

Universidad Autónoma Metropolitana Unidad Azcapotzalco

División de Ciencias Básicas en Ingeniería

Licenciatura en Ingeniería en Computación

Proyecto Tecnológico

Detección acústica de disparos de armas de fuego usando técnicas de minería datos


Trimestre 2014 Otoño

Christian Josué Manrique Mejía
Matricula: 206241151
christian.josue.manrique.mejia@gmail.com


Asesora: Dra. Silvia Beatriz González
Brambila
Profesora Titular C
Departamento de Sistemas
Núm.Ec.: 17204
sgb@correo.azc.uam.mx

Co-asesor: Dr. Juan Gaspar Vargas
Rubio
Profesor Titular B
Departamento de Electrónica
Núm.Ec.: 2011
jgvr@correo.azc.uam.mx


Yo, Dra. Silvia Beatriz González Brambila, declaro que aprobé el contenido del presente Reporte de Proyecto de Integración y doy mi autorización para su publicación en la Biblioteca Digital, así como en el Repositorio Institucional de UAM Azcapotzalco.


Dra. Silvia Beatriz González Brambila

Yo, Dr. Juan Gaspar Vargas Rubio, declaro que aprobé el contenido del presente Reporte de Proyecto de Integración y doy mi autorización para su publicación en la Biblioteca Digital, así como en el Repositorio Institucional de UAM Azcapotzalco.


Dr. Juan Gaspar Vargas Rubio

Yo, Christian Josué Manrique Mejía, doy mi autorización a la Coordinación de Servicios de Información de la Universidad Autónoma Metropolitana, Unidad Azcapotzalco para publicar el presente documento en la Biblioteca Digital, así como en el Repositorio Institucional de la Uam Azcapotzalco


Christian Josué Manrique Mejía

Resumen

La minería de datos tiene como objetivo la búsqueda y reconocimiento de patrones repetitivos como también encontrar tendencias en los datos.

La implementación de técnicas de minería de datos sobre el reconocimiento de patrones acústicos se ha desarrollado en áreas como el reconocimiento del habla, de locutor, detección de sonidos de disparo, entre otros.

*En este proyecto se trata de clasificar y distinguir un sonido de disparo entre otros sonidos similares, a través de la implementación de métodos de **clustering**: "K-Means", "K-Medoids" y Jerárquico; clasificación supervisada: Máquinas de soporte vectorial ("SVM" del inglés, "**Support Vector Machines**"); y por último el método de **clustering** jerárquico en conjunto con el "DTW" del inglés "**Dynamic Time Warping**" como distancia de similitud.*

En principio se recolectaron 150 videos con presencia de sonidos de disparo producidos por: arma de fuego, arma de fogueo (arma detonadora de balas de salva), efectos de sonido y explosivos. En particular para estos tres sonidos (arma de fuego, arma de fogueo y efecto de sonido), solo se tomó el sonido de disparo que producía un solo tipo arma (Prieto beretta 9mm). Los sonidos tomados de estos videos fueron en ambientes abiertos y cerrados. Posteriormente extrayendo el audio de cada uno de los videos y tomando características de la frecuencia usando la transformada rápida de Fourier, promediando por bandas de frecuencia en escala logarítmica y de 50 en 50 Hz.

*Por último, se aplicaron los métodos de minería de datos sobre las características extraídas, en donde, los métodos de **clustering** no obtuvieron un buen desempeño, en cuanto a reconocer el sonido de disparo real de entre los otros sonidos producidos por: Disparo de salva, explosivos y efectos de sonido. Caso contrario, al usar las máquinas de soporte vectorial (SVM, del inglés **Support Vector Machines**) se logró reconocer un porcentaje significativo de los sonidos de disparos reales, así también reconociendo los demás sonidos producidos por: Disparos de salva, efectos de sonido y explosivos.*

Tabla de contenido

RESUMEN	1
CAPÍTULO 1 INTRODUCCIÓN	15
1.1 ANTECEDENTES.....	15
1.1.1 <i>Trabajos internos</i>	15
1.2 JUSTIFICACIÓN.....	18
1.3 OBJETIVOS.....	19
1.3.1 <i>Objetivo general</i>	19
1.3.2 <i>Objetivos específicos</i>	19
CAPÍTULO 2 MARCO TEÓRICO	21
2.1 PROCESAMIENTO DE SEÑALES.....	21
2.1.1 <i>Transformada rápida de Fourier</i>	21
2.1.2 <i>Normalización de señales</i>	28
2.1.3 <i>Escala logarítmica</i>	28
2.2 TÉCNICAS DE MINERÍA DE DATOS	28
2.2.1 <i>Algoritmos de agrupamiento (clustering)</i>	28
2.2.1.1 Grupo (Cluster).....	29
2.2.1.2 Agrupamiento (Clustering)	29
2.2.1.3 K-medias (K-Means)	30
2.2.1.4 Jerárquico (Hierarchical).....	31
2.2.1.5 K-medoides (K-Medoids).....	31
2.2.2 <i>Alineamiento temporal dinámico (DTW, del inglés Dynamic Time Warping)</i>	32
2.2.3 <i>Máquinas de soporte vectorial (SVM, del inglés Support Vector Machines)</i>	35
2.3 RECAPITULACIÓN.....	37
CAPÍTULO 3 DESARROLLO DEL PROYECTO	39
3.1 DISEÑO	39
3.2 RECOLECCIÓN Y CLASIFICACIÓN DE VIDEOS.....	39
3.4 EXTRACCIÓN DEL AUDIO DE LOS VIDEOS	40
3.5 OBTENCIÓN DE LOS PARÁMETROS DEL AUDIO.....	40
3.6 PRE-PROCESAMIENTO.....	43
3.7 ALMACENAMIENTO EN ARCHIVOS DE TIPO TEXTO.....	44
3.8 APLICACIÓN DE LAS TÉCNICAS DE MINERÍA DE DATOS USANDO R.....	44
3.9 RECAPITULACIÓN.....	46
CAPÍTULO 4 RESULTADOS.....	47
4.1 RESULTADOS OBTENIDOS AL USAR EL PAQUETE clVALID DE R	47
4.1.1 <i>Resultados obtenidos al implementar las medidas de validación interna a través del paquete clValid</i>	47

4.2 RESULTADOS OBTENIDOS AL UTILIZAR LOS ALGORITMOS DE AGRUPAMIENTO (CLUSTERING) DE R.....	52
4.2.1 Resultados obtenidos al aplicar k-Means del paquete R.....	52
4.2.2 Resultados obtenidos al usar el algoritmo K-Medoids del paquete R.....	53
4.2.3 Resultados obtenidos al utilizar el algoritmo Hierarchical del paquete R.....	55
4.2.4 Resultados obtenidos al utilizar el algoritmo Hierarchical con distancia DTW del paquete R.....	59
4.3 RESULTADOS OBTENIDOS AL UTILIZAR EL ALGORITMO MÁQUINAS DE SOPORTE VECTORIAL (SUPPORT VECTOR MACHINES, POR SUS SIGLAS EN INGLÉS SVM) DEL PAQUETE R.....	61
4.4 RECAPITULACIÓN.....	63
CAPÍTULO 5 ANÁLISIS Y DISCUSIÓN DE RESULTADOS.....	65
5.1 COMPARACIÓN DE RESULTADOS.....	65
CAPÍTULO 6 CONCLUSIONES Y TRABAJOS FUTUROS.....	73
REFERENCIAS BIBLIOGRÁFICAS.....	75
APÉNDICE A	CÓDIGOS EN C++..... 78
A.1 CÓDIGO EN C ++ PARA LA OBTENCIÓN DE FRECUENCIAS EN ESCALA LOGARÍTMICA.....	78
A.2 CÓDIGO EN C ++ PARA LA OBTENCIÓN DE FRECUENCIAS POR BANDAS DE FRECUENCIA DE 50 EN 50 HZ.....	81
APÉNDICE B	CÓDIGOS EN R..... 84
B.1 ALGORITMOS DE CLUSTERING.....	84
B.1.1 K-Means.....	84
B.1.3 Método Jerárquico.....	89
B.1.4 Jerárquico con distancia "DTW".....	93
B.2 CLASIFICACIÓN CON SVM DEL INGLÉS SUPPORT VECTOR MACHINES.....	98
B.2.1 SVM con kernel lineal.....	98
B.2.2 SVM con kernel polynomial.....	98
B.2.3 SVM con kernel radial.....	99
B.2.4 SVM con kernel sigmoidal.....	99
B.3 VALIDACIÓN INTERNA.....	99
B.4 LECTURA DE ARCHIVOS.....	100
B.4.1 Lectura del archivo con 280 observaciones.....	100
B.4.1 Lectura del archivo con 301 observaciones.....	100
APÉNDICE C	RESULTADOS OBTENIDOS POR EL MÉTODO
K-MEANS	103
C.1. RESULTADOS OBTENIDOS AL IMPLEMENTAR K-MEANS SOBRE LAS 280 OBSERVACIONES POR BANDAS DE FRECUENCIA PROMEDIADAS EN ESCALA LOGARÍTMICA.....	103
C.2 RESULTADOS OBTENIDOS AL IMPLEMENTAR K-MEANS SOBRE 301 OBSERVACIONES PROMEDIADAS POR BANDAS DE FRECUENCIA DE 50 EN 50 HZ.....	139
APÉNDICE D	RESULTADOS
OBTENIDOS POR EL MÉTODO K-MEDOIDS.....	172
D.1 RESULTADOS OBTENIDOS AL IMPLEMENTAR K-MEDOIDS SOBRE LAS 280 OBSERVACIONES POR BANDAS DE FRECUENCIA PROMEDIADAS EN ESCALA LOGARÍTMICA.....	172

D.2 RESULTADOS OBTENIDOS AL IMPLEMENTAR K-MEDOIDS SOBRE 301 OBSERVACIONES PROMEDIADAS POR BANDAS DE FRECUENCIA DE 50 EN 50 HZ.....	190
APÉNDICE E	RESULTADOS
OBTENIDOS AL IMPLEMENTAR EL MÉTODO JERÁRQUICO	208
E.1 RESULTADOS OBTENIDOS AL IMPLEMENTAR JERÁRQUICO SOBRE LAS 280 OBSERVACIONES POR BANDAS DE FRECUENCIA PROMEDIADAS EN ESCALA LOGARÍTMICA.....	208
E.2 RESULTADOS OBTENIDOS AL IMPLEMENTAR JERÁRQUICO SOBRE 301 OBSERVACIONES PROMEDIADAS POR BANDAS DE FRECUENCIA DE 50 EN 50 HZ.....	226
APÉNDICE F	RESULTADOS
OBTENIDOS AL IMPLEMENTAR EL MÉTODO JERÁRQUICO CON DISTANCIA SIMILARIDAD “DTW” DEL INGLÉS “DYNAMIC TIME WARPING”	245
F.1 RESULTADOS OBTENIDOS AL IMPLEMENTAR EL MÉTODO JERÁRQUICO CON DISTANCIA “DTW” SOBRE LAS 280 OBSERVACIONES PROMEDIADAS EN ESCALA LOGARÍTMICA.....	245
F.2 RESULTADOS OBTENIDOS AL IMPLEMENTAR EL MÉTODO JERÁRQUICO CON DISTANCIA “DTW” SOBRE 301 OBSERVACIONES PROMEDIADAS POR BANDAS DE FRECUENCIA DE 50 EN 50 HZ.....	263
APÉNDICE G	RESULTADOS
OBTENIDOS AL IMPLEMENTAR EL MÉTODO “SVM” DEL INGLÉS “SUPPORT VECTOR MACHINES”	281
G.1 RESULTADOS OBTENIDOS AL IMPLEMENTAR EL MÉTODO “SVM” SOBRE 280 OBSERVACIONES PROMEDIADAS EN ESCALA LOGARÍTMICA	281
G.2 RESULTADOS OBTENIDOS AL IMPLEMENTAR EL MÉTODO “SVM” SOBRE 301 OBSERVACIONES PROMEDIADAS POR BANDAS DE FRECUENCIA DE 50 EN 50 HZ.....	285

Índice de figuras

FIGURA 2.1: COMPORTAMIENTO DE LA TRANSFORMADA DE FOURIER POR SUS SIGLAS EN INGLÉS FFT. FUENTE [25]	22
FIGURA 2.2: LOS ELEMENTOS COMPUTACIONALES DE LA TRANSFORMADA RÁPIDA DE FOURIER ES LA MARIPOSA. TOMA DOS NÚMEROS COMPLEJOS. REPRESENTADOS POR A Y B, Y FORMA LAS CANTIDADES MOSTRADAS. CADA MARIPOSA REQUIERE UNA MULTIPLICACIÓN COMPLEJA Y DOS SUMAS COMPLEJAS. FUENTE [26].	25
FIGURA 2.3: EL NÚMERO TOTAL DE CÁLCULOS QUE SE REALIZADOS SERÁN $N=8$ PARA CADA ETAPA Y $\log_2 N=3$ ETAPAS, HACIENDO EL NUMERO DE PROCESOS BÁSICOS DE $(8)(3)=N \log_2 N$. FUENTE [26]	27
FIGURA 2.4: MISMA SEÑAL CON VARIACIÓN EN TIEMPO Y VELOCIDAD	32
FIGURA 2.5: ENCONTRANDO PUNTOS DE SIMILITUD ENTRE AMBAS SEÑALES	33
FIGURA 3.1: DIAGRAMA DE BLOQUES DEL SISTEMA “DETECCIÓN ACÚSTICA DE DISPAROS DE ARMAS DE FUEGO”	39
FIGURA 3.2: PISTOLA “BERETTA” 9MM. FUENTE [28]	40
FIGURA 3.3: SEÑAL DE SONIDO DE DISPARO PRODUCIDO POR UN ARMA DE FUEGO REAL Y LA APLICACIÓN DE LA TRANSFORMADA DE FOURIER (FFT) SOBRE LA SEÑAL DE SONIDO	41
FIGURA 3.4: SEÑAL DE SONIDO DE DISPARO PRODUCIDO POR UN ARMA DE FOGUEO Y LA APLICACIÓN FFT SOBRE LA SEÑAL DE SONIDO	42
FIGURA 3.5: SEÑAL DE SONIDO PRODUCIDO POR EFECTOS DE SONIDO Y LA APLICACIÓN DE LA FFT SOBRE LA SEÑAL DEL SONIDO	42
FIGURA 3.6: SEÑAL DE SONIDO PRODUCIDO POR SONIDO DE UN EXPLOSIVO Y LA APLICACIÓN DE LA FFT SOBRE LA SEÑAL DE SONIDO	43
FIGURA 4.1: APLICANDO VALIDACIÓN INTERNA SOBRE LOS ALGORITMOS DE “CLUSTERING”	48
FIGURA 4.2: MEDIDA INTERNA “CONNECTIVITY”	49
FIGURA 4.3: MEDIDA INTERNA “DUNN”	50
FIGURA 4.4: MEDIDA INTERNA “SILHOUETTE”	51
FIGURA 4.5: K-MEANS CON $k=10$, ITERACIÓN 1	53
FIGURA 4.6: DENDOGRAMA DEL MÉTODO JERÁRQUICO AL PARTICIONAR CON 10 CLUSTERS	56
FIGURA 4.7: ÍNDICE <i>SILHOUETTE</i> DEL MÉTODO JERÁRQUICO AL PARTICIONAR CON 10 CLUSTERS	56
FIGURA 4.8: DENDOGRAMA DEL MÉTODO JERÁRQUICO AL PARTICIONAR CON 10 CLUSTERS	57
FIGURA 4.9: ÍNDICE <i>SILHOUETTE</i> DEL MÉTODO JERÁRQUICO AL PARTICIONAR CON 10 CLUSTERS	58
FIGURA 4.10: MÉTODO JERÁRQUICO CON DISTANCIA DTW PARTICIONANDO EN 10 CLUSTERS	60
FIGURA 4.11: ÍNDICE <i>SILHOUETTE</i> DEL MÉTODO JERÁRQUICO USANDO DISTANCIA DTW AL PARTICIONAR CON 10 CLUSTERS	60
FIGURA 5.1: MEDIDAS DE ERROR APLICADO SOBRE LOS RESULTADOS OBTENIDOS POR EL SVM UTILIZANDO DISTINTOS KERNELS	69
FIGURA 5.2: SILHOUETTE OBTENIDA PARA CADA UNO DE LOS MÉTODOS EMPLEADOS SOBRE CADA UNO DE LOS CONJUNTOS DE DATOS	70
FIGURA 5.3: TIEMPO PROMEDIO DE EJECUCIÓN DE LOS ALGORITMOS PARTE 1	71
FIGURA C.1: K-MEANS CON 2 CLUSTERS	103
FIGURA C.2: ITERACIÓN 1, K-MEANS CON 3 CLUSTERS	105
FIGURA C.3: ITERACIÓN 7, KMEANS CON 4 CLUSTER	108

FIGURA C.4: ITERACIÓN 9 CON 4 CLUSTERS	109
FIGURA C.5: ITERACIÓN 1, KMEANS CON 5 CLUSTERS	112
FIGURA C.6: ITERACIÓN 3. KMEANS CON 5 CLUSTERS	113
FIGURA C.7 ITERACIÓN 4 CON 5 CLUSTERS	114
FIGURA C.8: ITERACIÓN 9, KMEANS CON 5 CLUSTERS	115
FIGURA C.9: ITERACION1 CON 6 CLUSTERS	115
FIGURA C.10: ITERACIÓN 5, KMEANS CON 6 CLUSTERS	118
FIGURA C.11: ITERACIÓN 4. CON 7 CLUSTERS	119
FIGURA C.12: ITERACIÓN 8. KMEANS CON 7 CLUSTERS	122
FIGURA C.13: ITERACIÓN 9. KMEANS CON 7 CLUSTERS	122
FIGURA C.14: ITERACIÓN 1. KMEANS CON 8 CLUSTERS.....	124
FIGURA C.15: ITERACIÓN 3. KMEANS CON 8 CLUSTERS	124
FIGURA C.16: ITERACIÓN 8. KMEANS CON 8 CLUSTERS	124
FIGURA C.17: ITERACIÓN 2. KMEANS CON 8 CLUSTERS	126
FIGURA C.18: K-MEANS CON 8 CLUSTERS. ITERACIÓN 4	126
FIGURA C.19: ITERACIÓN 10, KMEANS CON 8 CLUSTERS	127
FIGURA C.20: ITERACIÓN 1. KMEANS CON 9 CLUSTERS	129
FIGURA C.21: ITERACIÓN 5. KMEANS CON 8 CLUSTERS	129
FIGURA C.22: ITERACIÓN 7. KMEANS CON 9 CLUSTERS	130
FIGURA C.23: ITERACIÓN 8. KMEANS CON 9 CLUSTERS	130
FIGURA C.24: ITERACIÓN 3. KMEANS CON 9 CLUSTERS	132
FIGURA C.25: ITERACIÓN 6. KMEANS CON 9 CLUSTERS	132
FIGURA C.26: ITERACIÓN 2. KMEANS CON 10 CLUSTERS	134
FIGURA C.27: ITERACIÓN 3. KMEANS CON 10 CLUSTERS	134
FIGURA C.28: ITERACIÓN 4. KMEANS CON 10 CLUSTERS	134
FIGURA C.29: ITERACIÓN 5. KMEANS CON 10 CLUSTERS	135
FIGURA C.30: ITERACIÓN 6. KMEANS CON 10 CLUSTERS	136
FIGURA C.31: ITERACIÓN 7. KMEANS CON 10 CLUSTERS	136
FIGURA C.32: ITERACIÓN 8. KMEANS CON 10 CLUSTERS	138
FIGURA C.33: ITERACIÓN 9. KMEANS CON 10 CLUSTERS	138
FIGURA C.34: ITERACIÓN 1. KMEANS CON 2 CLUSTERS	139
FIGURA C.35: K-MEANS CON 3 CLUSTERS	141
FIGURA C.36: ITERACIÓN 1. KMEANS CON 4 CLUSTERS	143
FIGURA C.37: ITERACIÓN 2. KMEANS CON 4 CLUSTERS	143
FIGURA C.38: ITERACIÓN 5. KMEANS CON 4 CLUSTERS	145
FIGURA C.39: ITERACIÓN 8. KMEANS CON 4 CLUSTERS	145
FIGURA C.40: ITERACIÓN 10. KMEANS CON 4 CLUSTERS	146
FIGURA C.41: ITERACIÓN 1. KMEANS CON 5 CLUSTERS	148
FIGURA C.42; ITERACIÓN 3. K-MEANS CON 5 CLUSTERS	148
FIGURA C.43: ITERACIÓN 8. KMEANS CON 5 CLUSTERS	150
FIGURA C.44: ITERACIÓN 2. KMEANS CON 6 CLUSTERS	152
FIGURA C.45: ITERACIÓN 4. KMEANS CON 6 CLUSTERS	152
FIGURA C.46: ITERACIÓN 5. KMEANS CON 6 CLUSTERS	154
FIGURA C . 47: ITERACIÓN 9. KMEANS CON 6 CLUSTERS	155
FIGURA C.48: ITERACIÓN 1. KMEANS CON 7 CLUSTERS	156
FIGURA C.49: ITERACIÓN 3. KMEANS CON 7 CLUSTERS	158

FIGURA C.50: ITERACIÓN 4. KMEANS CON 7 CLUSTERS	158
FIGURA C.51: ITERACIÓN 6. KMEANS CON 7 CLUSTERS	160
FIGURA C.52: ITERACIÓN 1. KMEANS CON 8 CLUSTERS	162
FIGURA C.53: ITERACIÓN 3. KMEANS CON 8 CLUSTERS	162
FIGURA C.54: ITERACIÓN 6. KMEANS CON 8 CLUSTERS	164
FIGURA C.55: ITERACIÓN 8. KMEANS CON 8 CLUSTERS	164
FIGURA C.56: ITERACIÓN 3. KMEANS CON 9 CLUSTERS	167
FIGURA D.1: KMEDOIDS CON 2 CLUSTERS.....	173
FIGURA D.2: “SILHOUETTE” SOBRE KMEDOIDS CON 2 CLUSTERS.....	173
FIGURA D. 3: KMEDOIDS CON 3 CLUSTERS.....	175
FIGURA D. 4: “SILHOUETTE” SOBRE KMEDOIDS EN 3 CLUSTERS	175
FIGURA D. 5: KMEDOIDS CON 4 CLUSTERS	177
FIGURA D. 6: “SILHOUETTE” SOBRE KMEDOIDS EN 4 CLUSTERS.....	177
FIGURA D. 7: MEDIOS CON 5 CLUSTERS	179
FIGURA D. 8: “SILHOUETTE” SOBRE KMEDOIDS CON 5 CLUSTERS.....	179
FIGURA D. 9: KMEDOIDS CON 6 CLUSTERS	181
FIGURA D.10: “SILHOUETTE”. KMEDOIDS CON 6 CLUSTERS.....	181
FIGURA D.11: KMEDOIDS CON 7 CLUSTERS.....	183
FIGURA D.12: “SILHOUETTE” SOBRE KMEDOIDS CON 7 CLUSTERS.....	183
FIGURA D.13: KMEDOIDS CON 8 CLUSTERS.....	185
FIGURA D.14: “SILHOUETTE” SOBRE KMEDOIDS CON 8 CLUSTERS.....	185
FIGURA D.15: KMEDOIDS CON 9 CLUSTERS.....	187
FIGURA D.16: “SILHOUETTE” SOBRE KMEDOIDS CON 9 CLUSTERS	187
FIGURA D.17: KMEDOIDS CON 10 CLUSTERS.....	189
FIGURA D.18; “SILHOUETTE” SOBRE KMEDOIDS CON 10 CLUSTERS.....	189
FIGURA D.19: KMEDOIDS CON 2 CLUSTERS.....	191
FIGURA D.20: ÍNDICE “SILHOUETTE” SOBRE KMEDOIDS CON 2 CLUSTERS	191
FIGURA D.21: KMEDOIDS CON 3 CLUSTERS.....	193
FIGURA D.22: “SILHOUETTE” SOBRE KMEDOIDS CON 3 CLUSTERS.....	193
FIGURA D.23: KMEDOIDS CON 4 CLUSTERS.....	195
FIGURA D.24: “SILHOUETTE” SOBRE KMEDOIDS CON 4 CLUSTERS	195
FIGURA D.25: KMEDOIDS CON 5 CLUSTERS.....	197
FIGURA D.26: “SILHOUETTE” SOBRE KMEDOIDS CON 5 CLUSTERS.....	197
FIGURA D.27: KMEDOIDS CON 6 CLUSTERS.....	199
FIGURA D.28: “SILHOUETTE” SOBRE KMEDOIDS CON 6 CLUSTERS.....	199
FIGURA D.29: MEDIOS CON 7 CLUSTERS	201
FIGURA D.30: KMEDOIDS CON 8 CLUSTERS.....	203
FIGURA D.31: “SILHOUETTE” SOBRE KMEDOIDS CON 8 CLUSTERS.....	203
FIGURA D.32: KMEDOIDS CON 9 CLUSTERS.....	205
FIGURA D.33: “SILHOUETTE” SOBRE KMEDOIDS CON 9 CLUSTERS	205
FIGURA D.34: KMEDOIDS CON 10 CLUSTERS.....	207
FIGURA D.35: “SILHOUETTE” SOBRE KMEDOIDS CON 10 CLUSTERS.....	207

