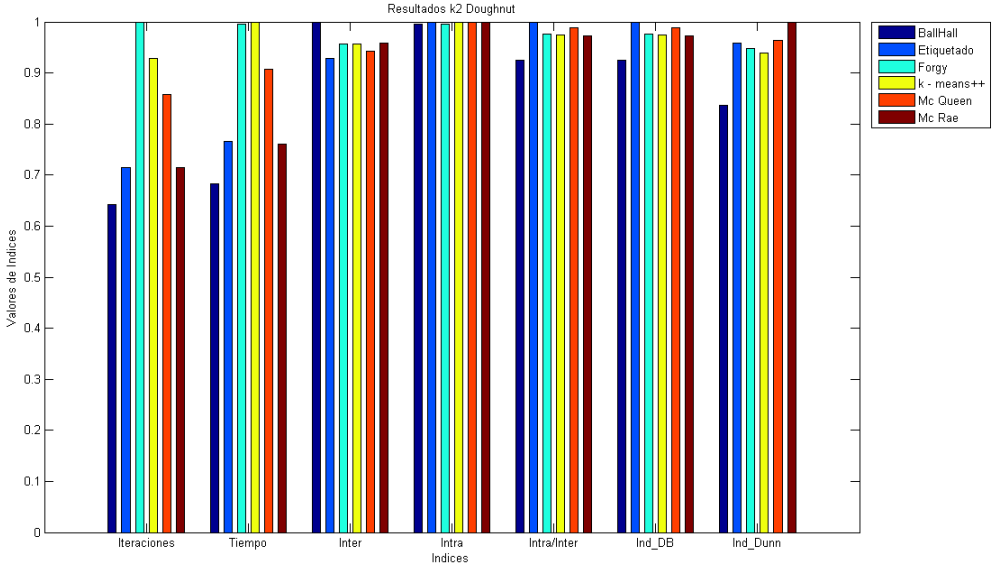

Fig. 13
DOUGHNUT CON $K = 2$.

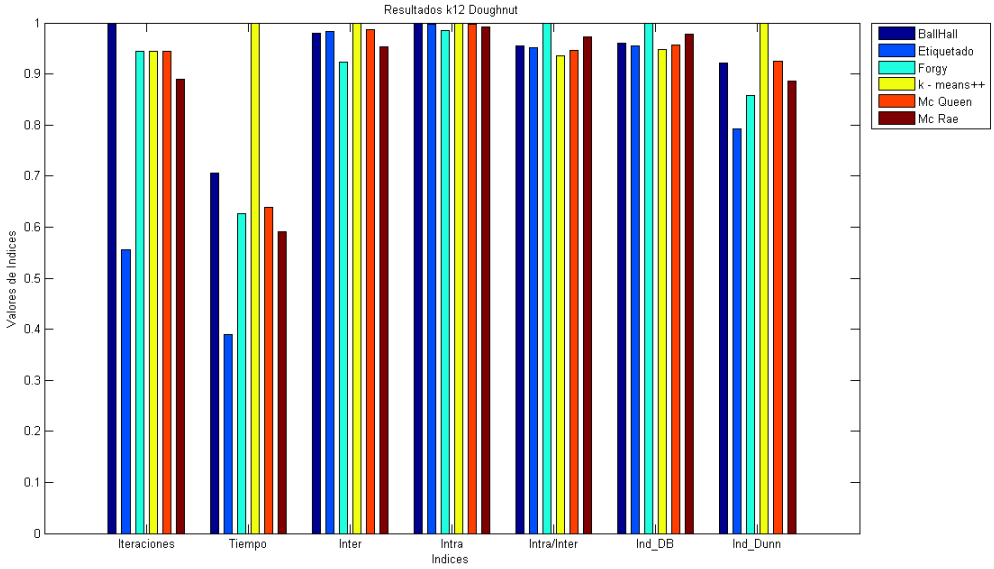

Fig. 14
DOUGHNUT CON $K = 12$.