FIGURA E.1: JERÁRQUICO CON DOS CLUSTERS	209
FIGURA E.2: DENDOGRAMA. JERÁRQUICO CON 2 CLUSTERS	209
FIGURA E.3: “SILHOUETTE” SOBRE JERÁRQUICO CON 2 CLUSTERS.....	209
FIGURA E.4: JERÁRQUICO CON 3 CLUSTERS	211
FIGURA E.5: DENDOGRAMA. JERÁRQUICO CON 3 CLUSTERS	211
FIGURA E.6: “SILOUETTE” SOBRE JERÁRQUICO CON 3 CLUSTERS	211
FIGURA E.7: JERÁRQUICO CON 4 CLUSTERS	213
FIGURA E.8: DENDOGRAMA. JERÁRQUICO CON 4 CLUSTERS	213
FIGURA E.9: “SILOHOUETTE”. JERÁRQUICO CON 4 CLUSTERS	213
FIGURA E.10: JERÁRQUICO CON 5 CLUSTERS	215
FIGURA E.11: DENDOGRAMA. JERÁRQUICO CON 5 CLUSTERS	215
FIGURA E . 12: “SILHOOUETTE” SOBRE JERÁRQUICO CON 5 CLUSTERS	215
FIGURA E.13: JERÁRQUICO CON 6 CLUSTERS	217
FIGURA E.14: DENDOGRAMA. JERÁRQUICO CON 6 CLUSTERS	217
FIGURA E.15: “SÍLHOUETTE” SOBRE JERÁRQUICO CON 6 CLUSTERS	217
FIGURA E.16: JERÁRQUICO CON 7 CLUSTERS	219
FIGURA E.17: DENDOGRAMA. JERÁRQUICO CON 7 CLUSTERS	219
FIGURA E.18: “SILHOUETTE” SOBRE JERÁRQUICO CON 7 CLUSTERS	219
FIGURA E.19: JERÁRQUICO CON 7 CLUSTERS	221
FIGURA E.20: DENDOGRAMA. JERÁRQUICO CON 8 CLUSTERS	221
FIGURA E.21: “SILHOUETTE” SOBRE JERÁRQUICO CON 8 CLUSTERS	221
FIGURA E.22: JERÁRQUICO CON 9 CLUSTERS	223
FIGURA E.23: DENDOGRAMA. JERÁRQUICO CON 9 CLUSTERS	223
FIGURA E.24: “SILHOUETTE” SOBRE JERÁRQUICO CON 9 CLUSTERS	223
FIGURA E.25: JERÁRQUICO CON 10 CLUSTERS	225
FIGURA E.26: DENDOGRAMA. JERÁRQUICO CON 10 CLUSTERS	225
FIGURA E.27: “SILHOUETTE” SOBRE JERÁRQUICO CON 10 CLUSTERS	225
FIGURA E.28: JERÁRQUICO CON DOS CLUSTERS	227
FIGURA E.29: DENDOGRAMA. JERÁRQUICO CON 2 CLUSTERS	227
FIGURA E.30: “SILHOUETTE” SOBRE JERÁRQUICO CON 2 CLUSTERS.....	227
FIGURA E.31: JERÁRQUICO CON 3 CLUSTERS	229
FIGURA E.32: DENDOGRAMA. JERÁRQUICO CON 3 CLUSTERS	229
FIGURA E.33: “SILOUETTE” SOBRE JERÁRQUICO CON 3 CLUSTERS.....	229
FIGURA E.34: JERÁRQUICO CON 4 CLUSTERS	231
FIGURA E.35: DENDOGRAMA. JERÁRQUICO CON 4 CLUSTERS	231
FIGURA E.36: “SILOHOUETTE”. JERÁRQUICO CON 4 CLUSTERS	231
FIGURA E.37: JERÁRQUICO CON 5 CLUSTERS	233
FIGURA E.38: DENDOGRAMA. JERÁRQUICO CON 5 CLUSTERS	233
FIGURA E.39: “SILHOOUETTE” SOBRE JERÁRQUICO CON 5 CLUSTERS	233
FIGURA E.40: JERÁRQUICO CON 6 CLUSTERS	235
FIGURA E.41: DENDOGRAMA . JERÁRQUICO CON 6 CLUSTERS	235
FIGURA E.42: “SÍLHOUETTE” SOBRE JERÁRQUICO CON 6 CLUSTERS	235
FIGURA E.43: JERÁRQUICO CON 7 CLUSTERS	237
FIGURA E.44: DENDOGRAMA. JERÁRQUICO CON 7 CLUSTERS	237
FIGURA E.45: “SILHOUETTE” SOBRE JERÁRQUICO CON 7 CLUSTERS	237
FIGURA E.46: JERÁRQUICO CON 8 CLUSTERS	239

FIGURA E.47: DENDOGRAMA. JERÁRQUICO CON 8 CLUSTERS	239
FIGURA E.48: “SILHOUETTE” SOBRE JERÁRQUICO CON 8 CLUSTERS	239
FIGURA E.49: JERÁRQUICO CON 9 CLUSTERS	241
FIGURA E.50: DENDOGRAMA. JERÁRQUICO CON 9 CLUSTERS	241
FIGURA E.51: “SILHOUETTE” SOBRE JERÁRQUICO CON 9 CLUSTERS	241
FIGURA E.52: JERÁRQUICO CON 10 CLUSTERS	243
FIGURA E.53: DENDOGRAMA. JERÁRQUICO CON 10 CLUSTERS	243
FIGURA E.54: “SILHOUETTE” SOBRE JERÁRQUICO CON 10 CLUSTERS	243
FIGURA E.55: JERÁRQUICO CON DISTANCIA “DTW”, CON 6 CLUSTERS.....	253
FIGURA E.56: DENDOGRAMA. JERÁRQUICO CON DISTANCIA “DTW”, CON 6 CLUSTERS.....	254
FIGURA E.57: “SILHOUETTE” SOBRE JERÁRQUICO CON DISTANCIA “DTW “, CON 6 CLUSTERS	254
FIGURA F.1: JERÁRQUICO CON DISTANCIA “DTW”, CON DOS CLUSTERS.....	245
FIGURA F.2: DENDOGRAMA. JERÁRQUICO CON DISTANCIA “DTW” CON 2 CLUSTERS.....	246
FIGURA F.3: “SILHOUETTE” SOBRE JERÁRQUICO CON DISTANCIA “DTW “,CON 2 CLUSTERS	246
FIGURA F.4: JERÁRQUICO CON DISTANCIA “DTW”, CON 3 CLUSTERS	247
FIGURA F.5: DENDOGRAMA. JERÁRQUICO CON DISTANCIA “DTW” , CON 3 CLUSTERS.....	248
FIGURA F.6: “SILHOUETTE” SOBRE JERÁRQUICO CON DISTANCIA “DTW “,CON 3 CLUSTERS	248
FIGURA F.7: JERÁRQUICO CON DISTANCIA “DTW”, CON 4 CLUSTERS	249
FIGURA F.8: DENDOGRAMA. JERÁRQUICO CON DISTANCIA “DTW” , CON 4 CLUSTERS.....	250
FIGURA F.9: “SILHOUETTE” SOBRE JERÁRQUICO CON DISTANCIA “DTW “,CON 4 CLUSTERS	250
FIGURA F.10: JERÁRQUICO CON DISTANCIA “DTW”, CON 5 CLUSTERS.....	251
FIGURA F.11: DENDOGRAMA. JERÁRQUICO CON DISTANCIA “DTW” , CON 5 CLUSTERS.....	252
FIGURA F.12: “SILHOUETTE” SOBRE JERÁRQUICO CON DISTANCIA “DTW “,CON 5 CLUSTERS	252
FIGURA F.13: JERÁRQUICO CON DISTANCIA “DTW”, CON 7 CLUSTERS.....	255
FIGURA F.14: DENDOGRAMA. JERÁRQUICO CON DISTANCIA “DTW” , CON 7 CLUSTERS.....	256
FIGURA F . 15: “SILHOUETTE” SOBRE JERÁRQUICO CON DISTANCIA “DTW “, CON 7 CLUSTERS.....	256
FIGURA F.16: JERÁRQUICO CON DISTANCIA “DTW”, CON 8 CLUSTERS	257
FIGURA F.17: DENDOGRAMA. JERÁRQUICO CON DISTANCIA “DTW” , CON 8 CLUSTERS.....	258
FIGURA F . 18: “SILHOUETTE” SOBRE JERÁRQUICO CON DISTANCIA “DTW “, CON 8 CLUSTERS.....	258
FIGURA F.19: JERÁRQUICO CON DISTANCIA “DTW”, CON 9 CLUSTERS.....	259
FIGURA F.20: DENDOGRAMA. JERÁRQUICO CON DISTANCIA “DTW” , CON 9 CLUSTERS.....	260
FIGURA F.21: “SILHOUETTE” SOBRE JERÁRQUICO CON DISTANCIA “DTW “, CON 9 CLUSTERS	260
FIGURA F.22: JERÁRQUICO CON DISTANCIA “DTW”, CON 10 CLUSTERS.....	261
FIGURA F.23: DENDOGRAMA. JERÁRQUICO CON DISTANCIA “DTW” , CON 10 CLUSTERS.....	262
FIGURA F.24: “SILHOUETTE” SOBRE JERÁRQUICO CON DISTANCIA “DTW “, CON 10 CLUSTERS	262
FIGURA F.25: JERÁRQUICO CON DISTANCIA “DTW”, CON DOS CLUSTERS.....	263
FIGURA F.26: DENDOGRAMA. JERÁRQUICO CON DISTANCIA “DTW” CON 2 CLUSTERS.....	264
FIGURA F.27: “SILHOUETTE” SOBRE JERÁRQUICO CON DISTANCIA “DTW “,CON 2 CLUSTERS	264
FIGURA F.28: JERÁRQUICO CON DISTANCIA “DTW”, CON 3 CLUSTERS.....	265
FIGURA F.29: DENDOGRAMA. JERÁRQUICO CON DISTANCIA “DTW” , CON 3 CLUSTERS.....	266
FIGURA F . 30: “SILHOUETTE” SOBRE JERÁRQUICO CON DISTANCIA “DTW “,CON 3 CLUSTERS.....	266
FIGURA F.31: JERÁRQUICO CON DISTANCIA “DTW”, CON 4 CLUSTERS.....	267
FIGURA F.32: DENDOGRAMA. JERÁRQUICO CON DISTANCIA “DTW” , CON 4 CLUSTERS.....	268
FIGURA F.33: “SILHOUETTE” SOBRE JERÁRQUICO CON DISTANCIA “DTW “,CON 4 CLUSTERS	268

FIGURA F.34: JERÁRQUICO CON DISTANCIA “DTW”, CON 5 CLUSTERS.....	269
FIGURA F.35: DENDOGRAMA. JERÁRQUICO CON DISTANCIA “DTW” , CON 5 CLUSTERS.....	270
FIGURA F.36: “SILHOUETTE” SOBRE JERÁRQUICO CON DISTANCIA “DTW “,CON 5 CLUSTERS	270
FIGURA F.37: JERÁRQUICO CON DISTANCIA “DTW”, CON 6 CLUSTERS.....	271
FIGURA F.38: DENDOGRAMA. JERÁRQUICO CON DISTANCIA “DTW” , CON 6 CLUSTERS.....	272
FIGURA F . 39: “SILHOUETTE” SOBRE JERÁRQUICO CON DISTANCIA “DTW “, CON 6 CLUSTERS.....	272
FIGURA F.40: JERÁRQUICO CON DISTANCIA “DTW”, CON 7 CLUSTERS.....	273
FIGURA F.41: DENDOGRAMA. JERÁRQUICO CON DISTANCIA “DTW” , CON 7 CLUSTERS.....	274
FIGURA F.42: “SILHOUETTE” SOBRE JERÁRQUICO CON DISTANCIA “DTW “, CON 7 CLUSTERS	274
FIGURA F.43: JERÁRQUICO CON DISTANCIA “DTW”, CON 8 CLUSTERS.....	275
FIGURA F.44: DENDOGRAMA. JERÁRQUICO CON DISTANCIA “DTW” , CON 8 CLUSTERS.....	276
FIGURA F . 45: “SILHOUETTE” SOBRE JERÁRQUICO CON DISTANCIA “DTW “, CON 8 CLUSTERS.....	276
FIGURA F.46: JERÁRQUICO CON DISTANCIA “DTW”, CON 9 CLUSTERS.....	277
FIGURA F.47: DENDOGRAMA. JERÁRQUICO CON DISTANCIA “DTW” , CON 9 CLUSTERS.....	278
FIGURA F.48: “SILHOUETTE” SOBRE JERÁRQUICO CON DISTANCIA “DTW “, CON 9 CLUSTERS	278
FIGURA F.49: JERÁRQUICO CON DISTANCIA “DTW”, CON 10 CLUSTERS.....	279
FIGURA F.50: DENDOGRAMA. JERÁRQUICO CON DISTANCIA “DTW” , CON 10 CLUSTERS.....	280
FIGURA F.51: “SILHOUETTE” SOBRE JERÁRQUICO CON DISTANCIA “DTW “, CON 10 CLUSTERS	280

Índice de tablas

TABLA 4.1: ITERACIÓN 1 CON KMEANS CON K=10	52
TABLA 4.2: IMPLEMENTADO K-MEDIODS Y PARTICIONANDO EN 10 CLUSTERS	54
TABLA 4.3: IMPLEMENTADO K-MEDIODS Y PARTICIONANDO EN 10 CLUSTERS	54
TABLA 4.4: MÉTODO JERÁRQUICO, PARTICIONANDO EN 10 CLUSTERS.....	55
TABLA 4.5: MÉTODO JERÁRQUICO, PARTICIONANDO EN 10 CLUSTERS.....	57
TABLA 4.6: MÉTODO JERÁRQUICO PARTICIONANDO 10 CLUSTERS USANDO DISTANCIA DE SIMILITUD DTW	59
TABLA 4.7: MÉTODO JERÁRQUICO PARTICIONANDO 10 CLUSTERS USANDO DISTANCIA DE SIMILITUD DTW	59
TABLA 4.8: SVM CON KERNEL LINEAL EMPLEANDO DATOS PROMEDIADOS EN ESCALA LOGARÍTMICA.....	62
TABLA 4.9: SVM CON KERNEL LINEAL EMPLEANDO LOS DATOS PROMEDIADOS DE 50 EN 50 Hz	63
TABLA C.1: K-MEANS CON 2 CLUSTERS	103
TABLA C.2: ITERACIÓN 1,3 Y 7, KMEANS CON 3 CLUSTERS.....	104
TABLA C.3 ITERACIÓN 1,3 Y 7, KMEANS CON 3 CLUSTERS.....	105
TABLA C.4: ITERACIÓN 4,5,9 Y 10, KMEANS CON 3 CLUSTERS.....	106
TABLA C.5: ITERACIÓN 1,3,6,8 , KMEANS CON 4 CLUSTERS	107
TABLA C.6: ITERACIÓN 2 KMEANS CON 4 CLUSTERS	108
TABLA C.7: ITERACIÓN 9 CON 4 CLUSTERS	109
TABLA C.8: ITERACIÓN 3,6 Y 10, KMEANS CON 4 CLUTERS.....	110
TABLA C.9; ITERACIÓN 1, 5 Y 7, KMEANS CON 5 CLUSTERS	111
TABLA C.10: ITERACIONES 2 Y 3. KMEANS CON 5 CLUSTERS.....	112
TABLA C.11: ITERACIÓN 4 Y 10 , KMEANS CON 5 CLUSTERS	113
TABLA C.12: ITERACIÓN 5,8 Y 9, KMEANS CON 5 CLUSTERS.....	114
TABLA C.13: ITERACIONES 1,2,3 Y 4. KMEANS CON 6 CLUSTERS	116
TABLA C.14: ITERACIONES 5,6,7 Y 8. KMEANS CON 6 CLUSTERS	117
TABLA C.15: ITERACIONES 9 Y 10. K-MEANS CON 6 CLUSTERS	118
TABLA C.16: ITERACIONES 3 Y 4. KMEANS CON 7 CLUSTERS.....	119
TABLA C.17: ITERACIONES 1, 2, 5 Y 6. KMEANS CON7 CLUSTERS	120
TABLA C.18: ITERACIONES 7,8,9 Y 10. KMEANS CON 7 CLUSTERS	121
TABLA C.19: ITERACIONES 1,3,6 Y 8. KMEANS CON 8 CLUSTERS	123
TABLA C.20: ITERACIONES 2, 4,,5 Y 7. KMEANS CON 8CLUSTERS	125
TABLA C.21: ITERACIONES 9 Y 10. KMEANS CON 8 CLUSTERS	127
TABLA C.22: ITERACIONES 1,5,7,8 Y 9. KMEANS CON 9 CLUSTERS	128
TABLA C.23: ITERACIÓN 2,3,4,6 Y 10. KMEANS CON 9 CLUSTERS.....	131
TABLA C.24: ITERACIONES 2,3,4 Y 5. KMEANS CON 10 CLUSTERS	133
TABLA C.25: ITERACIÓN 6 Y7 . KMEANS CON 10 CLUSTERS.....	135
TABLA C.26: ITERACIONES 8,9 Y 10. KMEANS CON 10 CLUSTERS	137
TABLA C.27: RESULTADOS OBTENIDOS DEL ALGORITMO K-MEDIAS CON K=2. ITERACIONES DE 1 A 10.....	139
TABLA C.28: K-MEANS CON 3 CLUSTERS	140
TABLA C.29: TIEMPO DE EJECUCIÓN POR CADA UNAS DE LAS ITERACIONES	140
TABLA C.30: ITERACIONES 1,2,3 Y 4. K-MEANS CON 4 CLUSTERS	142
TABLA C.31: ITERACIONES 5,6,7 Y 8. K-MEANS CON 4 CLUSTERS	144

TABLA C.32: ITERACIONES 9 Y 10. KMEANS CON 4 CLUSTERS	146
TABLA C.33: ITERACIÓN 1,2,3 Y 4. K-MEANS CON 4 CLUSTERS.....	147
TABLA C . 34: ITERACIONES 5,6,7 Y 8. KMEANS CON 5 CLUSTERS	149
TABLA C.35: ITERACIONES 9 Y 10. KMEANS CON 5 CLUSTERS	150
TABLA C.36: ITERACIONES 1,2,3 Y 4 . KMEANS CON 6 CLUSTERS	151
TABLA C.37: ITERACIONES 5,6,7 Y 8. KMEANS CON 6 CLUSTERS	153
TABLA C.38: ITERACIONES 9 Y 10. KMEANS CON 6 CLUSTERS	154
TABLA C.39: ITERACIONES 1 Y 2. K-MEANS CON 7 CLUSTERS	155
TABLA C.40: ITERACIONES 3,4 Y 5. KMEANS CON 7 CLUSTERS	157
TABLA C.41: ITERACIONES 6,7,8 Y 9. KMEANS CON 7 CLUSTERS	159
TABLA C . 42: ITERACIÓN 10. KMEANS CON 7 CLUSTERS	160
TABLA C.43: ITERACIONES 1,2,3 Y 4. KMEANS CON 8 CLUSTERS	161
TABLA C.44: ITERACIONES 5,6,7 Y 8. KMEANS CON 8 CLUSTERS	163
TABLA C.45: ITERACIONES 9 Y 10. KMEANS CON 8 CLUSTERS	165
TABLA C.46: ITERACIÓN 1,2,3 Y 4. KMEANS CON 9 CLUSTERS.....	166
TABLA C.47: ITERACIONES 5 Y 6. KMEANS CON 9 CLUSTERS	167
TABLA C.48: ITERACIONES 7,8,9 Y 10. KMEANS CON 9 CLUSTERS	168
TABLA C.49: ITERACIONES 1,2,3 Y 4. KMEANS CON 10 CLUSTERS	169
TABLA C.50: ITERACIONES 5,6,7 Y 8. KMEANS CON 10 CLUSTERS	170
TABLA C.51: ITERACIONES 9 Y 10. KMEANS CON 10 CLUSTERS	171
TABLA D.1: KMEDOIDS CON 2 CLUSTERS	172
TABLA D.2: TIEMPOS DE EJECUCIÓN	172
TABLA D.3: KMEDOIDS CON 3 CLUSTERS	174
TABLA D.4: TIEMPOS DE EJECUCIÓN. KMEDOIDS CON 3 CLUSTERS	174
TABLA D.5: KMEDOIDS CON 4 CLUSTERS	176
TABLA D.6: TIEMPOS DE EJECUCIÓN. KMEDOIDS CON 4 CLUSTERS	176
TABLA D.7: K-MEDOIDES CON K=5	178
TABLA D.8: TIEMPOS DE EJECUCIÓN. KMEDOIDS CON 5 CLUSTERS	178
TABLA D.9: KMEDOIDS CON 6 CLUSTERS	180
TABLA D.10: TIEMPOS DE EJECUCIÓN. KMEDOIDS CON 6 CLUSTERS	180
TABLA D.11: KMEDOIDS CON 7 CLUSTERS	182
TABLA D.12: TIEMPOS DE EJECUCIÓN. KMEDOIDS CON 7 CLUSTERS	182
TABLA D.13: KMEDOIDS CON 8 CLUSTERS	184
TABLA D.14: TIEMPO DE EJECUCIÓN. KMEDOIDS CON 8 CLUSTERS.....	184
TABLA D.15: KMEDOIDS CON 9 CLUSTERS	186
TABLA D.16: TIEMPOS DE EJECUCIÓN. KMEDOIDS CON 9 CLUSTERS	186
TABLA D.17: KMEDOIDS CON 10 CLUSTERS	188
TABLA D.18: TIEMPOS DE EJECUCIÓN. KMEDOIDS CON 10 CLUSTERS	188
TABLA D.19: KMEDOIDS CON 2 CLUSTERS	190
TABLA D.20: TIEMPOS DE EJECUCIÓN. KMEDOIDS CON 2 CLUSTERS	190
TABLA D.21: K-MEDOIDES CON 3 CLUSTERS	192
TABLA D.22: TIEMPOS DE EJECUCIÓN. K-MEDOIDES CON 3 CLUSTERS	192
TABLA D.23: KMEDOIDS CON 4 CLUSTERS	194
TABLA D.24: TIEMPO DE EJECUCIÓN. KMEDOIDS CON 4 CLUSTERS.....	194

TABLA D.25: KMEDOIDS CON 5 CLUSTERS	196
TABLA D.26: TIEMPOS DE EJECUCIÓN, KMEDOIDS CON 5 CLUSTERS	196
TABLA D.27: KMEDOIDS CON 6 CLUSTERS	198
TABLA D.28: TIEMPOS DE EJECUCIÓN POR CADA ITERACIÓN, KMEDOIDS CON 6 CLUSTERS	198
TABLA D.29: KMEDOIDS CON 7 CLUSTERS	200
TABLA D.30: TIEMPOS DE EJECUCIÓN POR CADA ITERACIÓN, KMEDOIDS CON 7 CLUSTERS	200
TABLA D.31: KMEDOIDS CON 8 CLUSTERS	202
TABLA D.32: TIEMPOS DE EJECUCIÓN POR CADA ITERACIÓN, KMEDOIDS CON 8 CLUSTERS	202
TABLA D.33: KMEDOIDS CON 9 CLUSTER	204
TABLA D.34: TIEMPOS DE EJECUCIÓN POR CADA ITERACIÓN, KMEDOIDS CON 9 CLUSTERS	204
TABLA D.35: KMEDOIDS CON 10 CLUSTERS	206
TABLA D.36: TIEMPOS DE EJECUCIÓN POR CADA ITERACIÓN, KMEDOIDS CON 10 CLUSTERS	206
TABLA E.1: JERÁRQUICO CON 2 CLUSTERS	208
TABLA E.2: TIEMPOS DE EJECUCIÓN POR CADA ITERACIÓN, JERÁRQUICO CON 2 CLUSTERS	208
TABLA E.3: JERÁRQUICO CON 3 CLUSTERS	210
TABLA E.4: TIEMPOS DE EJECUCIÓN POR CADA ITERACIÓN, JERÁRQUICO CON 3 CLUSTERS	210
TABLA E.5: JERÁRQUICO CON 4 CLUSTERS	212
TABLA E.6: TIEMPOS DE EJECUCIÓN POR CADA ITERACIÓN, JERÁRQUICO CON 4 CLUSTERS	212
TABLA E.7: JERÁRQUICO CON 5 CLUSTERS	214
TABLA E.8: TIEMPOS DE EJECUCIÓN POR CADA ITERACIÓN, JERÁRQUICO CON 5 CLUSTERS	214
TABLA E.9: JERÁRQUICO CON 6 CLUSTERS	216
TABLA E.10: TIEMPOS DE EJECUCIÓN POR CADA ITERACIÓN, JERÁRQUICO CON 6 CLUSTERS	216
TABLA E.11: JERÁRQUICO CON 7 CLUSTERS	218
TABLA E.12: TIEMPOS DE EJECUCIÓN POR CADA ITERACIÓN, JERÁRQUICO CON 7 CLUSTERS	218
TABLA E.13: JERÁRQUICO CON 8 CLUSTERS	220
TABLA E.14: TIEMPOS DE EJECUCIÓN POR CADA ITERACIÓN, JERÁRQUICO CON 8 CLUSTERS	220
TABLA E.15: JERÁRQUICO CON 9 CLUSTERS	222
TABLA E.16: TIEMPOS DE EJECUCIÓN POR CADA ITERACIÓN, JERÁRQUICO CON 9 CLUSTERS	222
TABLA E.17: JERÁRQUICO CON 10 CLUSTERS	224
TABLA E.18: TIEMPOS DE EJECUCIÓN POR CADA ITERACIÓN, JERÁRQUICO CON 10 CLUSTERS	224
TABLA E.19: JERÁRQUICO CON 2 GRUPOS	226
TABLA E.20: TIEMPOS DE EJECUCIÓN POR CADA ITERACIÓN, JERÁRQUICO CON 2 CLUSTERS	226
TABLA E.21: JERÁRQUICO CON 3 CLUSTERS	228
TABLA E.22: TIEMPOS DE EJECUCIÓN POR CADA ITERACIÓN, JERÁRQUICO CON 3 CLUSTERS	228
TABLA E.23: JERÁRQUICO CON 4 CLUSTERS	230
TABLA E.24: TIEMPOS DE EJECUCIÓN POR CADA ITERACIÓN, JERÁRQUICO CON 4 CLUSTERS	230
TABLA E.25: JERÁRQUICO CON 5 CLUSTERS	232
TABLA E.26: TIEMPOS DE EJECUCIÓN POR CADA ITERACIÓN, JERÁRQUICO CON 5 CLUSTERS	232
TABLA E.27: JERÁRQUICO CON 6 CLUSTERS	234
TABLA E.28: TIEMPOS DE EJECUCIÓN POR CADA ITERACIÓN, JERÁRQUICO CON 6 CLUSTERS	234
TABLA E.29: JERÁRQUICO CON 7 CLUSTERS	236
TABLA E.30: TIEMPOS DE EJECUCIÓN POR CADA ITERACIÓN, JERÁRQUICO CON 7 CLUSTERS	236
TABLA E.31; JERÁRQUICO CON 8 CLUSTERS	238
TABLA E.32: TIEMPOS DE EJECUCIÓN POR CADA ITERACIÓN, JERÁRQUICO CON 8 CLUSTERS	238

TABLA E.33: JERÁRQUICO CON 9 CLUSTERS.....	240
TABLA E.34: TIEMPOS DE EJECUCIÓN POR CADA ITERACIÓN. JERÁRQUICO CON 9 CLUSTERS.....	240
TABLA E.35: JERÁRQUICO CON 10 CLUSTERS.....	242
TABLA E.36: TIEMPOS DE EJECUCIÓN POR CADA ITERACIÓN. JERÁRQUICO CON 10 CLUSTERS.....	242
TABLA F.1: JERÁRQUICO CON 2 CLUSTERS.....	245
TABLA F.2: JERÁRQUICO CON DISTANCIA “DTW” Y AGRUPANDO CON 3 CLUSTERS.....	247
TABLA F.3: JERÁRQUICO CON DISTANCIA “DTW” Y AGRUPANDO CON 4 CLUSTERS.....	249
TABLA F.4: JERÁRQUICO CON DISTANCIA “DTW” Y AGRUPANDO CON 5 CLUSTERS.....	251
TABLA F.5: JERÁRQUICO CON DISTANCIA “DTW” Y AGRUPANDO CON 6 CLUSTERS.....	253
TABLA F.6: JERÁRQUICO CON DISTANCIA “DTW” Y AGRUPANDO CON 7 CLUSTERS.....	255
TABLA F.7: JERÁRQUICO CON DISTANCIA “DTW” Y AGRUPANDO CON 8 CLUSTERS.....	257
TABLA F.8: JERÁRQUICO CON DISTANCIA “DTW” Y AGRUPANDO CON 9 CLUSTERS.....	259
TABLA F.9: JERÁRQUICO CON DISTANCIA “DTW” Y AGRUPANDO CON 10 CLUSTERS.....	261
TABLA F.10: JERÁRQUICO CON 2 CLUSTERS.....	263
TABLA F.11: JERÁRQUICO CON DISTANCIA “DTW” Y AGRUPANDO CON 3 CLUSTERS.....	265
TABLA F.12: JERÁRQUICO CON DISTANCIA “DTW” Y AGRUPANDO CON 4 CLUSTERS.....	267
TABLA F.13: JERÁRQUICO CON DISTANCIA “DTW” Y AGRUPANDO CON 5 CLUSTERS.....	269
TABLA F.14: JERÁRQUICO CON DISTANCIA “DTW” Y AGRUPANDO CON 6 CLUSTERS.....	271
TABLA F.15: JERÁRQUICO CON DISTANCIA “DTW” Y AGRUPANDO CON 7 CLUSTERS.....	273
TABLA F.16: JERÁRQUICO CON DISTANCIA “DTW” Y AGRUPANDO CON 8 CLUSTERS.....	275
TABLA F.17: JERÁRQUICO CON DISTANCIA “DTW” Y AGRUPANDO CON 9 CLUSTERS.....	277
TABLA F.18: JERÁRQUICO CON DISTANCIA “DTW” Y AGRUPANDO CON 10 CLUSTERS.....	279
TABLA G.1: SVM CON KERNEL LINEAL.....	281
TABLA G.2: EVALUACIÓN DEL MÉTODO SVM CON KERNEL LINEAL.....	282
TABLA G.3: CLASIFICACIÓN SVM USANDO KERNEL POLINOMIAL.....	282
TABLA G.4: EVALUACIÓN DEL MÉTODO SVM CON KERNEL POLINOMIAL.....	282
TABLA G.5: CLASIFICACIÓN SVM USANDO KERNEL RADIAL GAUSSIANA.....	283
TABLA G.6: EVALUACIÓN DEL MÉTODO SVM CON KERNEL RADIAL GAUSSIANA.....	283
TABLA G.7: CLASIFICACIÓN SVM USANDO KERNEL SIGMOIDAL.....	284
TABLA G.8: EVALUACIÓN DEL MÉTODO SVM CON KERNEL SIGMOIDAL.....	284
TABLA G.9: CLASIFICACIÓN SVM USANDO KERNEL LINEAL.....	285
TABLA G.10: EVALUACIÓN DEL MÉTODO SVM CON KERNEL LINEAL.....	285
TABLA G.11: CLASIFICACIÓN SVM USANDO KERNEL POLINOMIAL.....	286
TABLA G.12: EVALUACIÓN DEL MÉTODO SVM CON KERNEL POLINOMIAL.....	286
TABLA G.13: CLASIFICACIÓN SVM USANDO KERNEL RADIAL GAUSSIANA.....	287
TABLA G.14: EVALUACIÓN DEL MÉTODO SVM CON KERNEL RADIAL GAUSSIANA.....	287
TABLA G.15: CLASIFICACIÓN SVM USANDO KERNEL SIGMOIDAL.....	288
TABLA G.16: EVALUACIÓN DEL MÉTODO SVM CON KERNEL SIGMOIDAL.....	288

Capítulo 1

Introducción

La mayor parte de los individuos tienen la necesidad de efectuar trámites o compras en diversos establecimientos donde en ocasiones se llegan a suscitar actos delictivos en los que pueden existir detonaciones producidas por armas de fuego. La detección, clasificación y localización de disparos producidos por armas de fuego son de gran interés dentro del ámbito de la seguridad tanto pública como privada. La seguridad forma parte esencial del bienestar de una sociedad.

Hasta hoy en día se han realizado investigaciones con respecto al reconocimiento de patrones acústicos aplicando técnicas de minería de datos, en el área de identificación de armas de fuego basado en el sonido del disparo, redes neuronales para reconocer el disparo de distintas armas [6] [9] [10], y máquinas de soporte vectorial (“*Support Vector Machines*”, *SVMs*), para clasificación de sonidos [9]. También, se han realizado implementaciones de hardware y software los cuáles utilizan algoritmos complejos para la detección de sonidos de disparos [7] [8] [11].

En este proyecto se trata de clasificar y distinguir un sonido de disparo entre otros sonidos similares. Como punto de partida, se recolectaron una serie de videos que contenían sonidos de disparos de un solo tipo de arma (“*beretta*” calibre 9mm) y otros sonidos similares producidos por: arma de fogeo, efectos de sonido y explosivos.

Extrayendo del audio de cada uno de los videos se obtuvieron la frecuencia, intensidad, amplitud usando los coeficientes arrojados al aplicar la Transformada Rápida de Fourier, a partir de aquí, los datos mencionados anteriormente se deberán almacenar en archivos de tipo texto para su posterior clasificación aplicando algoritmos de agrupamiento o “*clustering*”: “*K-Means*”, “*k-Medoids*” y “*Hierarchical*” así como máquinas de soporte vectorial (“*Support Vector Machines*”, *SVMs*) y el alineamiento temporal dinámico (“*Dynamic Time Warping*”, DTW).

1.1 Antecedentes

1.1.1 Trabajos internos

- “**Plataforma web para visualizar videos y notificar actividades**” [1]. El cual es un sistema para poder ver y notificar la existencia de videos, donde no se realiza ninguna técnica de minería de datos, tampoco procesamiento para extraer el audio del video como se realiza en el presente proyecto.
- “**Aplicación de Distintas Técnicas de Minería de Datos para el Tratamiento de Información**” [2]. La similitud en ambos proyectos es el uso de la minería

de datos. La diferencia radica en que el proyecto [2] muestra el funcionamiento de algunas técnicas de minería, mientras que en este proyecto se utilizan en específico aquellas técnicas que tengan las características más idóneas para poder realizar la clasificación y poder distinguir el sonido de disparo entre otros sonidos similares y los datos a diferencia del proyecto [2] se extraerán del audio de los videos.

- **“Clasificación de llantos de bebé con wavelets”** [3]. La semejanza entre ambos proyectos radica en que se extraerán características distintivas de los archivos de audio y la clasificación de información mediante al menos una técnica de minería de datos. La diferencia es que en este proyecto se utiliza la Transformada Rápida de Fourier para extraer las características distintivas de los sonidos de disparos y en el proyecto [3] se utiliza la “Transformada Wavelet Discreta y el proyecto [3] clasifica distintos tipos de llantos de bebé.

1.1.2 Trabajos externos

- **“Extracción de información con algoritmos de clasificación”** [4]. La tesis [4] realiza una clasificación supervisada a través de los métodos: Naive Bayes (*NB*), C4.5, k-vecinos más cercanos (*K-NN*, por sus siglas en inglés) y máquinas de soporte vectorial (*SVM*, del inglés *Support Vector Machines*), con la finalidad de hacer más rápida la búsqueda y extracción de información relevante de una cantidad excesiva de documentos, mientras que en este proyecto tecnológico se usó el método *SVM* y los métodos de *clustering: K-Means, K-Medoids*, jerárquico y jerárquico empleando como distancia de similitud el alineamiento temporal dinámico (*DTW*, del inglés *Dynamic Time Warping*), con la finalidad de reconocer el sonido de un disparo real entre los sonidos de disparo de salva, explosivos y efectos de sonidos. La tesis [4] con el método *SVM* obtuvo una exactitud de clasificación por encima del 90%, mientras que en este proyecto se obtuvo una exactitud de reconocimiento del sonido de un disparo real usando un *kernel* lineal aproximadamente del 95%
- **“Sistema de sensores acústicos para detección y localización automática de disparos de armas de fuego en el espacio público”** [5]. Se basa en un conjunto de micrófonos que son distribuidos en distintos puntos de un sector urbano, con el propósito de capturar ruido ambiental y los datos capturados serán procesados y analizados, arrojando finalmente los resultados de detección, reconocimiento e identificación del disparo. La diferencia entre ambos proyectos es que en el proyecto tecnológico no se utilizan sensores y los datos fueron recabados del audio de varios videos en donde exista presencia de disparos producidos por armas de fuego, y se realiza la clasificación mediante el métodos de minería de datos: *K-Means, K-Medoids*, jerárquico, jerárquico empleando como distancia de similitud el *DTW*, y el *SVM* para diferenciar el sonido de un disparo real

entre los sonidos producidos por : Disparo de salva, explosivos y efectos de sonido.

- “**NEURAL NETWORK APPLIED TO GUN AND AMMUNITION RECOGNITION FROM SHOOTING SOUNDS**” [6]. Un proyecto que se basa en el reconocimiento de arma en base al sonido de los disparos en videos, a través de redes neuronales, tomando tres tipos de armas (*Star gun, Astra revolver, Police shotgun*) y dos tipos de munición, disparando en ambiente abierto y cerrado, tomando características de la frecuencia como: los formantes, la Transformada Rápida de Fourier, normalizando las bandas de frecuencia, la máxima energía, la densidad espectral en el dominio del tiempo y la frecuencia máxima, mientras que en este proyecto tecnológico en particular se tomó solo un tipo de arma (*Prieto Berreta 9mm*), no se tomó en cuenta el tipo de munición, se recolectaron videos con sonidos de disparos en ambiente abierto y cerrado, se tomaron los coeficientes arrojados por la Transformada Rápida de Fourier, se normalizaron las bandas de frecuencia, pero en este caso se promediaron las bandas de frecuencia en una escala logarítmica y de 50 en 50 Hz. El artículo [6] obtiene un resultado de reconocimiento del tipo de arma en base al sonido de entre un 96 y 98 por ciento, mientras que en este proyecto terminal los métodos de *clustering* solo reconocen entre 14 y 27 sonidos de disparo real y el método SVM con diferente *kernel* reconoce entre 60 y 98 por ciento de los sonidos de disparo real.
- “**Gunshot Detection System for JTRS Radios**” [7]. Un sistema que toma los radios JTRS para servir como sensores y capturar la señal de los disparos al igual que la posición del disparador utilizando el algoritmo filtros de Kalman, separando el disparo y la onda de choque, mientras que en este proyecto tecnológico se recabaron videos con sonidos de disparo real producidos por arma de fuego, disparos de salva producidos por arma de fogeo, efectos de sonido y explosivos, se usaron los métodos de *K-Means, K-Medoids*, Jerárquico, jerárquico empleando como distancia de similitud el *DTW* y el método *SVM* con la finalidad de reconocer el sonido de disparo real.
- “**Acoustic Shooter Localization with a Minimal Number of Single-Channel Wireless Sensor Nodes**” [8]. El sistema PinPtr se basó en un gran número de sensores en red para detectar la ubicación de la onda de choque, distancia y velocidad del disparo.
- “**A Naïve Mid-level Concept-based Fusion Approach to Violence Detection in Hollywood Movies**” [9]. Existe una gran diferencia entre ambos proyectos, ya que en el proyecto [9], busca detectar escenas violentas en las producciones de Hollywood, utilizando redes neuronales, redes bayesianas, máquinas de soporte vectorial y extrayendo características audio-visuales como: la longitud de tiro, el habla, la música, la intensidad de movimiento, en donde obtienen como resultado una precisión de reconocimiento del sonido de disparo aproximadamente del 10 %, mientras que en este presente proyecto tecnológico, no se realiza una clasificación a partir de audio visuales, si no, que se enfoca

más a reconocer el sonido de disparo real en base características del sonido por medio de los métodos *K-Means*, *K-Medoids*, jerárquico, jerárquico con distancia *DTW* y el *SVM*, en donde los métodos de *clustering* solo reconocieron entre 14 y 27 sonidos de disparo real, por otro lado el método SVM obtiene una precisión aproximadamente del 90% por ciento de reconocimiento del sonido de disparo real.

- “*Neural network classification of gunshot using spectral characteristics*” [10]. Un proyecto el cual emplea redes neuronales como clasificador, extrae características de la densidad espectral, realizan recopilación de disparos producidos por armas reales en distintas condiciones, colocan micrófonos de 10 a 25 metros de distancia desde la posición del disparo.
- “*ShotSpotter*” [11]. Se basa en sensores acústicos los cuales están emparejados con el software de análisis de audio que identifica la firma única de disparos. A diferencia del proyecto [11], es que los datos, se recopilaban del audio de varios videos y se realizara clasificación para detectar si hay presencia de sonidos de disparos en los videos y si estos disparos producidos por armas de fuego son reales o falsos.

1.2 Justificación

La finalidad de la vigilancia en cualquiera de sus modalidades es mantener el orden y la seguridad en los establecimientos públicos y privados de nuestro país y del resto del mundo. Hasta hoy en día, han existido situaciones no deseables de violencia como: el robo a mano armada, enfrentamientos entre las fuerzas del orden y delincuentes, entre otros en distintos lugares de nuestro país.

En el uso de la video vigilancia hay ocasiones en las que no se captura la imagen donde se acontece el acto violento o delictivo, ya sea porque la cámara de video esté colocada en un sitio no estratégico o la resolución del video es baja, no obstante, los videos registran el sonido. Aquí es donde recae la importancia de realizar éste proyecto tecnológico ya que se propone crear y obtener como resultado un programa con la capacidad de clasificar y discriminar el sonido de un disparo entre otros sonidos similares, aplicando distintas técnicas de minería de datos. Al distinguir un sonido de disparo, el proyecto podrá ayudar al proceso de detección de situaciones violentas en contenido audio-visual tales como los videos de seguridad y aportar un avance a los siguientes niveles de los sistemas de vigilancia. En los cuáles también se pueda emitir una señal al detectar un sonido de disparo notificando a los servicios de seguridad o vigilancia para que atiendan el evento lo más pronto posible.

1.3 Objetivos

1.3.1 Objetivo general

Crear un programa que logre distinguir el sonido de un disparo entre otros sonidos similares utilizando técnicas de minería de datos

1.3.2 Objetivos específicos

- **Recolección y clasificación manual de videos:** se cumplió este objetivo recopilando una serie de videos descargados del reproductor de videos en línea *YouTube*, estos videos contienen sonidos de disparos reales producidos por arma de fuego, disparos de salva producidos por arma de fogeo, explosivos y efectos de sonido, cabe mencionar que para estos tres sonidos (disparo real por arma de fuego, disparo de salva por arma de fogeo y efecto de sonido), solo se tomó el sonido de disparo que producía un solo tipo arma identificada cómo *Prieto Beretta 9mm*, ya teniendo conocimiento de este tipo de arma por experiencia propia. Los sonidos tomados de estos videos fueron en ambientes abiertos y cerrados, por último ordenando los videos descargados por distintas categorías (disparo real por arma de fuego, disparo de salva por arma de fogeo, efecto de sonido y explosivos).
- **Extraer el audio de los videos:** este objetivo se cumplió extrayendo el audio de los videos a través del convertidor en línea “*ONLINE-CONVERT*” [14], convirtiendo los videos en archivos de audio *WAV*.
- **Obtener los siguientes parámetros del audio:** frecuencia, intensidad, amplitud y coeficientes de la transformada rápida de Fourier: este objetivo se cumplió obteniendo los parámetros aplicando la Transformada Rápida de Fourier través del software “*Praat*” [15].
- **Pre-procesamiento de los archivos:** se cumplió el objetivo normalizando y promediando los datos que almacena cada uno de los archivos, en base a una escala logarítmica y de 50 en 50 Hz, por medio de dos programas creados en lenguaje C++. Estos códigos se pueden ver en el *Apéndice A*.
- **Almacenar los parámetros obtenidos del audio en un archivo de texto:** se cumplió el objetivo recabando toda la información de los archivos y almacenando toda la información en dos archivos de tipo texto, uno que almacena todos los datos promediados por bandas de frecuencia de 50 en 50 Hz y otro que almacena todos los datos promediados por bandas de frecuencia en escala logarítmica.
- **Clasificar los datos almacenados en el archivo de texto mediante técnicas de minería de datos:** se cumplió el objetivo usando y aplicando los algoritmos del

paquete R [32], empleando como datos de entrada los archivos generados y mencionados en el objetivo anterior para su clasificación de los datos.

Utilizando las funciones del paquete R: kmeans, hclust, PAM y los paquetes DTW [29] y SVM [30].

- **Analizar y comparar los resultados de las técnicas de minería de datos utilizadas:** se cumplió el objetivo analizando los algoritmos de *clustering* a través del paquete *cValid* [31], para el análisis de los resultados de *SVM* se usó la matriz de confusión, la función `proc.time` del paquete R para obtener el tiempo de la ejecución de cualquier método y por último la comparación de todos los resultados obtenidos.

Capítulo 2

Marco Teórico

2.1 Procesamiento de señales

El procesamiento de señales permite llevar las señales a una forma en el cual se pueda interpretar mejor. Una de las herramientas para el procesamiento de señales lo proporciona el análisis de Fourier, el cual permite cambiar el dominio en el que se encuentran los datos [22].

2.1.1 Transformada rápida de Fourier

El análisis de Fourier de una función periódica se refiere a la extracción de series de senos y cosenos que cuando se superponen, reproducen la función original.

La Transformada Rápida de Fourier (en inglés *Fast Fourier Transform*, abreviada *FFT*) es un algoritmo que convierte una función compleja, muestreada en tiempo, en una función muestreada en dominio de la frecuencia y que también es compleja [25].

La *FFT* permite saber dónde se concentra la energía de una señal analizándola en el dominio de la frecuencia.

Lo que hace la *FFT* es descomponer una señal que varía en el tiempo en una descomposición de varias señales que se comportan como ondas sinusoidales pero con diferente frecuencia, que sumadas se obtiene la señal original (*ver Figura 2.1*).

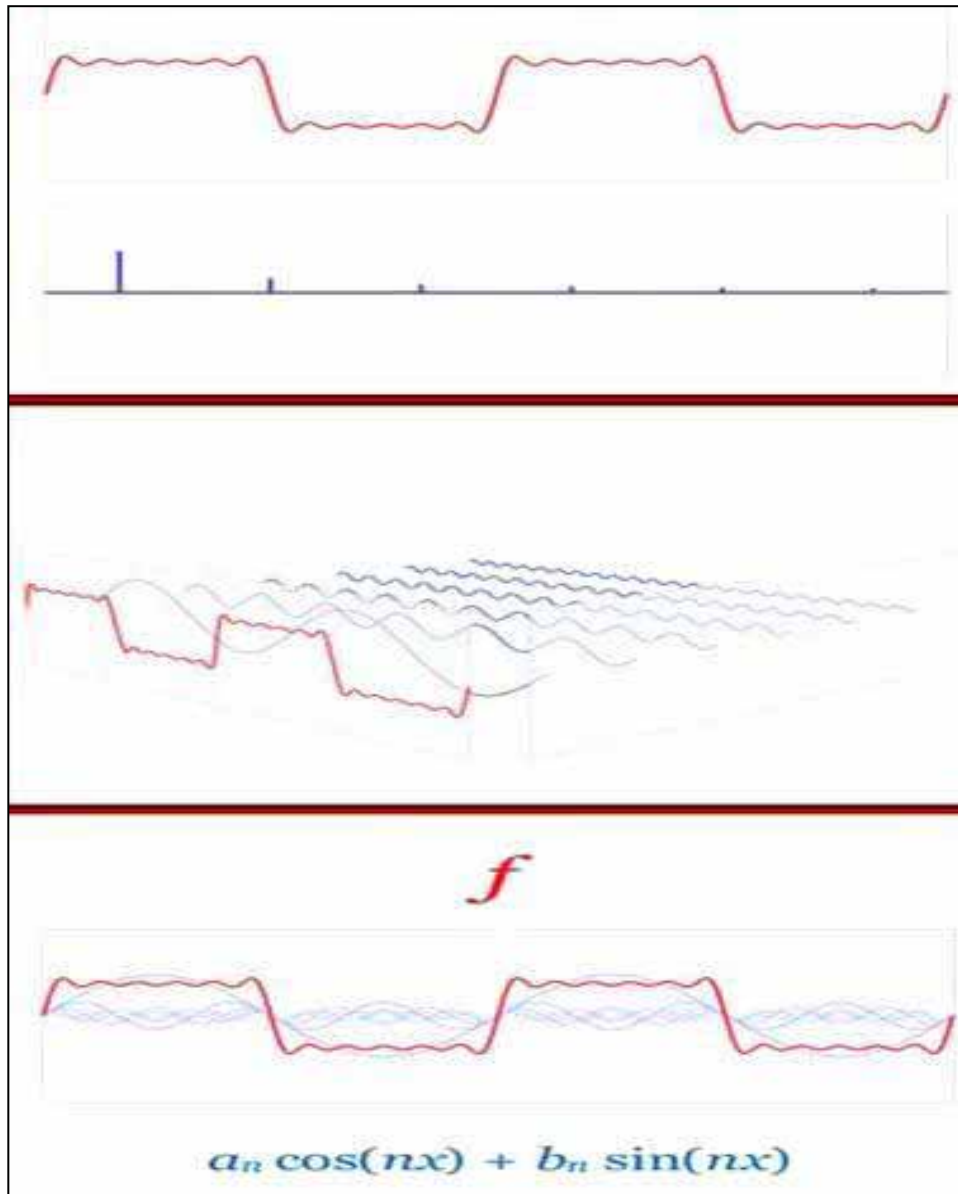


Figura 2.1: Comportamiento de la transformada de Fourier por sus siglas en inglés FFT.
Fuente [25]

La *FFT* se obtiene de simplificar el cálculo de la Transformada Discreta de Fourier (abreviada DFT) que se define mediante la *Ecuación 2.1*

$$X(k) = \sum_{n=0}^{N-1} x(n)W_N^{kn}$$

Ecuación 2.1

La **FFT** es el algoritmo para el cálculo de la **DFT** desarrollado por Cooley y Tukey [34] en 1965 el cual reduce el número de cálculos necesarios para N puntos de $2N^2$ a $2N \log N$, donde log es la base 2 de logaritmo, lo cual reduce el número de sumas y multiplicación respecto al algoritmo original.

Los valores de $x(k)$ son las muestras de la señal en el dominio del tiempo, $X(n)$ son las muestras de la señal en el dominio frecuencial, N es el número de puntos y W_N son los factores de giro.

Partiendo de la *Ecuación 2.1*

y asumiendo que el número de datos N es par, se descompone la sumatoria en sus términos pares e impares (*ver Ecuación 2.2*):

$$\begin{aligned} X(k) &= \sum_{n=0}^{(N/2)-1} x(2n)W_N^{k2n} + \sum_{n=0}^{(N/2)-1} x(2n+1)W_N^{k(2n+1)} \\ &= \sum_{n=0}^{(N/2)-1} x(2n)W_N^{k2n} + W_N^k \sum_{n=0}^{(N/2)-1} x(2n+1)W_N^{k(2n)} \end{aligned}$$

Ecuación 2.2: Ecuación “FFT”

Sea ahora: $x[2n]=f[n]$ y $x[2n+1]=g[n]$, donde el nuevo $n \in [0, N/2-1]$, y hacer que $W_N^{k2n} = e^{-j2nN^{k2n}} = e^{-j2nN/2^{kn}} = W_{N/2}^{kn}$, y definirse como la *Ecuación 2.3* y *Ecuación 2.4*:

$$F(k) = \sum_{n=0}^{N/2-1} f[n]W_{N/2}^{kn}$$

Ecuación 2.3: Términos pares

$$G(k) = \sum_{n=0}^{N/2-1} g[n]W_{N/2}^{kn}$$

Ecuación 2.4: Términos impares

Siendo F(k) y G(k) también periódicas (ver Ecuación 2.5)

$$F(k + N/2) = \sum_{n=0}^{N/2-1} f[n]W_{N/2}^{(k+N/2)n} = \sum_{n=0}^{N/2-1} f[n]W_{N/2}^{kn} e^{-j\frac{2\pi}{N/2}(N/2)n} = F(k)$$

Ecuación 2.5

Análogamente para G(k); con N/2 como periodo mínimo. Entonces el problema de calcular la *DFT* para N datos se redujo ahora a calcular dos *DFT* para N/2 datos cada una. Nuevamente se asume que N/2 es par y ahora para F(k) (ver *Ecuación 2.6*

)

$$\begin{aligned} F(k) &= \sum_{n=0}^{N/2-1} f[n]W_{N/2}^{kn} = \sum_{n=0}^{N/4-1} f[2n]W_{N/2}^{k2n} + \sum_{n=0}^{N/4-1} f[2n+1]W_{N/2}^{k(2n+1)} \\ &= \sum_{n=0}^{N/2-1} f[n]W_{N/2}^{kn} = \sum_{n=0}^{N/4-1} f_f[n]W_{N/2}^{k2n} + W_{N/2}^K \sum_{n=0}^{N/4-1} g_f[n]W_{N/2}^{k2n} \end{aligned}$$

Ecuación 2.6

$$n \in [0, N/4 - 1]$$

Donde f_f y g_f son lo mismo para $f[n]$ que para $x[n]$.

De igual forma para $g[n]$ (ver *Ecuación 2.7*

$$\begin{aligned} G(k) &= \sum_{n=0}^{N/4-1} g[2n]W_{N/2}^{k2n} + \sum_{n=0}^{N/4-1} f[2n+1]W_{N/2}^{k(2n+1)} \\ &= \sum_{n=0}^{N/4-1} f_g[n]W_{N/2}^{k2n} + W_{N/2}^K \sum_{n=0}^{N/4-1} g_g[n]W_{N/2}^{k2n} \end{aligned}$$

Ecuación 2.7

$$n \in [0, N/4 - 1]$$

Es decir que ahora $f[n]$ y $g[n]$ se realizan cuatro *DFT* de longitud N/4.