F. Gráficas para la base de datos Wine

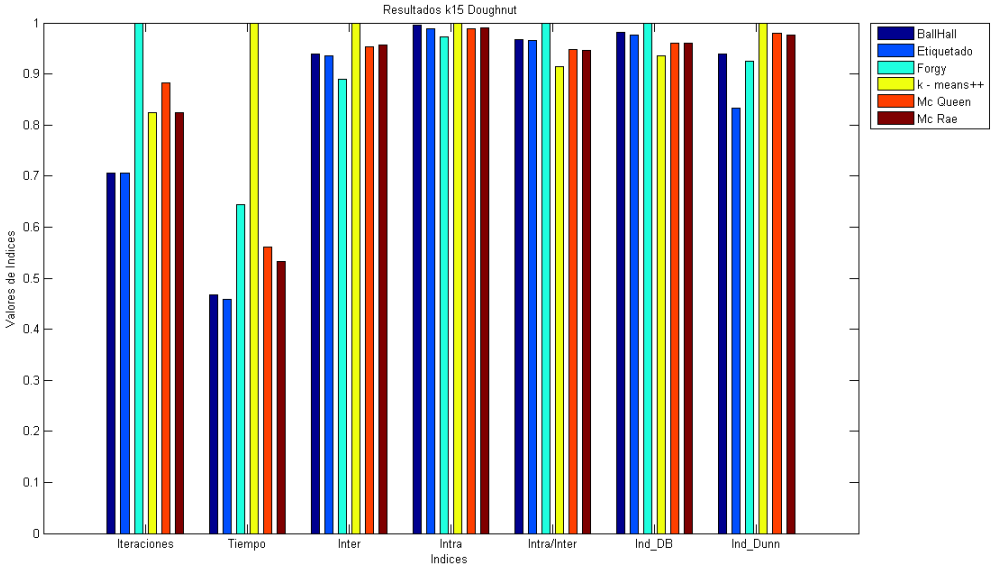

Fig. 15
DOUGHNUT CON $K = 15$.

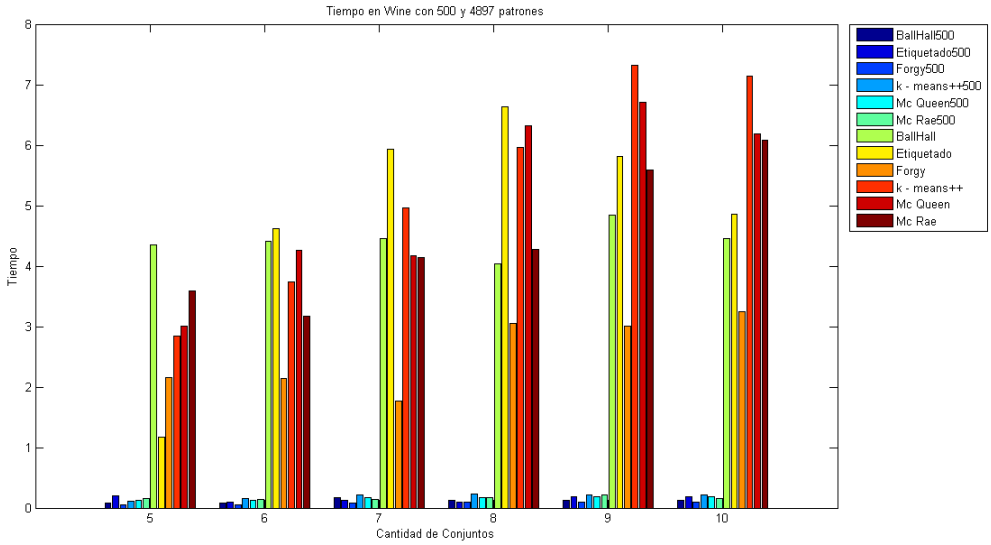

Fig. 16
TIEMPO DE WINE CON 500 Y 4897 PATRONES