Entonces podrían realizarse múltiples divisiones del intervalo [0,N-1] mientras se pueda dividir N entre 2

El método se puede generalizar:

Sea \bar{x} un vector de datos, de longitud $N-2^m$. Entonces sobre el intervalo [0,N-1] se puede realizar m particiones como se mostró anteriormente hasta llegar a una *DFT* de longitud dos, esta es la unidad básica de la *FFT* conocida como mariposa (o *butterfly* en inglés) en

donde solo se necesitará una multiplicación y dos sumas complejas como se muestra en la *Figura 2.2*:

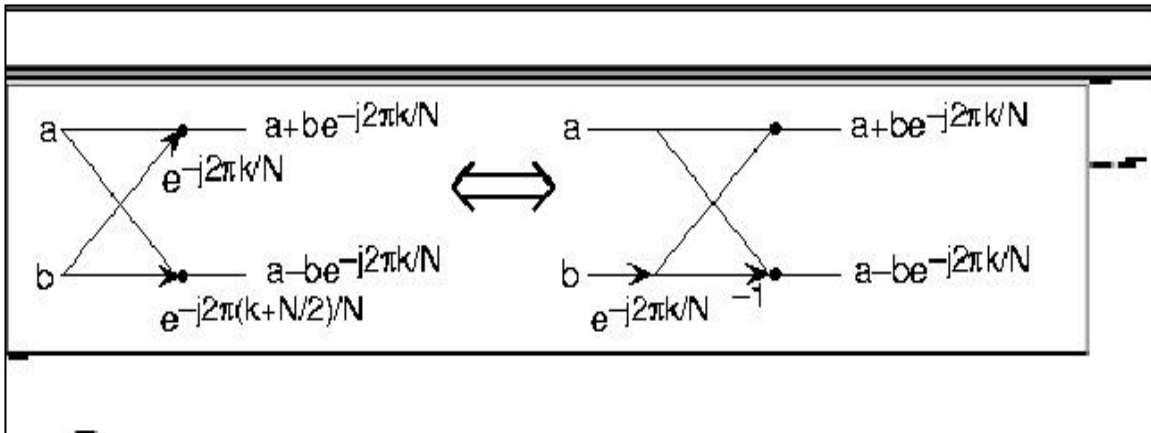


Figura 2.2: Los elementos computacionales de la transformada rápida de Fourier es la mariposa. Toma dos números complejos. Representados por a y b, y forma las cantidades mostradas. Cada mariposa requiere una multiplicación compleja y dos sumas complejas. Fuente [26].

n	0	1	2	3	4	5	6	7
x[n]	1	2	3	4	5	6	7	8

Desarrollando la *Ecuación 2.8*:

$$X(k) = \sum_{n=0}^7 x[n]W_8^{kn} = x[0]W_8^{0k} + x[1]W_8^k + x[2]W_8^{2k} + x[3]W_8^{3k} + x[4]W_8^{4k} + x[5]W_8^{5k} + x[6]W_8^{6k} + x[7]W_8^{7k}$$

Ecuación 2.8

Donde se cumple la *Ecuación 2.9*

para w

$$W_8^{2nk} = e^{-j\frac{2n}{8}2nk} = e^{-j\frac{2n}{8}nk} = W_4^{nk}$$

$$W_8^{(2n+1)k} = e^{-j\frac{2n}{8}(2n-1)k} = e^{-j\frac{2n}{4}nk} e^{-j\frac{2n}{8}k} = W_4^{nk} W_8^{nk}$$

Ecuación 2.9

Agrupando términos pares con impares (ver *Ecuación 2.10*)

)

$$X(k) = (x[0]W_4^{0k} + x[2]W_4^k + x[4]W_4^{2k} + x[6]W_4^{3k}) \\ + (x[1]W_4^{0k} + x[3]W_4^k + x[5]W_4^{2k} + x[7]W_4^{3k})W_8^k$$

Ecuación 2.10

Tomamos: $x[2r] = f[r]$ y $x[2r + 1] = g[r]$, $r=0,1, 2, 3$ y luego (ver Ecuación 2.11)

$$X(k) = (f[0]W_4^{0k} + f[1]W_4^k + f[2]W_4^{2k} + f[3]W_4^{3k}) \\ + (g[0]W_4^{0k} + g[1]W_4^k + g[2]W_4^{2k} + g[3]W_4^{3k})W_8^k$$

Ecuación 2.11

Se puede realizar una nueva factorización (ver Ecuación 2.12)

$$W_4^{2nk} = e^{-j\frac{2n}{4}2nk} = e^{-j\frac{2n}{2}nk} = W_2^{nk} \\ W_4^{(2n+1)k} = e^{-j\frac{2n}{4}(2n-1)k} = e^{-j\frac{2n}{2}nk} e^{-j\frac{2n}{4}k} = W_2^{nk} W_4^{nk}$$

Ecuación 2.12

Entonces (ver Ecuación 2.13)

$$X(k) = ((f[0] + f[2]W_4^{2k}) + (f[1] + f[3]W_4^{2k})W_4^k) \\ + ((g[0] + g[2]W_4^{2k}) + g[1] + g[3]W_4^{3k})W_4^k)W_8^k$$

Ecuación 2.13

Nuevamente: $f[2s] = f_f[s]$ y $f[2s + 1] = g_g[s]$, $s=0,1$ luego (ver Ecuación 2.14)

$$X(k) = ((f_f[0] + f_f[1]W_2^k) + (f_g[0] + f_g[1]W_2^{2k})W_4^k) \\ + ((g_f[0] + g_f[1]W_2^k) + (g_g[0] + g_g[1]W_2^k)W_4^k)W_8^k$$

Ecuación 2.14

Remplazando los índices iniciales (ver Ecuación 2.15)

$$X(k) = ((x[0] + x[4]e^{-\pi k}) + (x[2] + x[6]e^{-\pi k})e^{-j\frac{\pi}{2}k}) + ((x[1] + x[5]e^{-\pi k}) + (x[3] + x[7]e^{-\pi k})e^{-j\frac{\pi}{2}k})e^{-j\frac{\pi}{4}k}$$

Ecuación 2.15

Implica que (ver Ecuación 2.16):

$$\Rightarrow X(k) = ((x[0] + x[4](-1)^k) + (x[2] + x[6](-1)^k)(-j)^k) + ((x[1] + x[5](-1)^k) + (x[3] + x[7](-1)^k)(-j)^k)e^{-j\frac{\pi}{4}k}$$

Ecuación 2.16

Esta forma general se obtiene el k-esimo termino para N=8, puede verse que se deberán ejecutar cuatro **DFT** de longitud 2, y los resultados que se obtengan se usaran para ejecutar dos **DFT** de longitud 4, para lo cual se realizaron tres particiones. Para interpretar lo que se indica en la ecuación 2.5 (ver la Figura 2.3), donde cada flecha en diagonal representa una suma y las acompañan sus factores multiplicativos [26]:

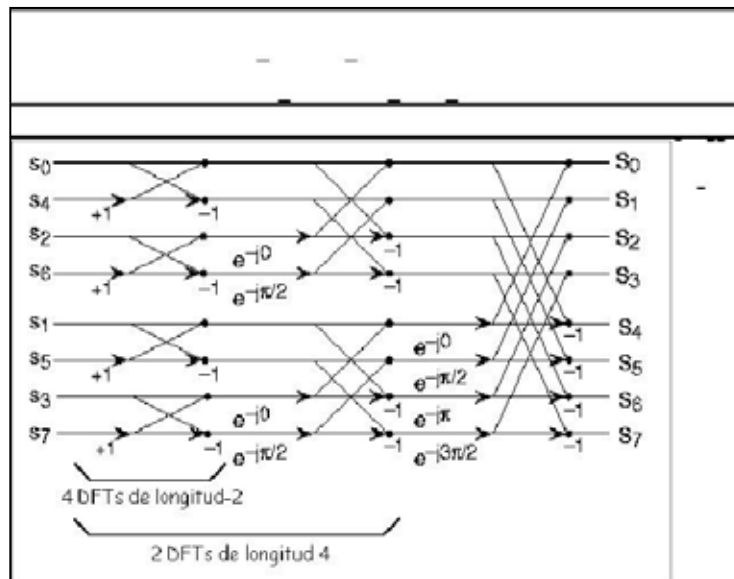


Figura 2.3: El número total de cálculos que se realizados serán N=8 para cada etapa y $\log_2 N=3$ etapas, haciendo el numero de procesos básicos de $(8)(3)=N \log_2 N$. Fuente [26]

La transformada rápida de Fourier (**FFT, Fast Fourier Transform**) es un algoritmo para el cálculo de la "**DFT**" desarrollado por Tukey y Cooley [34] en 1965 el cual reduce el número de cálculos necesarios para N puntos de $2N^2$ a $2N \log_2 N$, donde log es la base 2 de logaritmo, lo cual reduce el número de sumas y multiplicación respecto al algoritmo original.

2.1.2 Normalización de señales

El proceso de normalización consiste en llevar una señal pasada como entrada a un nivel de referencia común, con el fin de prepararla para su posterior procesamiento y análisis. Esto se hace dado que señales de diferentes fuentes pueden tener diferentes amplitudes y así reducir los efectos de estas variaciones sobre los resultados de salida, así como para capturar eventos o cambios súbitos de la señal de entrada.

2.1.3 Escala logarítmica

La escala logarítmica es una escala de medida, que se utiliza para representar más cómodamente cantidades físicas que tienen valores muy grandes y muy pequeños. Una escala logarítmica comprime los valores de manera que se puedan observar adecuadamente.

Cuando representamos una magnitud sobre una recta utilizando una escala logarítmica, lo que representamos no es propiamente la magnitud, sino el logaritmo de dicha magnitud. Por ello, si utilizamos logaritmos en base 10, que es lo más frecuente, un salto de una unidad en la escala logarítmica equivale a multiplicar por 10 el valor real de la magnitud. De ese modo, las marcas 1, 2, 3, que leemos en nuestra escala representarán realmente los valores 10, 100, 1000... de la magnitud representada. Esa es precisamente la gran ventaja de una escala logarítmica, ya que los logaritmos crecen mucho más despacio que los números a los que se aplican.

2.2 Técnicas de minería de datos

Las técnicas de minería de datos son usadas para la obtención de patrones en los datos y para la extracción de información valiosa en el campo de la Ingeniería del Software.

Entre las técnicas más representativas se encuentran los árboles de decisión, redes neuronales, modelos estadísticos y algoritmos de agrupamiento.

2.2.1 Algoritmos de agrupamiento (*clustering*)

El agrupamiento de los datos o “*clustering*” puede definirse como un método para clasificar o describir propiedades de un sistema a partir de un conjunto de datos adquiridos o representativos de dicho sistema.

Los métodos de agrupamiento no paramétricos pueden dividirse en tres grupos fundamentales: jerárquicos, particionales y basados en densidad.

Los algoritmos jerárquicos son aquellos en los que se toma el conjunto de datos y se va uniendo o dividiendo en dos grupos por cada nivel según sea el caso, si es un aglomerativo o divisivo.

Los algoritmos particionales son los que realizan una división inicial de los datos en grupos y posteriormente los objetos se van moviendo de un grupo a otro según se optimice mejor alguna función objetivo.

Los algoritmos basados en densidad enfocan el problema de la división de una base de datos en grupos teniendo en cuenta la distribución de densidad de los puntos, de modo tal que los grupos que se forman tienen una alta densidad de puntos en su interior mientras que entre ellos aparecen zonas de baja densidad.

2.2.1.1 Grupo (*Cluster*)

Es una colección o subconjunto de objetos que son similares entre sí y colectivamente pueden tratarse como un grupo tal que la distancia entre dos objetos cualesquiera pertenecientes al grupo es menor que la distancia entre cualquier otro objeto no localizado dentro del grupo. En el contexto de minería de datos se usa el nombre *cluster* para referirse a esos grupos [35].

2.2.1.2 Agrupamiento (*Clustering*)

Es el proceso de dividir un conjunto de objetos y agruparlos en subconjuntos o *clusters* de tal forma que, los objetos de un mismo *cluster* tienen una alta similitud. Se pueden tener diferentes formas de hacer los:

- Bien separados: Un *cluster* es un conjunto de puntos en el que cualquier punto en el *cluster* es más cercano a otro punto perteneciente al mismo *cluster* que cualquier otro punto que se encuentre en un *cluster* diferente.
- Basados en el centro: Es un conjunto de objetos en el que un objeto está más cerca al centro al *cluster* que pertenece, que al centro de otro *cluster*. El centro de un *cluster* frecuentemente es llamado **centroide**, el promedio de todos los puntos en el *cluster* o el “*medoid*”, el punto más representativo del *cluster*.
- *Clusters* contiguos: Es un conjunto de puntos donde un punto en el *cluster* está más próximo a otro punto o puntos encontrados en el mismo *cluster* que a cualquier otro punto que no pertenezca al *cluster*.
- Basados en densidad: un *cluster* es una región densa de puntos, separados por regiones de baja densidad, de otras regiones de alta densidad.

Se usan cuando los *clusters* son irregulares o entrelazados, y cuando se presenta ruido y datos atípicos.

- Agrupación jerárquica: los objetos que pertenecen a un grupo hijo también pertenecen al grupo de los padres.
- **Clusters** conceptuales: de propiedad compartida o **clusters** conceptuales. Son **clusters** que tienen propiedad compartida o representan un concepto particular.
- De función objetivo: son **clusters** que minimizan o maximizan una función objetivo. Enumeran todas las posibles formas a dividir los puntos dentro de un **cluster** y evalúan la “bondad” de cada conjunto potencial de **clusters** usando una función objetivo dada (NP Hard).

2.2.1.3 K-medias (*K-Means*)

El algoritmo empieza definiendo centros para los futuros **cluster** a formar, asignando a cada punto el centro más cercano. Una vez que se tiene un estado inicial o asignación para los datos, el algoritmo avanza conforme se van calculando los nuevos centros dentro del grupo (**cluster**) [21].

En otras palabras, es un método de agrupamiento, que tiene como objetivo la partición de un conjunto de n objetos u observaciones en k grupos, en el que cada objeto pertenece al grupo más cercano a la media.

Se toma como parámetro k que es el número de **clusters** que forma.

Selecciona k elementos aleatoriamente, los cuales representan el centro o media de cada **cluster**¹. A cada objeto restante se le asigna el **cluster** con el cual más se parece, basándose en una distancia entre el objeto y la media del **cluster**. Después calcula la nueva media del **cluster** e itera hasta no cambiar de medias.

- Selecciona k objetos aleatoriamente
- Re(asigna) cada objeto al clúster mas similar con el valor medio
- Actualiza el valor de las medias de los clústeres

Se utiliza una medida de similitud basada en el error cuadrático (*ver Ecuación 2.17*)

$$E = \sum_{i=1}^k \sum_{p \in C_i} |p - m_i|^2$$

Ecuación 2.17

Donde: P representa el objeto y m_i a la media del **cluster** C_i (ambos son objetos multidimensionales).

¹ El termino clúster (del inglés cluster) es un grupo, racimo, o conjunto.

El algoritmo *K-Means* se considera un método de reasignación ya que admite que un objeto asignado a un grupo en un paso del proceso anterior sea reubicado en otro grupo en un paso posterior si esto optimiza el criterio de selección. El proceso concluye cuando no quedan individuos cuya reasignación permita optimizar el resultado [23].

2.2.1.4 Jerárquico (*Hierarchical*)

Definición: La agrupación se realiza mediante procesos con fases de agrupación o desagrupación sucesivas. El resultado final es una jerarquía de unión completa en la que cada grupo se une o separa en una determinada fase [19].

El *clustering*² jerárquico es un algoritmo de agrupamiento de aglomeración que produce un diagrama llamado dendograma³ que se puede cortar a una altura elegida para producir el número deseado de grupos. Cada observación se coloca inicialmente en su propio grupo, y los grupos se unen sucesivamente juntos con el fin de su “cercanía” o similitud [13].

La “cercanía” de los grupos se determina por una matriz de similitud, y puede basarse en una variedad de métodos de aglomeración.

Los algoritmos jerárquicos se clasifican en:

Algoritmos jerárquicos divisivos: comienza el conjunto de objetos como un solo grupo y en cada iteración se va dividiendo en dos grupos y así sucesivamente en cada iteración hasta que queden tantos grupos como objetos individuales existen en el conjunto de objetos.

Algoritmos jerárquicos aglomerativos: basados en distancias comienzan considerando a cada objeto como grupos unitarios y en cada iteración se unen los dos grupos más cercanos hasta que se obtenga un único grupo.

Método Ward: Cuando se unen dos conglomerados, con independencia del método utilizado, la varianza aumenta. El método de Ward une los casos buscando minimizar la varianza dentro de cada grupo. Para ello se calcula, en primer lugar, la media de todas las variables en cada conglomerado. A continuación, se calcula la distancia entre cada caso y la media del conglomerado, sumando después las distancias entre todos los casos. Posteriormente se agrupan los conglomerados que generan menos aumentos en la suma de las distancias dentro de cada conglomerado. Este procedimiento crea grupos homogéneos y de tamaños similares [20].

2.2.1.5 K-medoides (*K-Medoids*)

Se describe como una estrategia de *clustering*, en donde se seleccionan elementos representativos (*medoids*) dentro del conjunto de datos. Posteriormente se divide el conjunto en *clusters*, estando cada uno conformado por el subconjunto de puntos más cercanos a cada uno de los elementos. Luego, estos últimos se van cambiando por otro del

² Clustering son algoritmos de agrupamiento.

³ Es un tipo de representación gráfica o diagrama en forma de árbol.

conjunto en la medida en que mejoran la solución. La función objetivo queda definida como la distancia promedio u otra medida de diferenciación entre un punto y su correspondiente *medoid* [21].

El método *PAM (Partitioning Around Medoides)* fue desarrollado por Kauffman y Rousseeuw [33]. En éste se asume que existen n objetos, y que para encontrar k agrupamientos (*clusters*), se determina un objeto representativo para cada agrupación. Tal objeto representativo, es un punto centralmente localizado dentro de un grupo, denominado medoide. Después de seleccionar los medoides de k , el algoritmo intenta analizar todos los pares posibles de objetos, tales que, cada objeto no seleccionado es agrupado con el medoide similar. La calidad de la medida de agrupamiento se calcula para cada combinación.

2.2.2 Alineamiento temporal dinámico (*DTW*, del inglés *Dynamic Time Warping*)

DTW es un algoritmo para medir la similitud entre dos secuencias, mismas que pueden variar en el tiempo o velocidad (*ver Figura 2.4*)

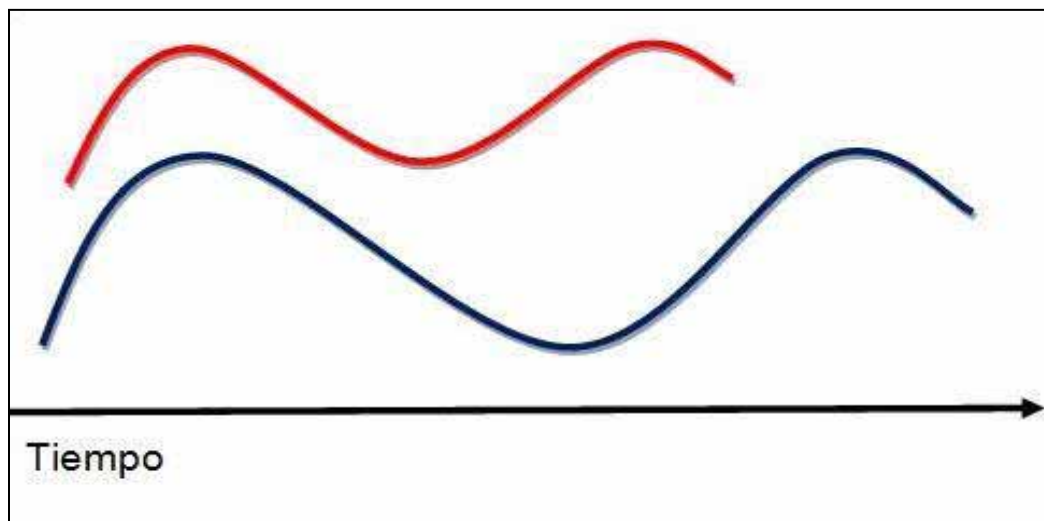


Figura 2.4: Misma señal con variación en tiempo y velocidad

Si se tienen dos señales, pero con distinta duración en el tiempo, el algoritmo **DTW** nos permite estirar y encoger ambas señales para tratar de hacerlas coincidir lo mejor posible, de esta manera, si presentan el mismo patrón ambas señales, el algoritmo lograra reconocerlas con mayor exactitud (ver *Figura 2.5*).

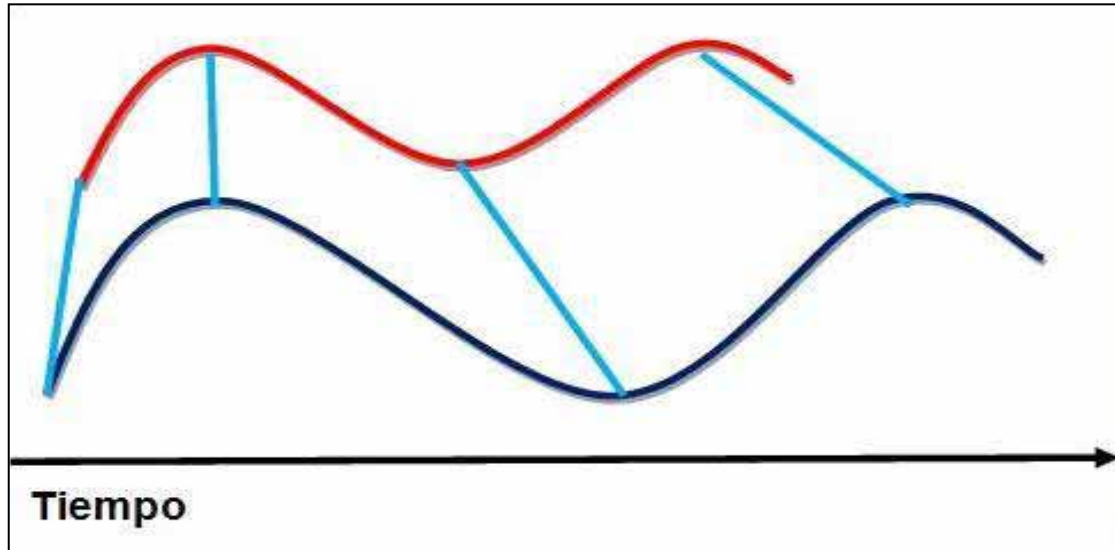


Figura 2.5: encontrando puntos de similitud entre ambas señales

Esa distancia se calcula de forma iterativa usando unas restricciones que pueden ser locales y/o globales. Las restricciones locales indican sobre qué puntos vecinos pueden conectarse para formar así el camino de alineamiento, por otro lado las restricciones globales permiten reducir el número de cálculos limitando espacio (la ventana).

Definición del algoritmo

Asumimos que queremos comparar dos series de tiempo: Un conjunto $X=(x_1, \dots, x_N)$; y un conjunto $Y=(y_1, \dots, y_M)$. para $i = 1 \dots N$ para el conjunto X y $j=1 \dots M$ para Y , i y j siendo positivos [12].

La función de similitud f es definida entre cualquier par de elementos x_i y y_j con (ver *Ecuación 2.18*)

$$d(i, j) = f(x_i, y_j) \geq 0$$

Ecuación 2.18

Siendo d la matriz cruzada de distancias entre los vectores X e Y , el algoritmo “*DTW*” recibe como entrada: los elementos de x_i e y_j únicas entradas para el cálculo de los argumentos de f .

En el núcleo de la técnica se encuentra la curva de deformación $\phi(k)$, $k = 1 \dots T$ (Ecuación 2.19)

$$\begin{aligned}\phi(k) &= (\phi_x(k), \phi_y(k)) \\ \phi_x(k) &\in \{1 \dots N\} \\ \phi_y(k) &\in \{1 \dots N\}\end{aligned}$$

Ecuación 2.19

La función de deformación ϕ_x y ϕ_y reasigna los índices de tiempo de X e Y respectivamente. Dada ϕ , calculamos el promedio acumulado de distorsión entre las series de deformación X and Y (ver Ecuación 2.20)

$$d_\phi(X, Y) = \sum_{k=1}^T d(\phi_x(k), \phi_y(k)) m_\phi(k) / M_\phi$$

Ecuación 2.20

Donde $m_\phi(k)$ es un coeficiente de ponderación por paso y M_ϕ es la constante de normalización correspondiente, que asegura que las distorsiones acumuladas son comparables a lo largo de diferentes caminos.

Para garantizar deformaciones razonables, las restricciones se imponen por lo general en ϕ . Por ejemplo, se impone la monotonía de preservar su orden en tiempo y evitar bucles sin sentido (ver Ecuación 2.21)

$$\begin{aligned}\phi_x(k+1) &\geq \phi_x(k) \\ \phi_y(k+1) &\geq \phi_y(k)\end{aligned}$$

Ecuación 2.21

La idea fundamental del **DTW** es encontrar la alineación óptima ϕ de tal manera que

$D(X, Y) = \min d_{\phi}(X, Y)$ La salida del algoritmo **DTW** puede ser considerada como la medida de estiramiento insensible de la "diferencia inherente" entre dos series de tiempo determinadas. Esta distancia tiene una aplicación directa en la agrupación jerárquica y clasificación (por ejemplo, con los clasificadores k-NN). La forma de la curva de deformación ϕ en sí proporciona información acerca de qué puntos coinciden. Una vez obtenida, la función de deformación se puede aplicar a cualquier serie de tiempo, permitiendo inspeccionar señales alineadas o medir distorsiones de tiempo.

2.2.3 Máquinas de soporte vectorial (SVM, del inglés Support Vector Machines)

Las máquinas de soporte vectorial buscan un hiperplano que separe de forma óptima a los puntos de una clase, que eventualmente han podido ser previamente proyectados a un espacio de dimensionalidad superior [16]. Sin ningún conocimiento del mapeo, la **SVM** encuentra un hiperplano óptimo utilizando el producto punto con funciones en el espacio de características que son llamadas **kernels** [17].

En ese concepto de "separación óptima" es donde reside la característica fundamental de las **SVM**: este tipo de algoritmos buscan el hiperplano que tenga la máxima distancia (margen) con los puntos que estén más cerca de él mismo. Por eso también a veces se les conoce a las **SVM** como **clasificadores de margen máximo**. De esta forma, los puntos del vector que son etiquetados con una categoría estarán a un lado del hiperplano y los casos que se encuentren en la otra categoría estarán al otro lado [16].

La mejor solución es aquella que permita un margen máximo entre los elementos de las dos categorías. Se denominan vectores de soporte a los puntos que conforman las dos líneas paralelas al hiperplano, siendo la distancia entre ellas (margen) la mayor posible [18] (ver *Figura 2.6*).

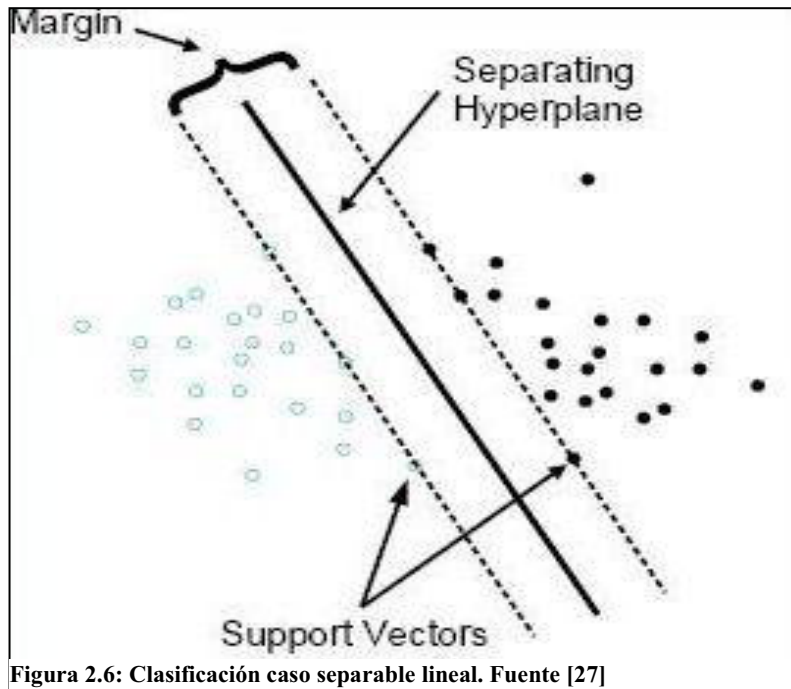


Figura 2.6: Clasificación caso separable lineal. Fuente [27]

Una **SVM** primero mapea los puntos de entrada a un espacio de características de una dimensión mayor (por ejemplo si los puntos de entrada están en \mathbb{R}^2 entonces son mapeados por la **SVM** a \mathbb{R}^3) y encuentra un hiperplano que los separe y maximice [17].

Función Kernel

Las funciones **kernel** (núcleo) tienen la utilidad de proyectar la información a un espacio de características de mayor dimensión. Es decir mapear el espacio de entradas X a un nuevo espacio de características de mayor dimensionalidad [16].

Tipos de funciones **Kernel**

- Lineal (ver Ecuación 2.22)

$$k(x, x') = (x, x')$$

Ecuación 2.22

- Función de base radial Gaussiana (Gaussian Radial Function, RBF) (ver Ecuación 2.23)

$$k(x, x') = \exp(-\sigma \|x - x'\|^2)$$

Ecuación 2.23

- Polinomial (ver Ecuación 2.24)

$$k(x, x') = (scale \cdot (x - x') + offset)$$

Ecuación 2.24

- Función Bessel (ver Ecuación 2.25)

$$k(x, x') = \frac{Bessel_{(v+1)}^n(\sigma \|x - x'\|)}{(\|x - x'\|)^{-n(v+1)}}$$

Ecuación 2.25

- Función de Base Radial laplace (Laplace Radial Basis Function, RBF) (ver Ecuación 2.26)

$$k(x, x') = \left(\sum_{k=1}^n \exp(-\sigma \|x^k - x'^k\|^2) \right)^d$$

Ecuación 2.26

- Multidimensional (ver Ecuación 2.27)

$$k(x, x') = \prod_{k=1}^n k(x^k, x'^k)$$

Ecuación 2.27

2.3 Recapitulación

En este capítulo se presentó el marco teórico. Iniciando con una breve explicación del procesamiento de señales, posteriormente una demostración del cálculo de la Transformada Rápida de Fourier. Después se introdujo una explicación de las técnicas de minería de datos, abarcando los algoritmos de **clustering** y los tipos de agrupamientos posteriormente describiendo los métodos **K-Means**, **K-Medoids**, jerárquico. Se continuó con el método **DTW** y su definición. Terminando el capítulo con una explicación de las máquinas de soporte vectorial (**SVM**, del inglés **Support Vector Machines**).

En general la razón de esta teoría descrita anteriormente conlleva al aprendizaje el cual genera un conocimiento de cómo se debe atacar, o indicar cuales son los pasos a seguir para resolver en particular este proyecto tecnológico.

Para el caso del tema del procesamiento de señales, sirvió para poder comprender cómo tratar las señales de sonido de tal forma que se puedan interpretar mejor, y una de estas herramientas es la **FFT** ya que convierte una señal muestreada en función del dominio del

tiempo a una función en el dominio de la frecuencia permitiendo conocer dónde se concentra la mayor energía o las variaciones de energía del sonido.

El tema de técnicas de minería de datos permite conocer los tipos de algoritmos y sus funciones, principalmente cómo funcionan los algoritmos de ***K-Means***, ***K-Medoids***, jerárquico, ***DTW*** y ***SVM***. Como el método ***K-Means*** selecciona k elementos aleatoriamente que representaran los centros o media de cada ***cluster*** y a cada objeto restante se le asigna el ***cluster*** con el cual más se parece, basándose en una distancia entre el objeto y la media del ***cluster***, después calculando la nueva media del ***cluster***, e itera hasta no cambiar de medias. El funcionamiento del algoritmo ***K-Medoids*** en donde también se toman objetos representativos del conjunto de datos llamados ***medoids***, después se divide el conjunto de datos estando cada uno conformado por el subconjunto de puntos más cercanos a cada uno de los elementos ***medoids***. Luego, estos últimos se van cambiando por otro del conjunto en la medida en que mejoran la solución. El funcionamiento del algoritmo jerárquico y sus tipos divisivos y aglomerativos. El funcionamiento del algoritmo ***DTW*** el cual se usa para medir la similitud de dos secuencias, misma que pueden variar en el tiempo o en velocidad y por último el funcionamiento del algoritmo ***SVM*** el cual mapea los puntos de entrada a un espacio de características de una dimensión mayor (por ejemplo si los puntos de entrada están en R^2 entonces son mapeados por la ***SVM*** a R^3) y encuentra un hiperplano que los separe y maximice.

Capítulo 3 Desarrollo del proyecto

3.1 Diseño

El proyecto consta de 7 etapas (ver Figura 3.1).

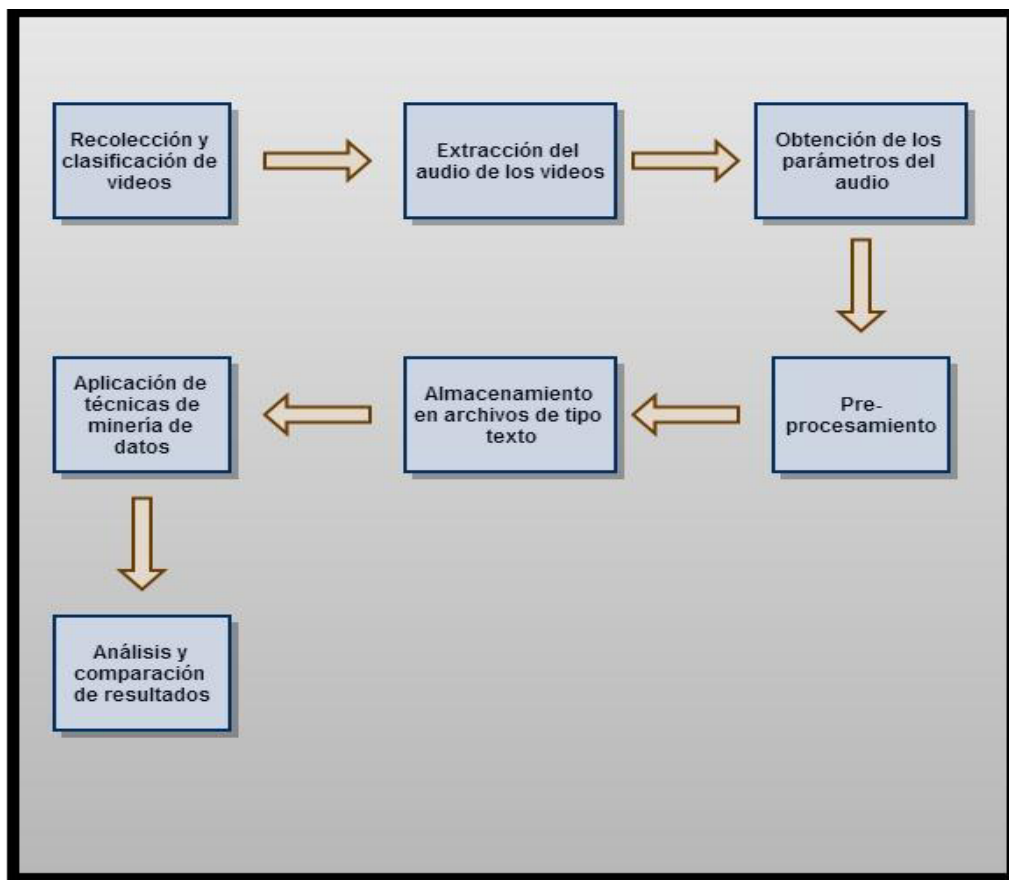


Figura 3.1: Diagrama de bloques del sistema "Detección acústica de disparos de armas de fuego"

3.2 Recolección y clasificación de videos

Debido a la dificultad de conseguir bases de datos con muestras de sonidos o videos de disparos de un solo tipo de arma o de grabarlos en vivo, se optó por la búsqueda y recolección de 150 videos del reproductor de videos en línea "*You Tube*", con sonidos de

disparos producidos por: arma de fuego, fogeo (pistola de salva), efectos de sonido y explosivos. Posteriormente se clasificaron manualmente los videos en: disparos producidos por arma de fuego real, por arma de fogeo (salva), producido por efectos de sonido y producido por explosivos.

Teniendo conocimiento por experiencia propia del tipo de arma “*Beretta*” 9mm, en particular solo se reunieron videos con detonaciones producidas por este tipo de arma (ver *Figura 3.2*). Cabe mencionar que los sonidos de estos videos fueron en ambientes abiertos y cerrados, por último ordenando los videos descargados en distintas categorías (disparo real por arma de fuego, disparo de salva por arma de fogeo, efecto de sonido y explosivos).



Figura 3.2: pistola “beretta” 9mm. Fuente [28]

3.4 Extracción del audio de los videos

Extracción del audio de los videos utilizando el software “*ONLINE-CONVERTER*” [14]. Es un convertidor de archivos online que nos permite la conversión de los archivos multimedia a otro formato, el cual se utilizó para convertir los videos en archivos de audio .WAV muestreados a 44100 Hz.

3.5 Obtención de los parámetros del audio

Se tomaron cada uno de los archivos de audio .WAV y se analizó la señal de disparo contenida en ellos por medio del software “*Praat*” [15], para la extracción de aquellos parámetros considerados, que nos permitieran diferenciar el sonido de un disparo producido por un arma de fuego real (ver *Figura 3.3*), entre otros similares como: el sonido producido

por una arma de salva (ver Figura 3.4), por un efecto de sonido (ver Figura 3.5), o por producido por una explosión (ver Figura 3.6),

Los parámetros considerados a extraer en base al análisis espectral fueron: los coeficientes de la “*FFT*”: frecuencia, amplitud y intensidad. Y trabajar en el dominio de la frecuencia ya que muestra el cambio de fase o la magnitud de una señal.

Los parámetros extraídos fueron almacenados en archivos.txt.

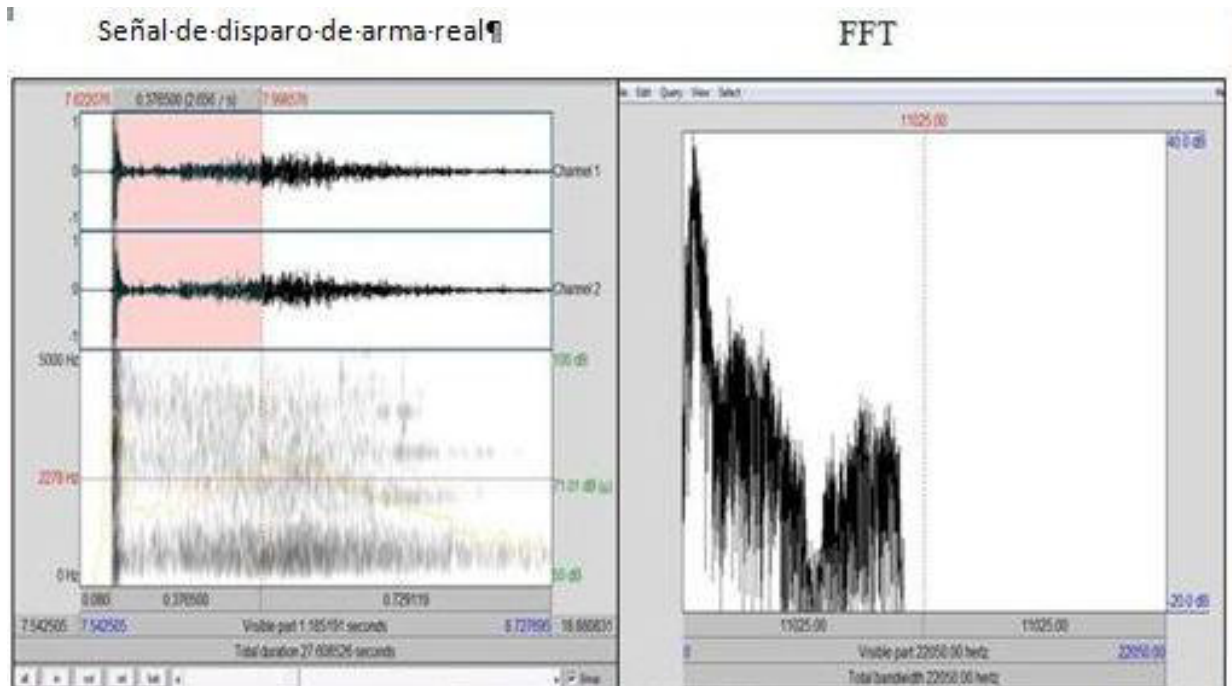


Figura 3.3: Señal de sonido de disparo producido por un arma de fuego real y la aplicación de la transformada de Fourier (FFT) sobre la señal de sonido

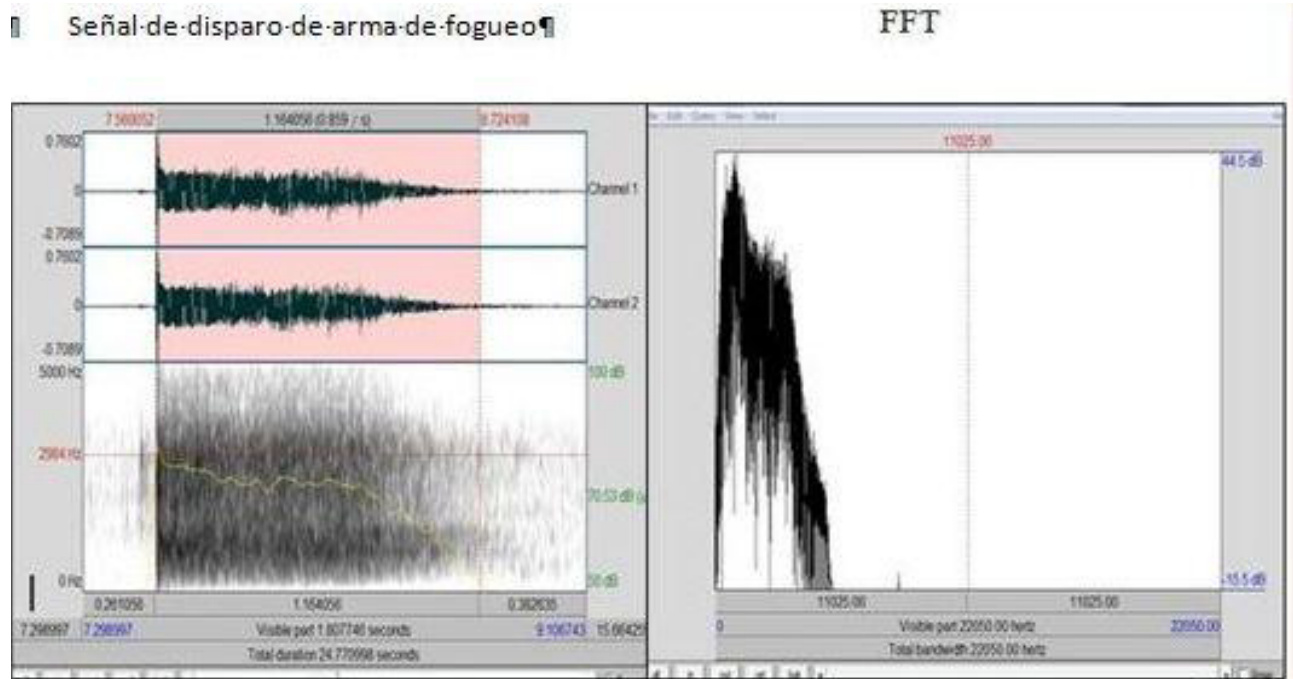


Figura 3.4: Señal de sonido de disparo producido por un arma de fuego y la aplicación FFT sobre la señal de sonido

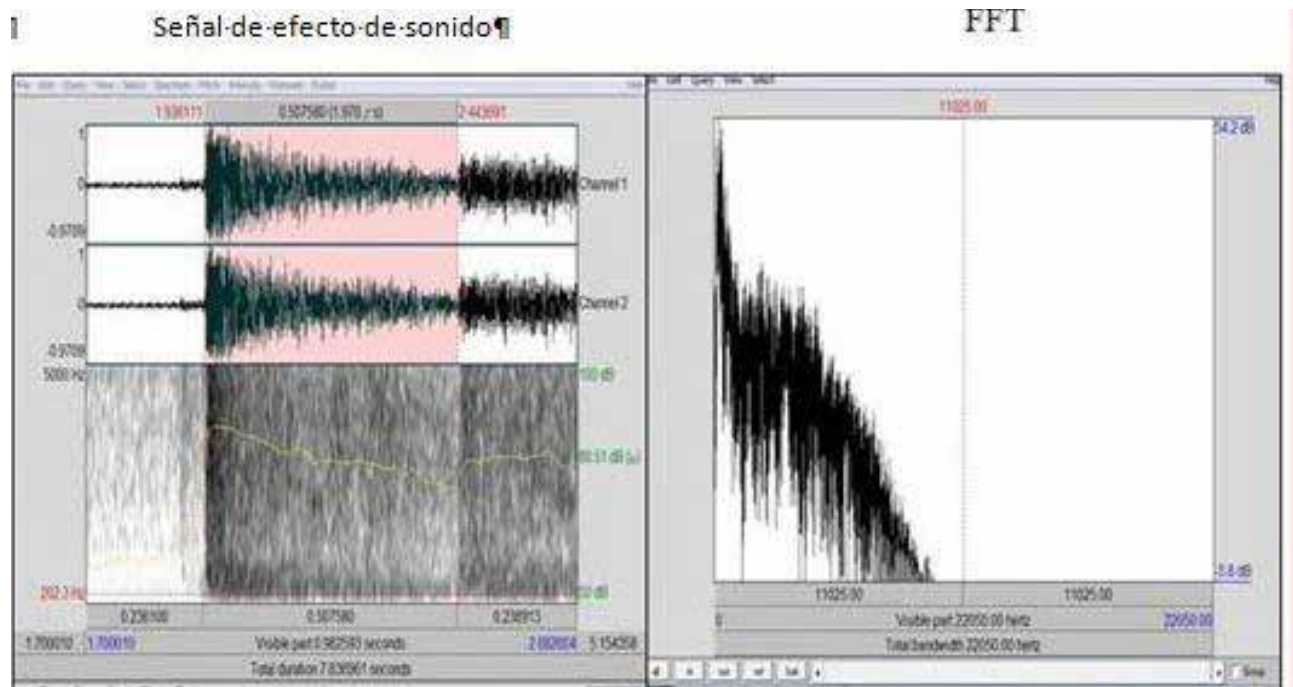


Figura 3.5: Señal de sonido producido por efectos de sonido y la aplicación de la FFT sobre la señal del sonido

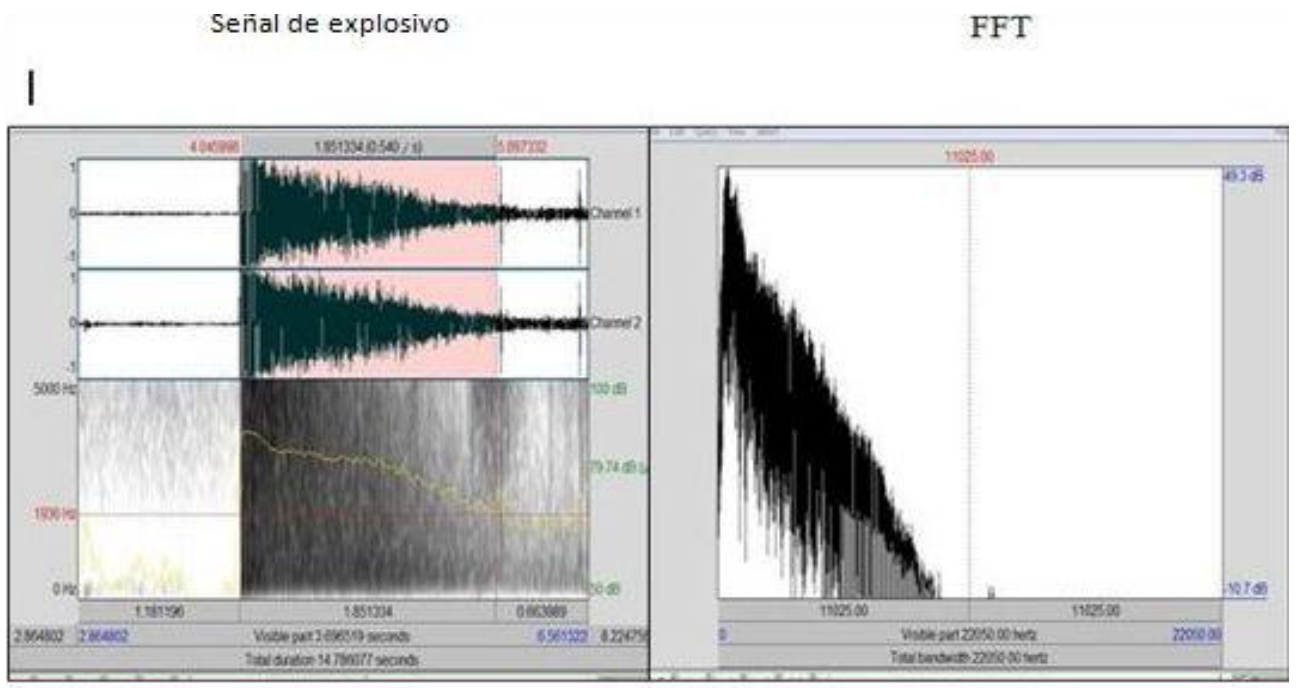


Figura 3.6: Señal de sonido producido por sonido de un explosivo y la aplicación de la FFT sobre la señal de sonido

3.6 Pre-procesamiento

Debido a la gran dimensión de datos extraídos y obtenidos a través de la *FFT*. En esta etapa se exploraron dos formas para tratar los parámetros obtenidos por la *FFT*:

- Promediar la frecuencia por bandas en una escala logarítmica
- Promediar por bandas de frecuencia de 50 en 50 Hz.

Para esta etapa del pre-procesamiento se procedió a tomar cada uno de los archivos de texto, donde se almacena una observación o muestra y sus correspondientes parámetros obtenidos a través de la *FFT*.

Para obtener el promedio por bandas de frecuencia de 50 en 50 Hz y en base a una escala logarítmica fueron diseñados dos programas en lenguaje C++, que reciben como entrada un archivo.txt tantas veces como muestras se tengan. Los cuáles contienen los parámetros obtenidos por la *FFT*. El programa procesa estos datos realizando el cálculo del promedio y normaliza los parámetros almacenados en el archivo.txt de entrada y obteniendo como resultado otro archivo.txt con los valores promediados y normalizados.

Cabe mencionar que cada uno de los archivos resultado del pre-procesamiento, almacena una observación y para el caso en que fueron promediados por bandas de frecuencia de 50 en 50 Hz, estos archivos almacenarían 147 bandas de frecuencia y para el caso en que fueron promediados por bandas de frecuencia en escala logarítmica estos archivos almacenan 47 bandas de frecuencia y también tener claro que cada banda de frecuencia

independientemente como haya sido promediada, se componen por una distinta cantidad de magnitudes de la *FFT*.

3.7 Almacenamiento en archivos de tipo texto

En esta etapa se procede a recopilar todos los datos que ya fueron procesados en la etapa anterior de cada uno de los archivos que corresponden a una muestra u observación de un sonido. Por conveniencia se generaron dos archivos con extensión txt, con el propósito de que un archivo almacenase todas aquellas observaciones promediadas por bandas en escala logarítmica y el otro archivo almacenase todas las observaciones promediadas por bandas de frecuencia de 50 en 50 Hz.

Se obtuvieron dos archivos de tipo texto y a su vez almacenando un conjunto de datos:

- Un archivo almacenando 280 muestras en total, compuesto aproximadamente por el 49% de disparos producidos por arma de fuego, 16% sonidos de disparo producidos por armas de fuego, 17% de efectos de sonido, 18% de explosivos y por último promediados por bandas de frecuencia en escala logarítmica y posteriormente normalizados.
- Otro archivo almacenando 301 muestras en total, compuesto aproximadamente por el 50% de disparos producidos por arma de fuego, 17% sonidos de disparo producidos por armas de fuego, 16% de efectos de sonido, 17% de explosivos y por último promediados por bandas de frecuencia de 50 en 50 Hz.

3.8 Aplicación de las técnicas de minería de datos usando R

En esta etapa se analizan los dos archivos generados, resultado de la etapa anterior mediante la implementación de los métodos de *clustering* y el método de clasificación supervisada o predictivo.

Se utilizan los métodos de *K-Means*, *K-Medoids*, jerárquico, máquinas de soporte vectorial y el método jerárquico con distancia *DTW* del paquete R para determinar:

- Si los métodos de *clustering*, empleando los conjuntos de datos generados y obtenidos en la *sección 3.7*, son capaces de lograr separar entre los sonidos.
- Si el método de clasificación supervisada o las máquinas de soporte vectorial empleando los datos generados y obtenidos en la *sección 3.7*, son capaces de reconocer o predecir los sonidos.

Para la aplicación de los métodos de *clustering*: *K-Means*, *K-Medoids* y jerárquico del paquete R.

- En estos tres casos, a los dos conjuntos de datos obtenidos en la etapa anterior y almacenados en diferentes archivos de tipo texto, se les aplicó la distancia **Euclidiana** para la obtención de la matriz de similitudes o matriz de distancias, la cual nos indica que tan semejantes son las observaciones o muestras de sonidos entre sí.
- Ya teniendo la matriz de similitudes se procedió a implementar los tres métodos mencionados anteriormente a partir de las funciones *kmeans*, *pam* y *hclust* para la obtención de los *clusters*.
- Por cada método se iteró 10 veces por cada k-grupos, siendo k=10

Para la aplicación del método de *clustering* jerárquico con distancia *DTW* del paquete R.

- El *Dynamic Time Warping* es un método que funciona de mejor manera con series de tiempo o trabajando en el dominio del tiempo, pero en este trabajo se experimentó con los mismos datos ya obtenidos por la etapa anterior, trabajando en el dominio de la frecuencia.
- El *DTW* se utilizó para obtener la matriz de similitudes o de distancias, para indicar que tan semejantes son las muestras.
- Ya teniendo la matriz de similitudes se procedió a implementar el método jerárquico con distancia de similitudes *DTW* a partir del paquete *DTW* [29] y la función *hclust*.
- Debido a la tardanza computacional para resolver este método solo se aplicó una sola vez por cada k-grupos, siendo k=10.

Para la aplicación del método de clasificación *SVM* del paquete R.

- Se implementó el método SVM a partir paquete **e1071** [30].

El cual obtiene un modelo a partir de los parámetros: matriz de datos, *kernel* a utilizar, método y el tipo que indicara el comportamiento que tomaran las máquinas de soporte vectorial, para este caso se usó *C-classification*

Para la aplicación de las medidas de validación interna del paquete R.

- Se implementaron estas validaciones internas a partir del paquete *clValid* [31].

El cual obtiene como resultado las tres medidas de validación interna (*Connectivity*, *Dunn* y *Silhouette*)

Cabe mencionar que todos los métodos de minería de datos implementados en este proyecto, tomaron como parámetros las columnas correspondientes a las bandas de frecuencia promediadas ya sea en base a escala logarítmica o de 50 en 50 Hz y las filas representando cada una de las muestras u observaciones de cada uno de los sonidos.

Se usaron los algoritmos *K-Means*, *K-Medoids*, *Jerárquico*, *DTW* y *SVM* del paquete R [32].

3.9 Recapitulación

En este capítulo se expuso el desarrollo del proyecto, presentando el diseño que expone todas las etapas del proceso de desarrollo del proyecto, comenzando con la recolección de videos, donde se recopilan 150 videos con sonidos disparos hechos con arma de fuego real, con arma de fogeo (arma detonadora de balas de salva), sonidos producidos por explosivos y sonidos producidos por efectos de sonido. Los cuáles fueron clasificados manualmente. Después se describe la extracción del audio de los videos, a través de una conversión a archivos *.WAV*. Posteriormente se explica la obtención de los parámetros, a través de la *FFT*. Continuando después con el pre-procesamiento, donde se explica que los datos fueron normalizados y promediados por bandas de frecuencia, mediante dos programas creados en lenguaje C++. Posteriormente se describe el almacenamiento de los datos en archivos de tipo texto. Finalizando con la aplicación de las técnicas de minería de datos.

Capítulo 4 Resultados

En este capítulo se presentan los resultados obtenidos por los métodos: *K-Means*, *K-Medoids*, jerárquico, jerárquico con distancia *DTW* y las máquinas de soporte vectorial (*SVM*, del inglés *Support Vector Machines*), pero solo aquellos resultados considerados con un mejor desempeño en cuanto a reconocer o llegar a lograr distinguir los sonidos de disparos reales. Así también los resultados obtenidos al implementar las medidas de validación interna (*Connectivity*, *Dunn* y *Silhouette*).

4.1 Resultados obtenidos al usar el paquete *cValid* de R

En esta sección se presentan los resultados obtenidos al implementar las medidas de validación interna (*Connectivity*, *Dunn* y *Silhouette*) sobre los métodos: *K-Means*, *K-Medoids*, jerárquico y jerárquico con distancia *DTW*), empleando los dos conjuntos de datos mencionados en la *sección 3.7*.

Cabe mencionar que solo el índice *Silhouette* fue implementado sobre los cuatro métodos de *clustering* mencionados anteriormente

4.1.1 Resultados obtenidos al implementar las medidas de validación interna a través del paquete *cValid*.

Los resultados arrojados al implementar las medidas de validación interna (*Connectivity*, *Dunn* y *Silhouette*) sobre los métodos de *clustering* y empleando cualquiera de los dos conjuntos de datos ya sea el promediado por bandas de frecuencia en base a una escala logarítmica de 50 en 50 Hz, sugieren que el método jerárquico obtiene un valor óptimo en cuanto a *clusters* homogéneos, compactos, con alta cohesión intra-cluster. Cuando se aglomeran en 2 grupos. Sin embargo, el método *K-Means* con los datos promediados en escala logarítmica obtiene su valor óptimo al particionar en 10 *clusters*

La *Figura 4.1* muestra lo mencionado anteriormente así como el valor que toma cada uno de los índices aplicados sobre cada uno de los métodos de *clustering*. Se puede ver que el índice de *Connectivity* el cual es valuado entre el intervalo $[0, \infty)$ y se considerará el óptimo aquel valor más pequeño, señalando así que el valor óptimo lo obtiene el método jerárquico formando 2 *clusters* (ver *Figura 4.2*) y empleando cualquiera de los dos conjuntos de datos. El índice *Dunn* es valuado entre el intervalo de $[0, \infty)$ y se considerará el óptimo aquel valor máximo entre los demás valores, sugiriendo que obtiene el valor óptimo el método jerárquico formando 2 *clusters* (ver *Figura 4.3*) y de igual forma empleando cualquiera de

los dos conjuntos de datos. El índice *Silhouette* es valuado entre el intervalo de [-1,1] y se considerará óptimo más cercano a 1 o igual 1, indicando que el método *K-Means* para los datos promediados en escala logarítmica al particionar en 10 *clusters* y el método jerárquico para los datos promediados por bandas de frecuencia de 50 en 50 Hz al aglomerar en 2 *clusters*, obtienen grupos homogéneos (ver *Figura 4.4*).

Pero la *Figura 4.1* también expone que los resultados obtenidos al implementar las medidas de validación interna y al emplear los datos promediados por bandas de frecuencia de 50 en 50 Hz. sugiere que se obtienen grupos más homogéneos.

clustering methods											
Hierarchical Kmeans pam											
Cluster sizes:											
2 3 4 5 6 7 8 9 10											
Validation Measures:											
datos promediados			2	3	4	5	6	7	8	9	10
escala logarítmica	hierarchica	Connectivit	6.2325	21.0254	50.4242	50.8714	56.1429	57.1246	59.8552	71.3135	75.5254
		kmeans	41.3389	67.8361	62.8397	61.5452	74.0901	87.4857	90.5111	98.569	102.7591
		pam	48.6365	74.3849	87.7571	75.8599	80.669	93.1758	107.6587	103.5933	109.4873
50 en 50 Hz	hierarchica		2.929	7.5492	12.6774	29.4083	41.154	61.7353	61.8603	61.8603	62.5187
		kmeans	36.6548	72.6885	75.1917	74.1877	72.806	85.9222	102.8294	90.7198	85.4103
		pam	32.594	50.3988	78.3687	108.0218	115.3325	122.3377	133.5964	143.0679	137.9214
escala logarítmica	hierarchica	Dunn	0.3685	0.1851	0.1871	0.1915	0.205	0.205	0.205	0.2236	0.2236
		kmeans	0.203	0.1138	0.1812	0.2006	0.215	0.173	0.173	0.173	0.173
		pam	0.1279	0.1088	0.1932	0.2065	0.2065	0.0694	0.0694	0.0956	0.0956
50 en 50 Hz	hierarchica		0.4375	0.4077	0.3161	0.1833	0.1833	0.1833	0.1833	0.1833	0.1833
		kmeans	0.1302	0.0922	0.094	0.1036	0.1069	0.1173	0.1222	0.1782	0.1195
		pam	0.1242	0.1452	0.1201	0.094	0.0666	0.094	0.094	0.0979	0.0979
escala logarítmica	hierarchica	Silhouette	0.1976	0.185	0.2258	0.2424	0.2423	0.2407	0.2377	0.2695	0.2611
		kmeans	0.2586	0.2585	0.2607	0.267	0.2708	0.2686	0.2719	0.2698	0.2845
		pam	0.2511	0.2251	0.2417	0.2625	0.2665	0.272	0.2582	0.2763	0.2729
		dtw	0.03	0.05	0.07	0.08	0.07	0.08	0.05	0.06	0.07
50 en 50 Hz	hierarchica		0.4179	0.30093	0.2627	0.2737	0.2619	0.2774	0.2741	0.2644	0.2455
		kmeans	0.3018	0.2678	0.292	0.2912	0.3012	0.2998	0.2707	0.2958	0.2933
		pam	0.2983	0.283	0.2145	0.2296	0.2055	0.2251	0.2311	0.2271	0.241
		dtw	0.2	0.19	0.13	0.06	0.08	0.09	0.09	0.1	0.1

Figura 4.1: aplicando validación interna sobre los algoritmos de “clustering”

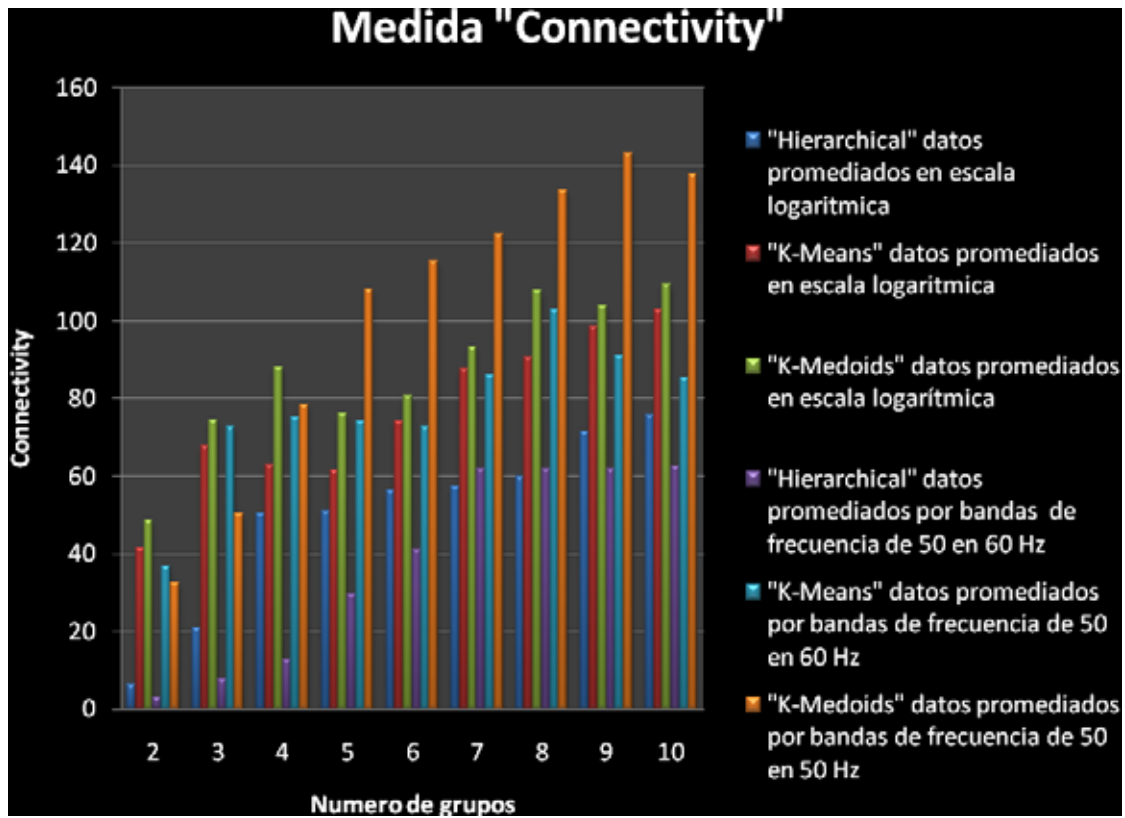


Figura 4.2: Medida Interna "Connectivity"

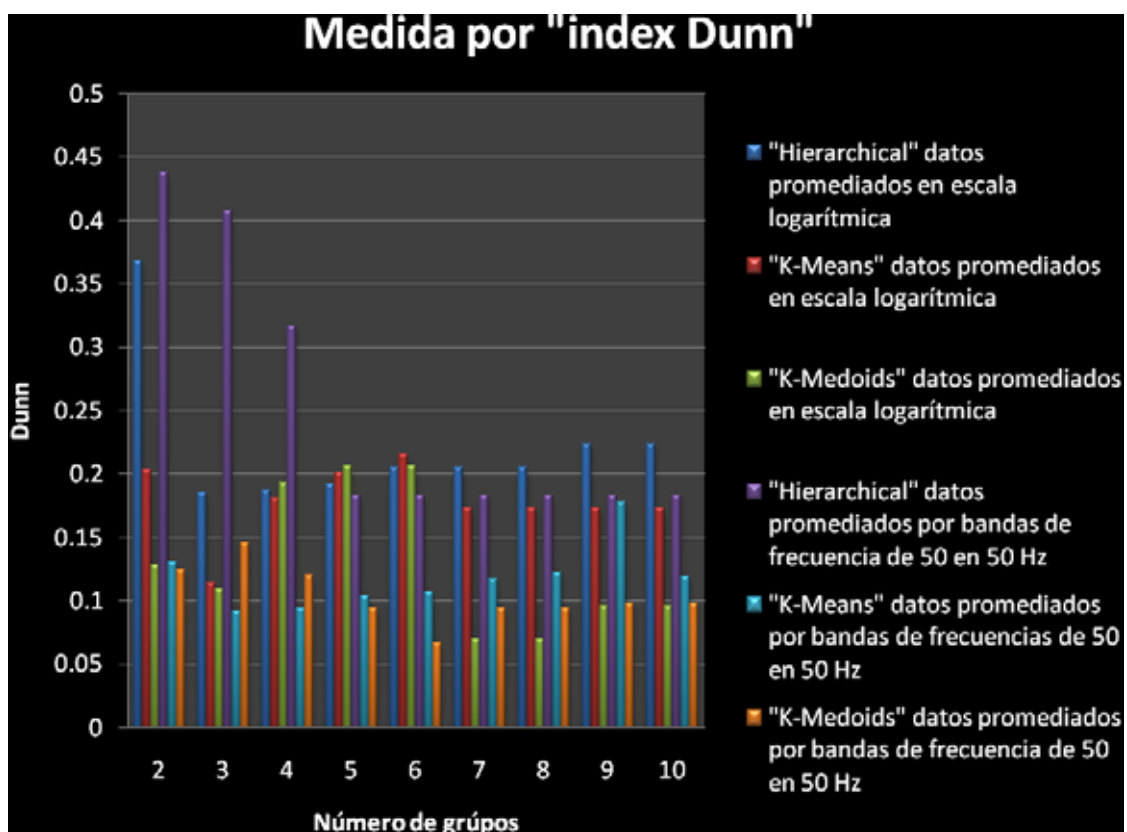


Figura 4.3: Medida Interna "Dunn"

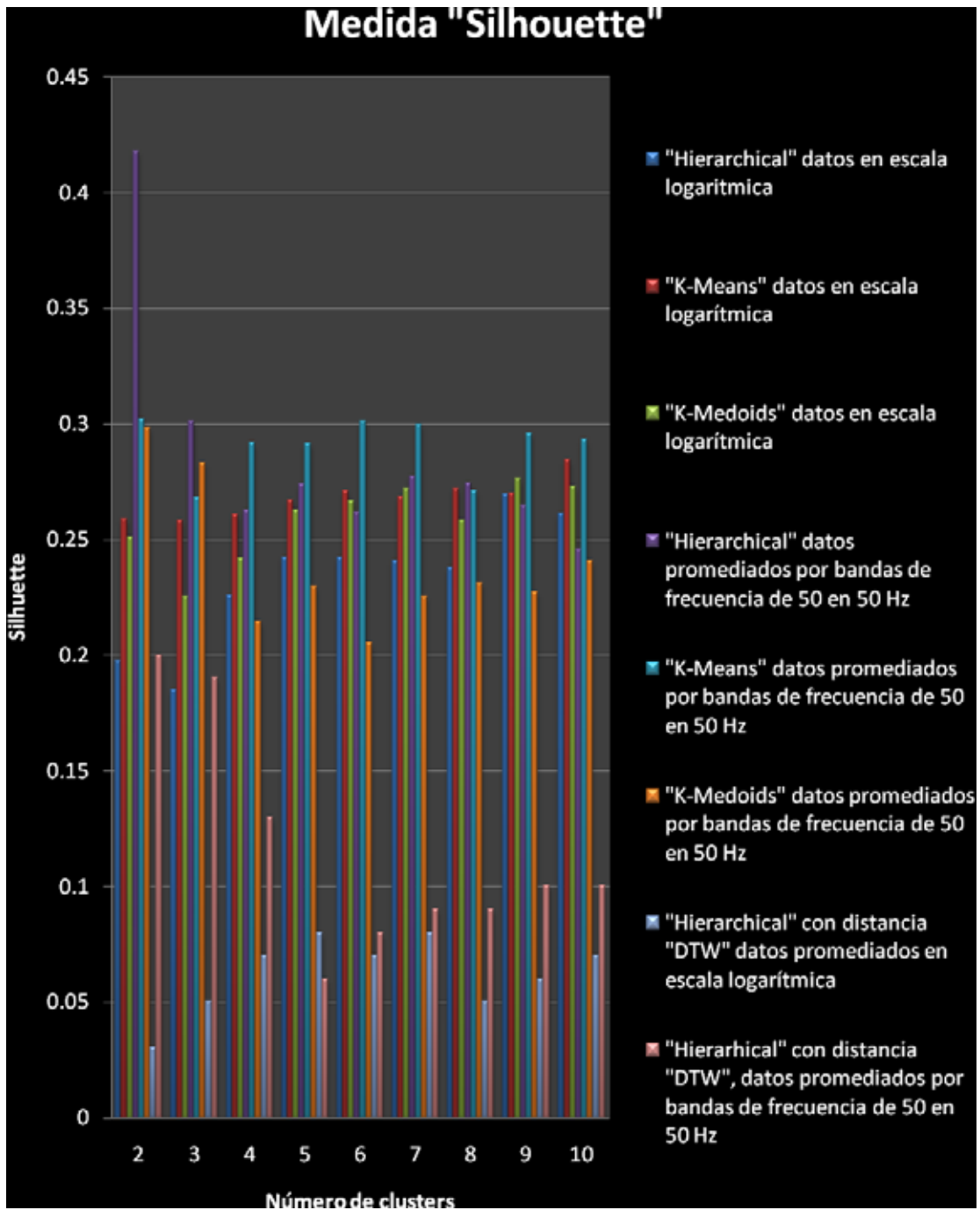


Figura 4.4: Medida Interna "Silhouette"

4.2 Resultados obtenidos al utilizar los algoritmos de agrupamiento (clustering) de R

En esta sección se exponen los resultados obtenidos por los algoritmos de agrupamiento: *K-Means*, *K-Medoids*, jerárquico y jerárquico con distancia *DTW*.

4.2.1 Resultados obtenidos al aplicar *k-Means* del paquete R

Al implementar el método *K-Means* solo se llegaron a distinguir algunos sonidos de disparos reales al emplear los datos promediados por bandas de frecuencia en escala logarítmica. Caso contrario al emplear los datos promediados por bandas de frecuencia de 50 en 50 Hz, y por este motivo solo se expone el mejor resultado obtenido por el método *K-Means* usando los datos promediados por bandas de frecuencia en escala logarítmica.

Al implementar el método *K-Means* e ir particionando en *k* grupos tomando *k* los valores desde 2 hasta 10, se pudo notar que al particionar en 10 grupos, se llegó a formar un grupo y distinguir 28 observaciones pertenecientes a sonidos de disparos reales y otros dos grupos en los que su mayoría se componen por sonidos de disparos reales. Estos resultados son expuestos en la tabla *Tabla 4.1*.

Tabla 4.1: Iteración 1 con kmeans con k=10

1 Iteración k.means con 10 clusters de tamaño: 27, 30, 37, 24, 11, 38, 34, 16, 35, 28										
Muestras	Grupos									
	1	2	3	4	5	6	7	8	9	10
DisparoReal	26	7	20	2	0	20	8	15	11	28
DisparoSalva	1	16	1	0	1	5	8	1	11	0
EfectoSonido	0	0	0	15	9	12	10	0	3	0
Explosivo	0	7	16	7	1	1	8	0	10	0
Tiempo de ejecución 0.73 segundos										

La *Figura 4.5* representa gráficamente lo obtenido por la *Tabla 4.1*. en donde se puede constatar que solo existe un grupo que se llega a separar por completo de los demás *clusters*, y los demás *clusters* presentan poca separación entre los otros grupos y existen grupos traslapados.

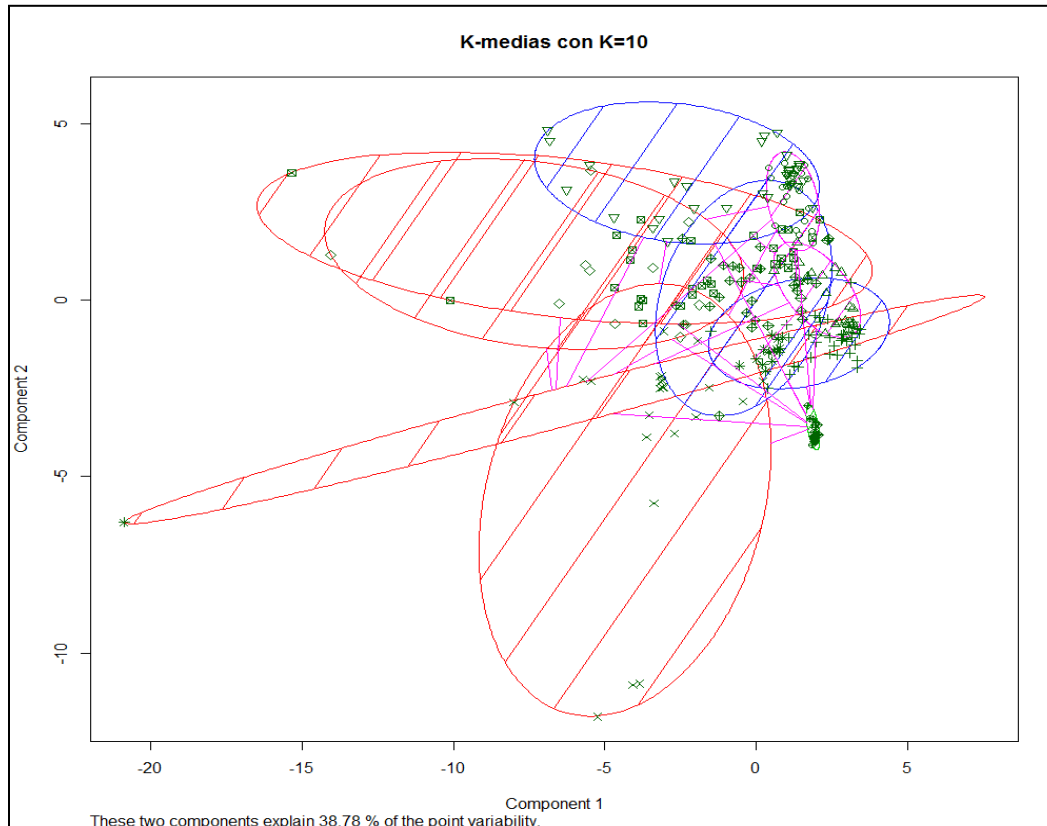


Figura 4.5: K-Means con k=10, iteración 1

Para ver los demás resultados obtenidos por el método ***K-Means***, empleando los datos promediados por bandas de frecuencia en base a una escala logarítmica ver el *Apéndice C sección C.1* y para observar los demás resultados obtenidos al emplear los datos promediados por bandas de frecuencia de 50 en 50 Hz ver *sección C.2*

4.2.2 Resultados obtenidos al usar el algoritmo K-Medoids del paquete R

Al implementar el método ***K-Medoids*** empleando cualquiera de los dos conjuntos de datos, los resultados obtenidos no presentaron ninguna distinción entre los sonidos: de disparos reales, disparos de salva, explosivos y efectos de sonido.

En las siguientes tablas se exponen los resultados obtenidos al particionar en 10 ***clusters*** empleando los dos conjuntos de datos

La *Tabla 4.2* exhibe los resultados obtenidos al emplear el conjunto de datos promediados por bandas de frecuencia en base a una escala logarítmica, y en esta *Tabla 4.2* se pueden resaltar tres pequeños grupos que en su mayoría lo conforma sonidos disparos reales sin embargo sin llegar a separar por completo los sonidos de disparo real de alguna otra observación.

Tabla 4.2: implementado K-Medoids y particionando en 10 clusters

k-medoids con 10 clusters de tamaño: 49, 29,26,28, 22,30,31,26,16,22										
Muestras	Grupos									
	1	2	3	4	5	6	7	8	9	10
DisparoReal	27	17	25	3	13	2	28	7	15	0
DisparoSalva	8	0	1	9	4	0	0	17	1	4
EfectoSonido	12	0	0	2	2	16	0	0	0	17
Explosivo	2	12	0	14	3	9	2	7	0	1
Tiempo de ejecución 0.42 segundos										

La *Tabla 4.3* expone los resultados obtenidos al particionar en 10 *clusters*, empleando el conjunto de datos promediados por bandas de frecuencia de 50 en 50 Hz. En la cual se puede notar que los resultados no obtienen una clara distinción entre los sonidos, al igual que los resultados exhibidos por la *Tabla 4.2* al emplear el conjunto de datos promediados por bandas de frecuencia en escala logarítmica

Tabla 4.3: implementado K-Medoids y particionando en 10 clusters

k-medoids con 10 clusters de tamaño: 21,33,31,10,36,28,41,50,34,17										
Muestras	Grupos									
	1	2	3	4	5	6	7	8	9	10
DisparoReal	20	26	18	6	2	9	25	17	29	0
DisparoSalva	1	7	2	0	13	18	7	1	0	1
EfectoSonido	0	0	0	0	8	0	2	24	14	14
Explosivo	0	0	11	4	13	1	7	8	4	2
Tiempo de ejecución 0.56 segundos										

Para ver los demás resultados obtenidos por el método *K-Medoids* empleando los datos promediados por bandas de frecuencia en escala logarítmica dirigirse al *Apéndice D sección D.1*.

Para ver los demás resultados obtenidos por el método *K-Medoids* empleando los datos promediados por bandas de frecuencia de 50 en 50 Hz dirigirse al *Apéndice D sección D.2*

4.2.3 Resultados obtenidos al utilizar el algoritmo Hierarchical del paquete R

Al implementarse el método jerárquico pudo verse que se genera un mejor resultado al particionar en 10 *clusters*, porque se pudo notar que al aumentar el número de grupos se incrementa el número de observaciones de sonidos de disparo real que pueden reconocerse y conformar pequeños grupos con estas observaciones, pero también puede verse que al emplear los datos promediados en escala logarítmica se obtiene un mejor desempeño al distinguir sonidos de disparos reales, comparando con los datos promediados de 50 en 50 Hz.

La *Tabla 4.4* expone los resultados obtenidos al emplear los datos promediados en escala logarítmica, y pueden verse 4 grupos conformados por muestras que corresponden a ser sonidos de disparos reales.

Tabla 4.4: método jerárquico, particionando en 10 clusters

Hierarchical con 10 clusters de tamaño: 20,36,25,58,23,41,30,14,27,6										
Muestras	Grupos									
	1	2	3	4	5	6	7	8	9	10
DisparoReal	20	18	25	23	2	1	7	14	27	0
DisparoSalva	0	0	0	18	0	9	17	0	0	0
EfectoSonido	0	0	0	2	14	28	0	0	0	5
Explosivo	0	18	0	15	7	3	6	0	0	1
Tiempo de ejecución 0.09 segundos										

La *Figura 4.6* podemos ver el dendograma y marcado por un rectángulo azul aquellos grupos conformados por sonidos de disparos reales, sin embargo al ver el resultado que muestra la *Figura 4.7* la cual expone el resultado del obtenido por el índice *Silhouette* la cual indica la estructura de los *clusters* generados, señala que el grupo 1 y 3 no son compactos o no hay una alta cohesión entre las observaciones que componen estos *clusters* y el grupo 8 y 9 si son compactos.

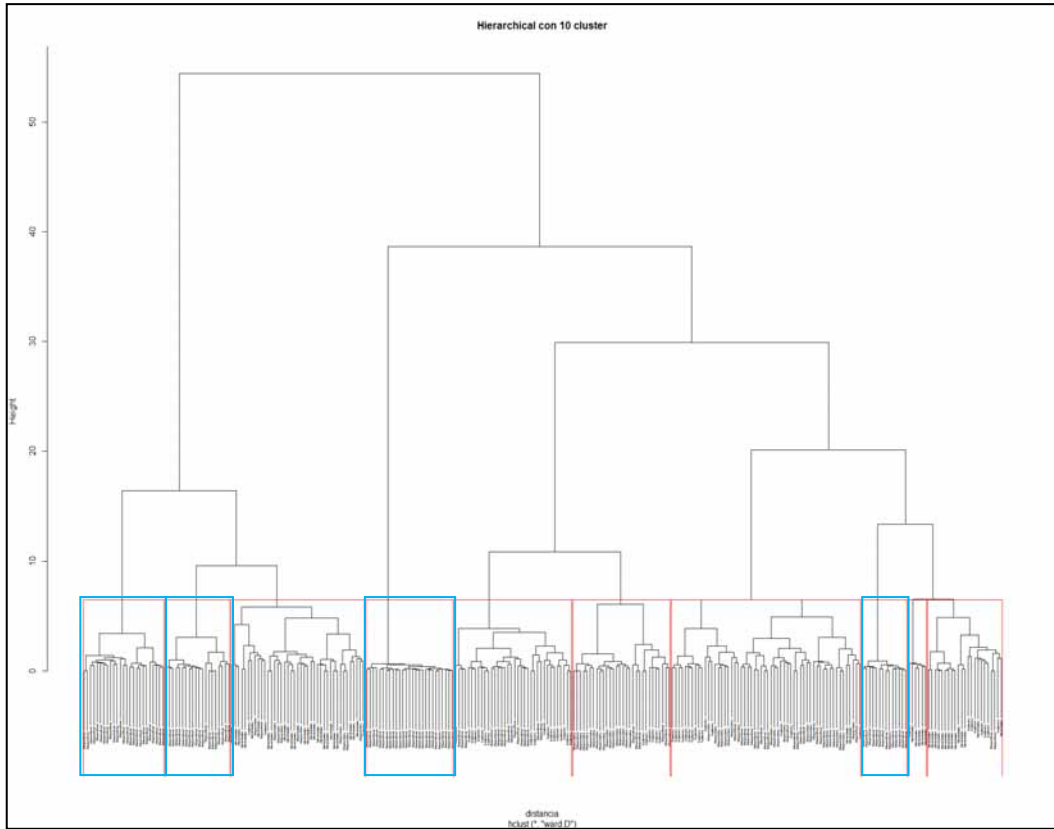


Figura 4.6: dendrograma del método jerárquico al particionar con 10 clusters

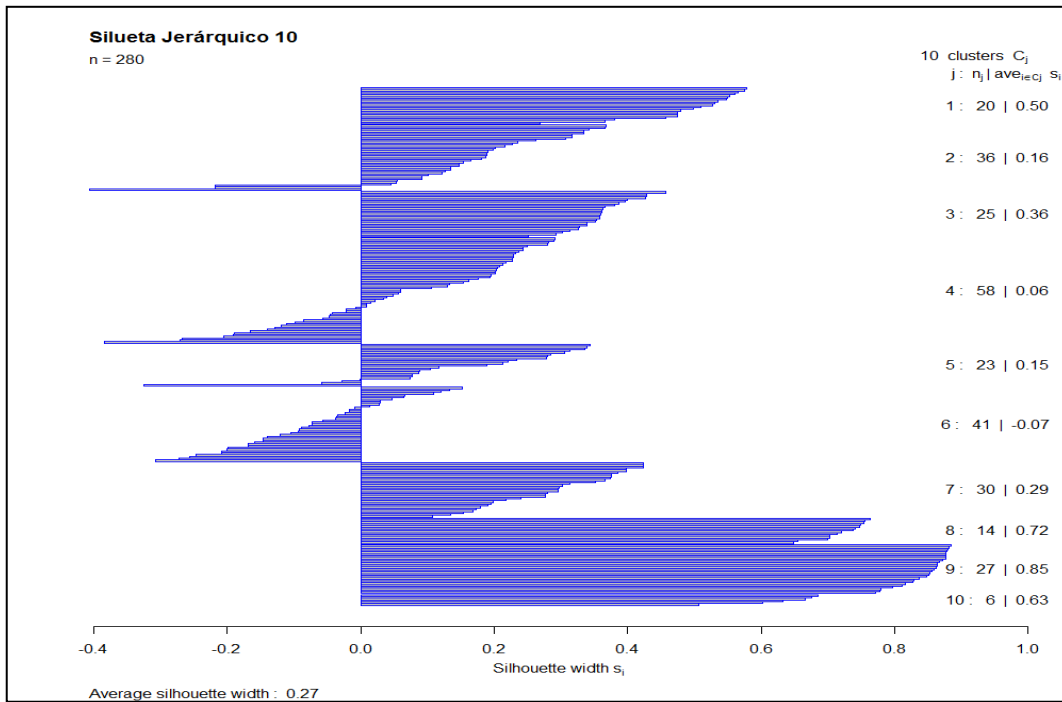


Figura 4.7: índice *silhouette* del método jerárquico al particionar con 10 clusters

La *Tabla 4.5* expone los resultados obtenidos al emplear los datos promediados de 50 en 50 Hz y puede notarse que genera un grupo menos que el resultado expuesto por la *Tabla 4.4* empleando los datos promediados en escala logarítmica.

Tabla 4.5: método jerárquico, particionando en 10 clusters

Hierarchical con 10 clusters de tamaño: 26, 20, 20, 24, 27, 55, 61, 9, 27, 32										
Muestras	Grupos									
	1	2	3	4	5	6	7	8	9	10
DisparoReal	26	11	20	6	7	30	20	5	27	0
DisparoSalva	0	0	0	11	19	14	1	0	0	5
EfectoSonido	0	0	0	6	0	3	25	0	0	15
Explosivo	0	9	0	1	1	8	15	4	0	12
Tiempo de ejecución 0.24 segundos										

La *Figura 4.8* muestra el dendograma con los *clusters* formados y marcados con un rectángulo azul aquellos *clusters* conformados por sonidos de disparo real.

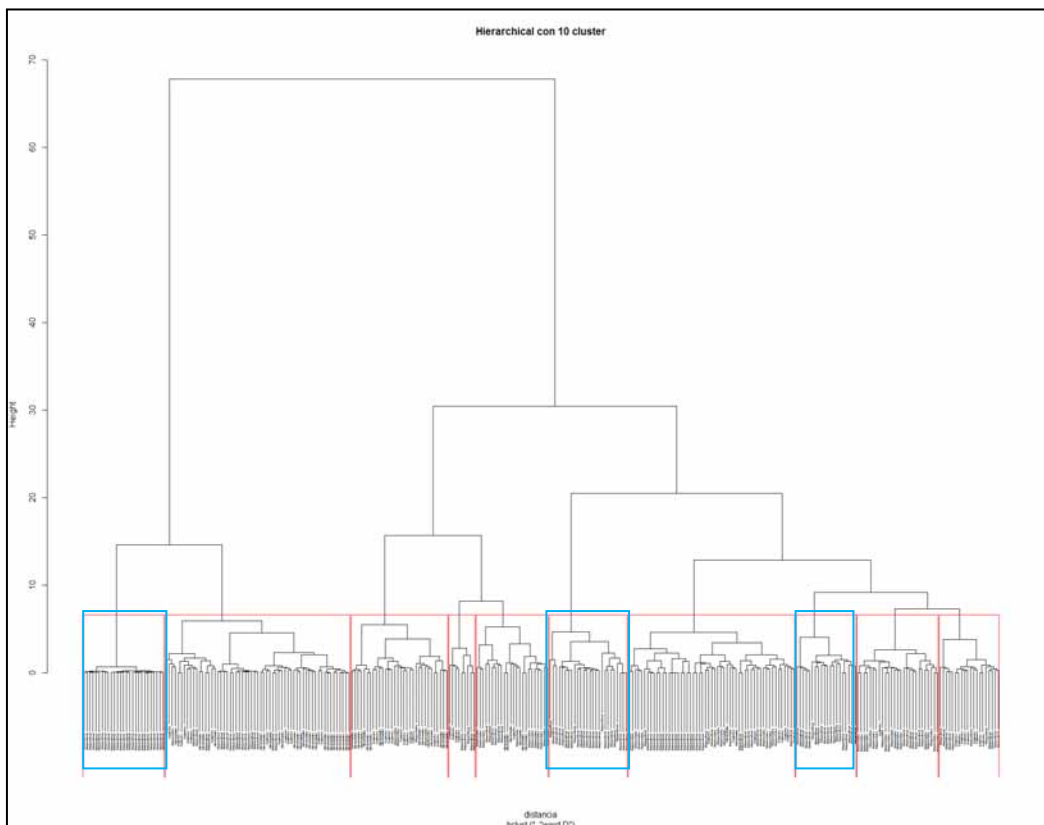


Figura 4.8: dendograma del método jerárquico al particionar con 10 clusters

La *Figura 4.9* expone que tan compactos son los *clusters* formados, el *cluster* 1 y 3 muestran una estructura débil y por otro lado el *cluster* 9 presenta cohesión entre sus observaciones que lo conforman.

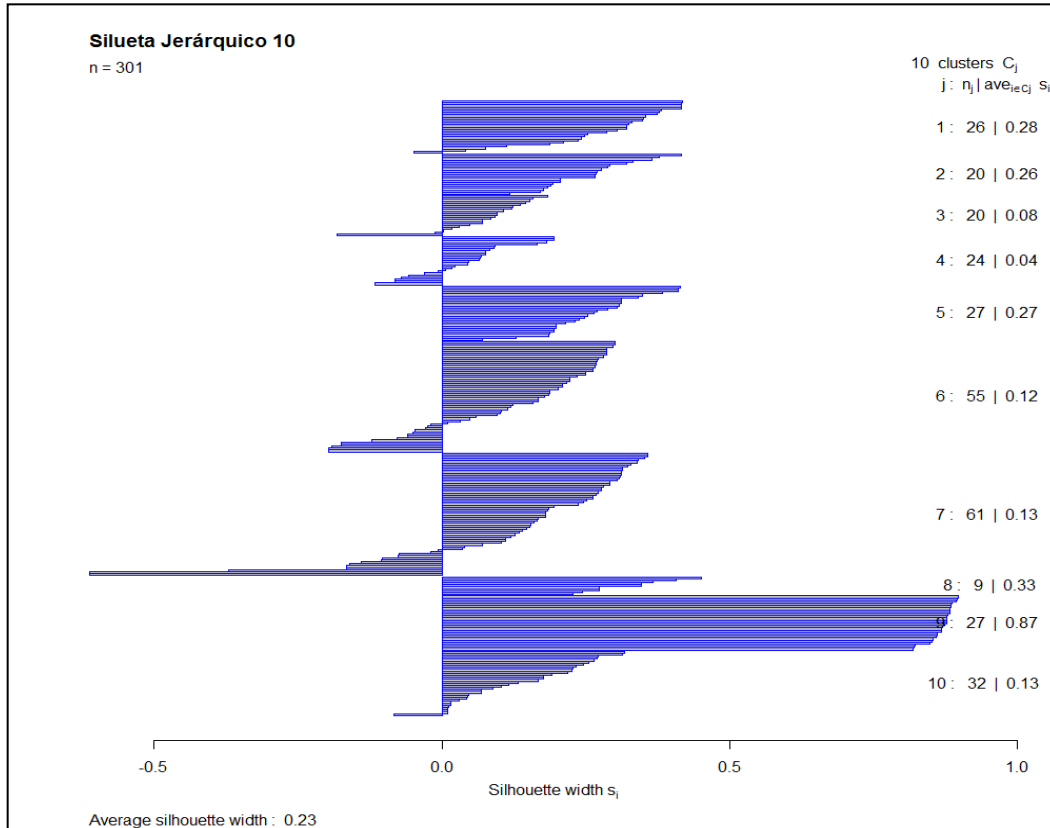


Figura 4.9: índice *silhouette* del método jerárquico al particionar con 10 clusters

Para ver todos los demás resultados obtenidos por el método jerárquico al emplear los datos promediados en escala logarítmica ver el *Apéndice E sección E.1* o para ver los resultados obtenidos al emplear los datos promediados de 50 en 50 Hz ver la *sección E.2*.

4.2.4 Resultados obtenidos al utilizar el algoritmo Hierarchical con distancia DTW del paquete R

La *Tabla 4.6* exhibe los resultados al emplear los datos promediados en escala logarítmica y la *Tabla 4.7* muestra los resultados al utilizar los datos promediados de 50 en 50 Hz. Al comparar estos resultados puede identificarse que al emplear los datos promediados de 50 en 50 Hz, se obtiene una mayor distinción de los sonidos de disparo real, al generar 3 grupos conformados por sonidos de disparos reales.

Tabla 4.6: método Jerárquico particionando 10 clusters usando distancia de similitud DTW

Hierarchical con distancia "DTW", aglomerando 10 clusters de tamaño:46,41,37,25,29,31,30,23, 11, 7										
Muestras	Grupos									
	1	2	3	4	5	6	7	8	9	10
DisparoReal	7	15	27	24	13	22	28	1	0	0
DisparoSalva	22	6	2	0	4	1	0	6	3	0
EfectoSonido	13	6	0	0	6	7	0	2	8	7
Explosivo	4	14	8	1	6	1	2	14	0	0
Tiempo de ejecución 125.13 segundos										

Tabla 4.7: método Jerárquico particionando 10 clusters usando distancia de similitud DTW

Hierarchical con distancia "DTW", aglomerando 10 clusters de tamaño: 5,58,47,19,6,22,67,26,20, 31										
Muestras	Grupos									
	1	2	3	4	5	6	7	8	9	10
DisparoReal	4	23	47	19	6	21	24	6	2	0
DisparoSalva	0	5	0	0	0	0	29	4	2	10
EfectoSonido	0	25	0	0	0	0	0	3	12	9
Explosivo	1	5	0	0	0	1	14	13	4	12
Tiempo de ejecución 482.62segundos										

La *Figura 4.10* expone los resultados expuestos por la *Tabla 4.7* y en donde el método jerárquico con distancia *DTW* uso los datos promediados por bandas de frecuencia de 50 en 50 Hz, y se marcan con un rectángulo azul aquellos grupos en los que se reconocen los sonidos de disparo real.

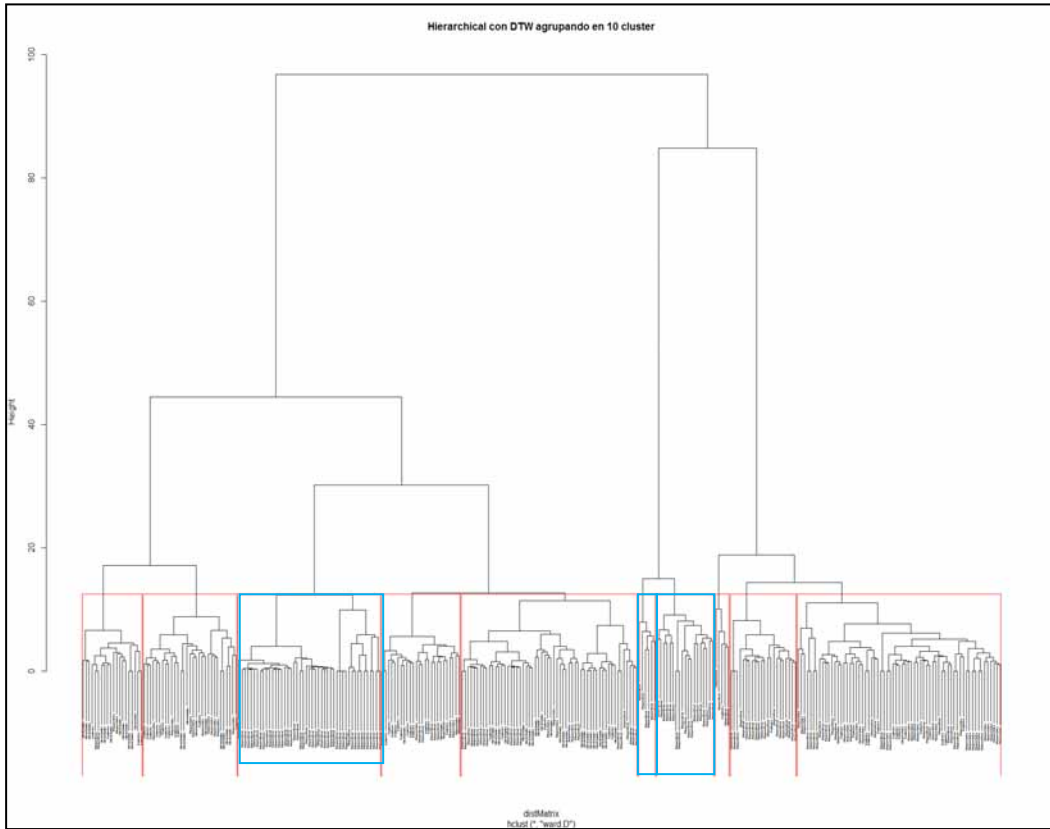


Figura 4.10: método jerárquico con distancia DTW particionando en 10 clusters

Pero la *Figura 4.11* muestra el resultado del índice *Silhouette* sobre esos grupos generados y sugiere que los grupos no son tan compactos y que tienen una estructura débil o nula estructura.

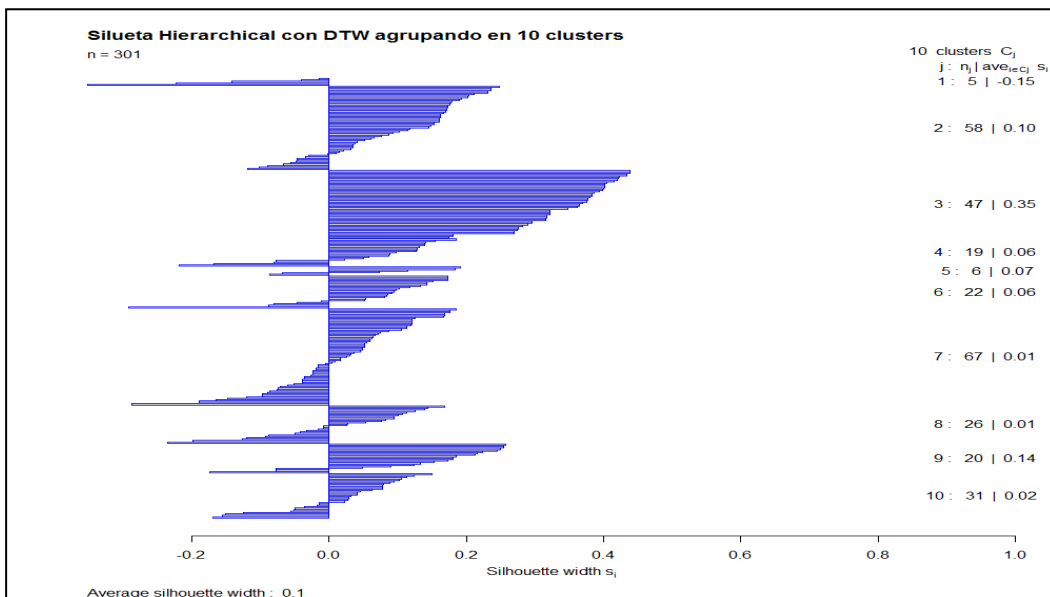


Figura 4.11: índice *silhouette* del método jerárquico usando distancia DTW al particionar con 10 clusters

Para ver los demás resultados obtenidos por el método jerárquico con distancia *DTW* ver *Apéndice F. Sección F.1* para ver los resultados al emplear los datos promediados en escala logarítmica.

Sección F.2 para ver los resultados obtenidos al emplear los datos promediados de 50 en 50 Hz

4.3 Resultados obtenidos al utilizar el algoritmo máquinas de soporte vectorial (Support Vector Machines, por sus siglas en inglés SVM) del paquete R

Para la aplicación de las máquinas de soporte vectorial se emplearon cuatro funciones de *kernels* (núcleos).

Kernels utilizados: lineal, polinomial, radial Gaussiana y sigmoideal.

Los resultados obtenidos al implementar el *SVM*, usando un *kernel* lineal y empleando cualquiera de los dos conjuntos de datos se obtuvo una buena solución al clasificar correctamente arriba del 95% los distintos sonidos (disparos de reales, disparos de salva, explosivos y efectos de sonido).

La *Tabla 4.8* muestra un resumen de la clasificación obtenida a través del *SVM* con *kernel* lineal empleando los datos promediados en escala logarítmica y la *Tabla 4.9* expone los resultados obtenidos al emplear los datos promediados de 50 en 50 Hz, así también los vectores de soporte, que son los puntos que tocan el límite del margen y por otro lado también exhiben la precisión, que indica la probabilidad que tiene una observación de dar positivo a ser un disparo real, salva, efecto de sonido o explosivo en la prueba, la sensibilidad indica la probabilidad de clasificar correctamente un sonido, F y la exactitud la cual indica la probabilidad de resultados correctos en la prueba.

Como se mencionó anteriormente la *Tabla 4.8* exhibe los resultados obtenidos a través del *SVM* usando un *kernel* lineal y empleando los datos promediados en escala logarítmica y se puede ver que de 137 disparos reales el sistema predijo que dos fueron disparos de salva y dos explosivos, de 44 disparos de salva el sistema predijo que uno era de disparo real, los 49 sonidos producidos por efecto de sonido se lograron clasificar y de 50 sonidos producidos por sonido de un explosivo uno se predijo como disparo de salva y cuatro como disparo real. Y una exactitud de clasificación del sistema aproximadamente del 96%.

Tabla 4.8: SVM con kernel lineal empleando datos promediados en escala logarítmica

Clasificación con el algoritmo SVM utilizando kernel lineal				
Muestras	DisparoReal	DisparoSalva	EfectoSonido	Explosivo
DisparoReal	133	2	0	2
DisparoSalva	1	43	0	0
EfectoSonido	0	0	49	0
Explosivo	4	1	0	45
Number of Support Vectors: 120 (40 26 33 21) Number of Classes: 4 Levels: DisparoReal DisparoSalva EfectoSonido Explosivo				
Tiempo de ejecución 0.42 segundos				
Medidas	Disparos reales	Disparos de salva	Efectos de sonido	explosivo
precisión	0.9708029	0.9772727	1	0.9
sensibilidad	0.9637681	0.9347826	1	0.9574468
F	0.9672727	0.9555556	1	0.9278351
Exactitud	0.9642857			

Por otro lado la *Tabla 4.9* exhibe los resultados obtenidos a través del *SVM* usando un *kernel* lineal y empleando los datos promediados de 50 en 50 Hz y se puede ver que de 152 disparos reales el sistema lo clasificó correctamente con una precisión del 100%, de 50 disparos de salva el sistema también clasificó correctamente el 100%, los 49 sonidos producidos por efecto de sonido el sistema también los reconoció completamente y de 50 sonidos producidos por explosivo uno se predijo como disparo de real y otro como disparo salva con una precisión del 96%. Y el sistema tiene una probabilidad de clasificar correctamente con estos datos aproximadamente el 99%.

Tabla 4.9: SVM con kernel lineal empleando los datos promediados de 50 en 50 Hz

Clasificación con el algoritmo SVM utilizando kernel lineal				
	DisparoReal	DisparoSalva	EfectoSonido	Explosivo
DisparoReal	152	0	0	0
DisparoSalva	0	50	0	0
EfectoSonido	0	0	49	0
Explosivo	1	1	0	48
Number of Support Vectors: 151 (51 30 38 32) Number of Classes: 4 Levels: DisparoReal DisparoSalva EfectoSonido Explosivo				
Tiempo de ejecución 0.75 segundos				
Medidas	Disparos reales	Disparos salva	efectos	explosivo
precisión	1	1	1	0.96
sensibilidad	0.9934641	0.9803922	1	1
F	0.9967213	0.990099	1	0.9795918
Exactitud	0.9933555			

4.4 Recapitulación

En este capítulo se presentaron los resultados obtenidos al usar los algoritmos de R *K-Means*, *K-Medoids*, jerárquico, jerárquico con distancia *DTW*, el *SVM*, y así también los resultados obtenidos por el paquete *cValid* [31], los cuáles exhiben que tan homogéneos son los *clusters*, pero solo se exponen aquellos resultados que en apreciación ofrecieron la mejor solución, en base, a que mostraron un mayor reconocimiento de los sonidos de disparo real.

Los resultados obtenidos por las medidas de validación interna al usar el paquete *cValid* de R, sugieren que el algoritmo jerárquico, al aglomerar en dos *clusters* forma grupos homogéneos, compactos, con alta cohesión intra-cluster en comparación con los otros algoritmos, pero también sugiere que los resultados obtenidos al implementar las medidas de validación interna y al emplear los datos promediados por bandas de frecuencia de 50 en 50 Hz se obtienen grupos más homogéneos.

Al usar el algoritmo *K-Means* del paquete R obtuvo mejores resultados al particionar en 10 *clusters* y empleando los datos promediados por bandas de frecuencia de 50 en 50 Hz reconociendo y generando un grupo con 28 sonidos de disparos reales.

Al usar el algoritmo *K-Medoids* empleando cualquiera de los dos conjuntos de datos, los resultados obtenidos no presentaron ninguna distinción entre los sonidos: de disparos reales, disparos de salva, explosivos y efectos de sonido.

Al usar el algoritmo jerárquico y empleando cualquiera de los dos conjuntos de datos, se logró reconocer y formar grupos compuestos por sonidos de disparos reales, pero se obtuvo un mejor resultado al particionar en 10 *clusters* y al emplear el conjunto de datos promediado por bandas de frecuencia en escala logarítmica, ya que se formaron cuatro grupos de tamaño: 28, 25, 14 y 27 conformados en su totalidad por sonidos de disparos reales.

Al usar el algoritmo jerárquico y empleando el paquete dtw de R se obtuvieron mejores resultados al usar los datos promediados por bandas de frecuencia de 50 en 50 Hz y particionando en 10 *clusters* ya que se formaron tres grupos de tamaño: 47, 19 y 6 conformados en su totalidad por sonidos de disparos reales.

Al usar el algoritmo *SVM* de R mostró que al usar cualquiera de los conjuntos de datos y al emplear un *kernel* lineal se obtiene una mayor exactitud de reconocimiento del sonido de disparo real, aproximadamente entre el 96 y 99 por ciento.

Capítulo 5

Análisis y discusión de Resultados

5.1 Comparación de resultados

El método jerárquico aplicado sobre cualquiera de los dos conjuntos de datos derivados a partir de los parámetros obtenidos a través de la *FFT* y promediados ya sea, por bandas de frecuencia o en base a una escala logarítmica formaron *clusters* en los que se lograban reconocer entre 14 y 27 disparos reales de entre un total de 280 observaciones para el caso de los datos promediados por bandas de frecuencia en escala logarítmica y 301 observaciones totales para el caso de los datos promediados por bandas de frecuencia de 50 en 50 Hz..

El método *K-Medoids* aplicado sobre cualquiera de los dos conjuntos de datos promediados ya sea, por bandas de frecuencia o en base a una escala logarítmica, no logro reconocer o distinguir el sonido de disparo real de entre los demás sonidos producidos por: disparo de salva, explosivo y efecto de sonido.

El método *K-Means* aplicado sobre los datos promediados por bandas de frecuencia en base a escala logarítmica presentó mejor solución formando pequeños *cluster* que lograron reconocer entre 27 y 29 sonidos de disparo real. Caso contrario empleando los datos promediados por bandas de frecuencia de 50 en 50 Hz los cuáles no lograron formar ningún *cluster* en el cual se reconocieran los disparos reales.

El método jerárquico empleando distancia *DTW* mostró mejor resultado, aplicándolo sobre los datos promediados por bandas de frecuencia de 50 en 50 Hz, formando pequeños *clusters* en los cuáles lograban reconocer entre 6 y 47 disparos reales. A diferencia de los resultados, utilizando el conjunto de datos promediados por bandas de frecuencia en base a escala logarítmica en el que se presentaron *clusters* que no distinguieron los sonidos de disparo real de los demás sonidos.

El método *SVM* empleando diferentes *kernels* obtuvo mejor resultado aplicándolo sobre los datos promediados por bandas de frecuencia en escala logarítmica al obtener una probabilidad de entre 71 y 96 por ciento de clasificación correcta para los distintos sonidos. Al aplicar el método *SVM* sobre los datos promediados por bandas de frecuencia se obtuvo entre 59 y 99 por ciento de clasificación correcta de las muestras de sonidos proporcionados en este proyecto. Sin embargo el “*SVM*” con *kernel* lineal y radial Gaussiana obtuvieron el

mejor resultado independientemente del conjunto de datos usado, clasificando correctamente las muestras arriba del 95%.

Resultados obtenidos al implementar métodos de *clustering*:

K-Means aplicado a 280 muestras promediadas por bandas de frecuencia en escala logarítmica

- Al aumentar el número de grupos se llegaban a separar entre 27 y 29 muestras aproximadamente el 10% de los sonidos disparo producidos por arma de fuego
- Poca distancia de separamiento entre *clusters*
- Tiempo de ejecución 1.62 segundos
- Índice “*Silhouette*” presenta débil estructura para el algoritmo ***K-Means*** lo que implica utilizar otro método para este conjunto de datos

K-Means aplicado a 301 muestras promediadas por bandas de frecuencia de 50 en 50 Hz

- No presentó separamiento entre las muestras
- *Clusters* traslapados
- Poco separamiento entre los *clusters*
- Tiempo de ejecución 1.02 segundos
- Índice “*Silhouette*” mostró que el algoritmo presenta débil estructura para el algoritmo ***K-Means*** lo que implica intentar otro método para este conjunto de datos

K-Medoids aplicado a 280 muestras promediadas por bandas de frecuencia en escala logarítmica

- Mostró traslapamiento de *clusters*
- Dificultad para distinguir entre las muestras
- Tiempo de ejecución 0.45 segundos
- Índice “*Silhouette*” presenta débil estructura para el algoritmo ***K-Medoids*** lo que implica utilizar otro método para este conjunto de datos

K-Medoids aplicado a 301 muestras promediadas por bandas de frecuencia de 50 en 50 Hz

- Mostró traslapamiento entre los *clusters*
- Dificultada para distinguir entre las muestras
- Tiempo de ejecución 0.41 segundos
- Índice “*Silhouette*” mostró que el algoritmo presenta débil estructura para el algoritmo ***K-Medoids*** lo que implica intentar otro método para este conjunto de datos

Jerárquico aplicado a 280 muestras promediadas por bandas de frecuencia en escala logarítmica

- Al aumentar el número de grupos se llegaban a separar entre 14 y 27 muestras aproximadamente el 10% de los sonidos disparo producidos por arma de fuego
- Dificultada para distinguir entre las muestras
- Tiempo de ejecución 0.12 segundos
- Índice “*Silhouette*” presenta débil estructura para el algoritmo jerárquico lo que implica utilizar otro método para este conjunto de datos

Jerárquico aplicado a 301 muestras promediadas por bandas de frecuencia de 50 en 50 Hz

- Al aumentar el número de grupos se llegaban a separar entre 20 y 27 muestras aproximadamente el 10% de los sonidos disparo producidos por arma de fuego
- Dificultada para distinguir entre las muestras
- Tiempo de ejecución 0.15 segundos
- Índice “*Silhouette*” mostró que el algoritmo presenta débil estructura para el algoritmo jerárquico, lo que implica intentar otro método para este conjunto de datos

Jerárquico con distancia “*DTW*” aplicado a 280 muestras promediadas por bandas de frecuencia en escala logarítmica

- Mostró traslapamiento entre los *clusters*
- Dificultada para distinguir entre las muestras
- Tiempo de ejecución 4 minutos
- Índice “*Silhouette*” presenta débil estructura para el algoritmo jerárquico lo que implica utilizar otro método para este conjunto de datos

Jerárquico con distancia “*DTW*” aplicado a 301 muestras promediadas por bandas de frecuencia de 50 en 50 Hz

- Al aumentar el número de grupos se llegaban a separar entre 6 y 47 muestras aproximadamente entre el 6% y 15% de los sonidos disparo producidos por arma de fuego
- Dificultada para distinguir entre las muestras
- Tiempo de ejecución 8 minutos segundos
- Índice “*Silhouette*” mostró que el algoritmo presenta débil estructura para el algoritmo jerárquico, lo que implica intentar otro método para este conjunto de datos

Los resultados obtenidos al implementar los métodos de *clustering* no fueron los esperados, ya que no lograron identificar o separar por completo las muestras del mismo tipo y los *cluster* generados reflejaron traslapamiento entre ellos.

Los índices de validación interna (*Connectivity*, *Dunn* y *Silhouette*) aplicados sobre los algoritmos de *clustering* (K-Means, K-Medoids y jerárquico) de R y empleando cualquiera de los dos conjuntos de datos obtenidos a partir de la *FFT*, sugirieron que el algoritmo jerárquico obtuvo un mejor desempeño al construir grupos más compactos y homogéneos, al formar dos *clusters*, que los demás métodos de *clustering* (*K-Means* y *K-Medoids*) utilizados. Pero puede notarse que al obtener los resultados de todos los métodos de *clustering* no se llegaban a reconocer por completo ningún sonido, y que existía mucha similitud entre los sonidos y dificultad para poder separar las distintas categorías de los sonidos en grupos y por último también se notó que a mayor número de *clusters* formados se llegaban a separar pequeños *clusters* conformados por una pequeña cantidad de sonidos de disparos reales.

El índice *Silhouette* mostró que todos los métodos de *clustering* implementados presentan estructura débil, esto es que no presentan cohesión o relación estrecha entre las observaciones que forman los *clusters* y poca o nula separación entre diferentes *clústeres* generados, lo cual implica que se debe tratar con otros métodos (ver *Figura 5.2*).

Por otro lado las máquinas de soporte vectorial (*SVM*, del inglés *Support Vector Machines*), mostraron una clasificación correcta arriba del 50%

	Exactitud del “ <i>SVM</i> ” aplicado a 280 muestras promediadas por bandas de frecuencia en escala logarítmica	Exactitud del “ <i>SVM</i> ” aplicado a 301 muestras promediadas por bandas de frecuencia de 50 en 50 Hz
" <i>SVM</i> " con <i>kernel lineal</i>	96.429	99.336
" <i>SVM</i> " con <i>kernel</i> polinomial	71.786	59.801
" <i>SVM</i> " con <i>kernel</i> radial Gaussiana	93.214	86.379
" <i>SVM</i> " con <i>kernel</i> sigmoidal	72.500	61.130

El “*SVM*” con *kernel* lineal y radial Gaussiana obtuvieron el mejor resultado independientemente el conjunto de datos usado, clasificando correctamente las muestras arriba del 95%.

La *Figura 5.1* muestra la precisión que es la proporción del número de predicciones que son correctas, la sensibilidad que representa la proporción de malos riesgos predichos como malos, F y la exactitud que indica la probabilidad de obtener resultados correctos en la prueba.

SVM con kernel lineal				
Disparo real				
Datos promediados	Exactitud	Precisión	Sensibilidad	F
en escala logarítmica	96.42857	0.970803	0.96376812	0.967273
de 50 en 50 H	99.33555	1	0.99346405	0.996721
SVM con kernel polinomial				
Disparo real				
Datos promediados	Exactitud	Precisión	Sensibilidad	F
en escala logarítmica	71.78571	1	0.6372093	0.778409
de 50 en 50 H	59.80066	1	0.55677656	0.715294
SVM con kernel radial Gaussiana				
Disparo real				
Datos promediados	Exactitud	Precisión	Sensibilidad	F
en escala logarítmica	93.21429	0.985401	0.91836735	0.950704
de 50 en 50 H	86.37874	0.980263	0.82320442	0.894895
SVM con kernel sigmoial				
Disparo real				
Datos promediados	Exactitud	Precisión	Sensibilidad	F
en escala logarítmica	72.5	0.970803	0.71505376	0.823529
de 50 en 50 H	61.12957	0.901316	0.61434978	0.730667

Figura 5.1: Medidas de error aplicado sobre los resultados obtenidos por el SVM utilizando distintos kernels

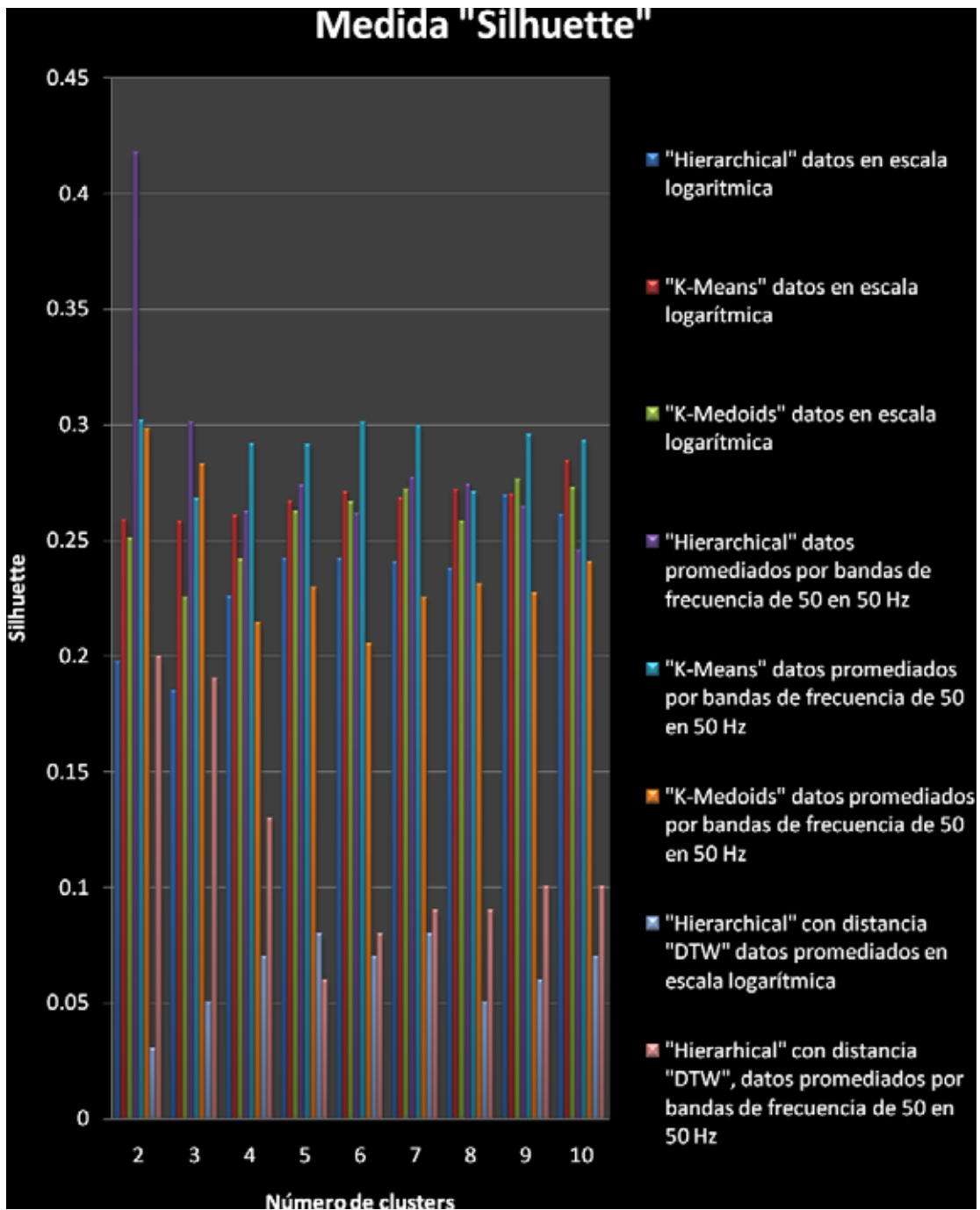


Figura 5.2: Silhouette obtenida para cada uno de los métodos empleados sobre cada uno de los conjuntos de datos

En cuanto a los tiempos de ejecución, las *Figura 5.3* y *Figura 5.4* muestran el tiempo de ejecución promedio que tardan los algoritmos empleados y se puede notar que algunos métodos tuvieron mejor rendimiento al ejecutarse sobre los datos promediados en escala logarítmica y otros métodos presentaron mejor rendimiento al ejecutarse sobre los datos promediados por bandas de frecuencia. Sin embargo, el método de agrupamiento jerárquico con distancia “*DTW*” es el menos indicado ya que tarda aproximadamente entre 2 y 8 minutos para su ejecución y el método que requirió menos tiempo para su ejecución fue el método *K-Medoids*, ejecutándose entre 0.12 y 0.15 décimas de segundo.

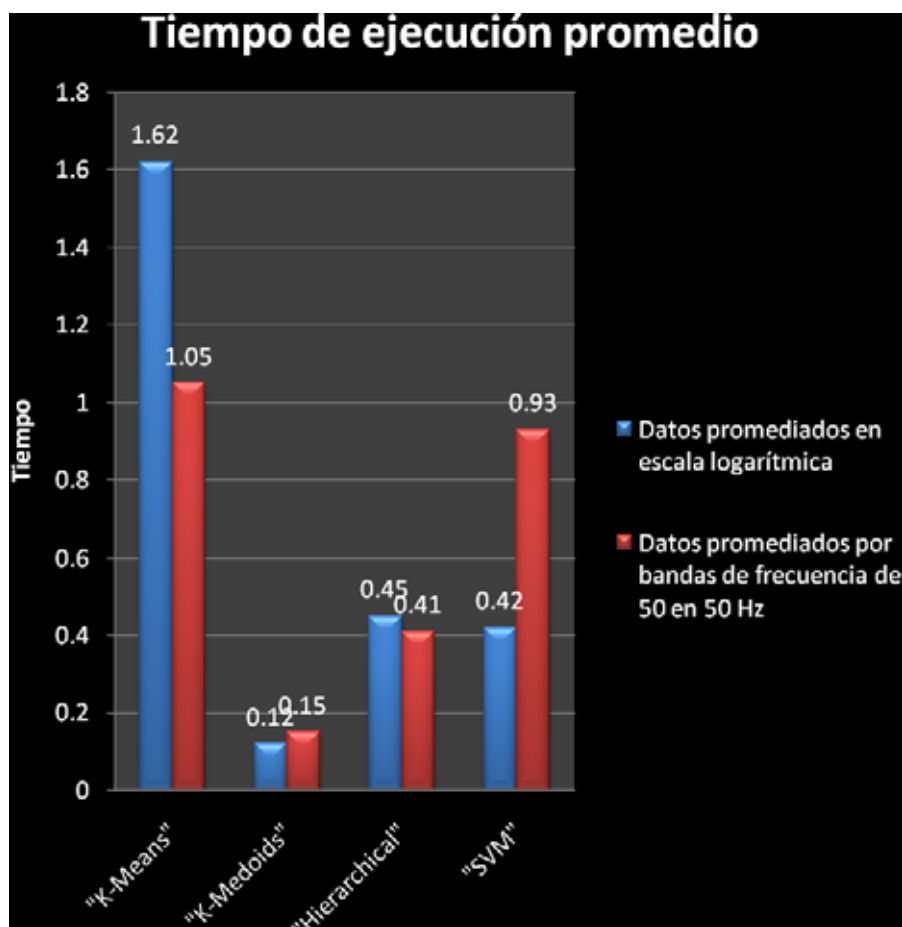


Figura 5.3: Tiempo promedio de ejecución de los algoritmos parte 1

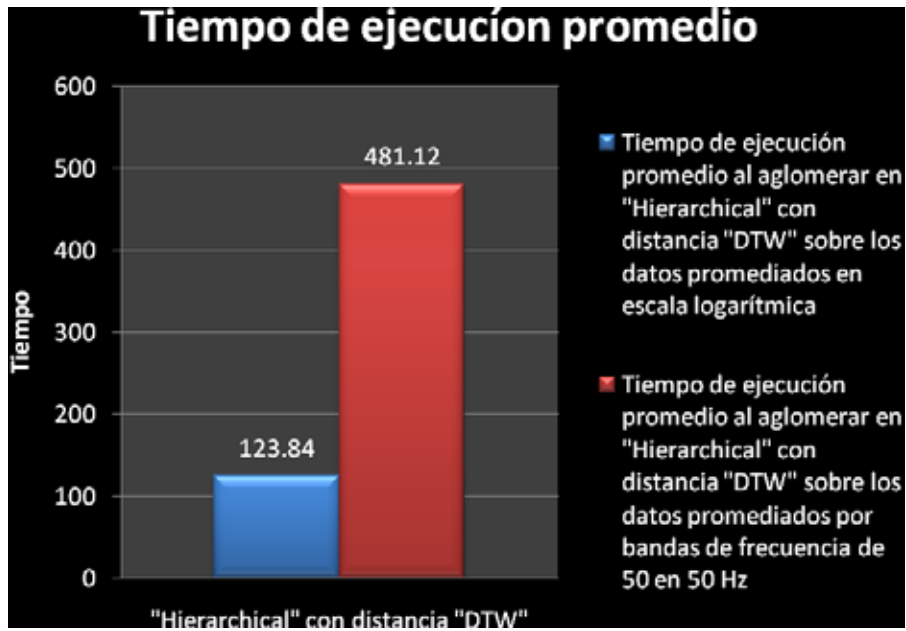


Figura 5.4: Tiempo promedio de ejecución de los algoritmos parte 2

Capítulo 6

Conclusiones y trabajos futuros

Se obtuvieron dos conjuntos de datos a partir de los parámetros obtenidos por la *FFT*, los cuáles fueron normalizados y promediados: uno por bandas de frecuencia en escala logarítmica y otro por bandas de frecuencia de 50 en 50 Hz.

Los métodos de *clustering* (*K-Means*, *K-Medoids*, jerárquico y Jerárquico con distancia *DTW*), implementados en este proyecto y que emplearon los conjuntos de datos mencionados anteriormente. No obtuvieron un buen desempeño, en cuanto a reconocer el sonido de disparo real de entre los otros sonidos producidos por: Disparos de salva, explosivos y efectos de sonido.

Las medidas de validación internas (*Connectivity*, *Dunn* y *Silhouette*), aplicadas sobre los métodos de *clustering* y las cuáles son indicativas de que tan compactos, homogéneos y distantes se encuentran los *clusters* sugieren utilizar otros métodos de minería de datos para emplear los conjuntos de datos tratados en este proyecto, debido a que los *clusters* presentaron estructura débil, esto es, que no presentan cohesión o una relación estrecha entre los objetos que conforman dichos *clusters* y poca o nula separación entre los *clusters* generados, lo cual implica que se debe tratar con otros métodos.

Por otro lado el método de clasificación supervisada *SVM* empleando cualquiera de los dos conjuntos de datos, mostró que utilizando un *kernel* lineal o radial *Gaussiana*, es capaz de reconocer entre los distintos sonidos: Disparo real, disparo de salva, explosivos y efectos de sonido. Con una probabilidad de clasificar correctamente las muestras arriba del 95%.

Para trabajos futuros, experimentar con otros métodos de minería de datos, tomar en cuenta si los sonidos de disparo son en un ambiente abierto o cerrado, ya que si se toman las observaciones de un ambiente abierto hay probabilidad de captar el ruido de otros sonidos y ya no sería tan fiel o puro el sonido del disparo y si se toman de un ambiente cerrado por ejemplo un stand de tiro si hay una alta probabilidad de captar un sonido más puro del disparo, experimentar con otros métodos de minería de datos por ejemplo un método de densidad ya que se usan cuando los *clusters* son irregulares, entrelazados o de diferente tamaño o si se desea trabajar con los mismos métodos de *clustering* empleados en este proyecto, tomar en cuenta que los *clusters* deben ser del mismo tamaño, esto es que sea el mismo número de observaciones para los diferentes sonidos, también se podría aumentar el número de observaciones que conforman la base de datos, suponiendo que se obtendría una mejor clasificación de las muestras,

Así también experimentar otra forma de tratar los parámetros extraídos de los sonidos del disparo, otra forma de normalizarlos, de promediarlos, de obtener los parámetros mismos extraídos del audio.

Tomar lapsos o intervalos más grandes o más pequeños del sonido del disparo, o tomar intervalos del mismo tamaño para todas las observaciones o muestras en base a la duración del sonido y se piensa que se obtendrían *clusters* del mismo tamaño y tal vez se obtendría un mejor resultado.

Experimentar si se puede diferenciar o reconocer el sonido de disparo entre otros sonidos no similares por ejemplo sonidos ambientales, empleando los mismos datos, y los mismos algoritmos de minería de datos usados en este proyecto tecnológico.

Para obtener la matriz de distancias con *DTW*, experimentar con parámetros en el dominio del tiempo o usar series de tiempo, ya que en este proyecto se utilizaron las frecuencias para obtener la matriz de distancias con el *DTW*.

Referencias bibliográficas

[1] Bernal Ávila, Jorge Jaime, “**Plataforma web para visualizar videos y notificar actividades**”, Proyecto Terminal, División de Ciencias Básicas e Ingeniería, Universidad Autónoma Metropolitana Azcapotzalco, México, 2010.

[2] Guzmán González Nancy, “**Aplicación de Distintas Técnicas de Minería de Datos para el Tratamiento de Información**”, Proyecto Terminal, División de Ciencias Básicas e Ingeniería, Universidad Autónoma Metropolitana Azcapotzalco, México, 2011.

[3] Hernández Nieves, María Sara, “**Clasificador de llantos de bebe con wavelets**”, Proyecto Terminal, División de Ciencias Básicas e Ingeniería, Universidad Autónoma Metropolitana Azcapotzalco, México, 2012.

[4] Valero Téllez, Alberto, "Extracción de información con algoritmos de clasificación", Tesis de maestría, Instituto Nacional de Astrofísica, Óptica y Electrónica, Puebla, 2005.

[5] Aguilar, Juan, “Educación Ejercito de Chile”, “**Sistema de Sensores acústicos para detección y localización automática de disparos de armas de fuego en el espacio público**”, v.38 fasc.1, p -1,2011

[6] Arrieta,A.G; Romero, L.A; Lazaro,A. L. S.; Garcia, J.R; Del Amos Rodriguez, J. M, “**NEURAL NETWORKS APPLIED TO GUN AND AMMUNITION RECOGNITION FROM SHOOTING SOUNDS**”,Security Technology, IEEE 33rd Annual1999 International Carnahan Conference on, pp. 67-70,1999.

[7] Smith, Michael; Buscemi, Scott; Xu, Dexiang Jhon; Ned T%E, Ph. D., “**Gunshot Detection System for JTRS Radio**”, pp. 266-271, 2010.

[8] Sallai, János; Lédeczi, Ákos; Vólgyesi, Péter, “**Acoustic Shooter Localization with a Minimal Number of Single-Channel Wireless Sensor Nodes**”, [En línea], Disponible en: <http://www.isis.vanderbilt.edu/sites/default/files/49.pdf> [Consultada el 14 de mayo de 2014]

[9] Lonescu, Bogdan; Mironică, Lonut; Schlüter, Jan; Schedi, Markus, “**A Naïve Mid-level Concept-based Fusion Approach to Violence Detection in Hollywood Movies**”, pp. 215-222.

[10] Navrátil, Milan; Kresálek, Vojtech; Dostálek, Petr, “**Neural network classification of gunshots using spectral characteristics**”, [En línea], Disponible en: <http://www.wseas.us/e-library/conferences/2011/Lanzarote/ACMOS/ACMOS-50.pdf> [Consultada el 13 de mayo de 2014]

[11] “**ShotSpotter**”, [En línea], Comercial en: <http://www.shotspotter.com/> [consultada el 30 de enero de 2014]

- [12] Giorgino Toni, “*Computing and Visualizing Dynamic Time Warping Alignments in R: The dtw Package*”, “Journal of Statistical Software”, Vol 31, Issue 7, 2009.
- [13] Brock, Guy; Datta Susmita; Pihur, Vasy; Data Somnath, “*cIValid: An R Package for Cluster Validation*”, “Journal of Statistical Software”, Vol 25, Issue 4, 2008.
- [14] “*Oline-convert.com*”, [En línea], Disponible en: <http://www.online-convert.com>. [Consultada el 31 de enero de 2014]
- [15] “*Praat*” versión 5.3.64, [En línea], Disponible en: http://www.fon.hum.uva.nl/praat/download_win.html [Consultada el 30 de enero de 2014]
- [16] Máquinas de vectores de soporte, [En línea], Disponible en: http://es.wikipedia.org/wiki/M%C3%A1quinas_de_vectores_de_soporte [Consultada el 19 de agosto de 2014]
- [17] Betancourt, Gustavo A., Las máquinas de soporte vectorial (SVMs), [En línea], Disponible en: <http://revistas.utp.edu.co/index.php/revistaciencia/article/viewFile/6895/4139> [Consultada el 19 de agosto de 2014]
- [18] Meyer, David, “*Support Vector Machines*”, [En línea], Disponible en: <http://cran.r-project.org/web/packages/e1071/vignettes/svmdoc.pdf> [Consultada el 29 de julio de 2014]
- [19] “*Análisis Cluster*”, [En línea], Disponible en: https://www.uam.es/personal_pdi/economicas/rmc/documentos/cluster.PDF [Consultada el 28 de agosto de 2014]
- [20] Santiago de la Fuente, Fernández. Análisis Conglomerados, [En línea]. Disponible en: <http://www.fuenterrebollo.com/Economicas/ECONOMETRIA/SEGMENTACION/CONGLOMERADOS/conglomerados.pdf> [Consultada el 28 de agosto de 2014]
- [21] Lüer Villagra. Armin; Venegas Quintrileo. Barbara; Bustos Gómez. Jaime., Estrategias de paralelización de metaheurísticas aplicadas a problemas de localización de instalaciones., [En línea]. Disponible en: <file:///C:/Users/Administrador/Downloads/Dialnet-ESTRATEGIASDEPARALELIZACIONDEMETAHEURISTICASAPLICA-3998904.pdf> [Consultada el 28 de agosto de 2014]
- [22] principios en procesamiento de señales, [En línea]. Disponible en: <http://www.oocities.org/vifibio/05PRINCIPIOSENPROCESAMIENTODESENALES.PDF> [Consultada el 28 de agosto de 2014]
- [23] Mariel Natali. Pamela, Análisis cluster: una aplicación a los departamentos de la provincia de Córdoba. [En línea], Disponible en: <http://www.eco.unrc.edu.ar/wp-content/uploads/2010/04/TRABAJO-N%C2%BA-32.pdf> [Consultada el 5 de junio de 2014]

[24] Pascual. D; Pla. F; Sánchez. S, Algoritmos de agrupamiento, [En línea], Disponible en: http://marmota.dlsi.uji.es/WebBIB/papers/2007/1_Pascual-MIA-2007.pdf [Consultada] [Consultada el 5 de junio de 2014]

[25] Transformada de Fourier. [En línea]. Disponible en: http://es.wikipedia.org/wiki/Transformada_de_Fourie [Consultada el 28 de junio de 2014]

[26] Gómez Chong. Paul Dremyn, Algoritmo FFT. [En línea]. Disponible en: <http://es.scribd.com/doc/2372867/algoritmo-FFT>. [Consultada el 28 de junio de 2014]

[27] Meyer, David. "Support Vector Machines". [En línea]. Disponible en: <http://cran.r-project.org/web/packages/e1071/vignettes/svmdoc.pdf>. [Consultada el 29 de julio de 2014]

[28] [En línea]. Disponible en: <http://www.vallechicama.com/2014/06/roban-dos-pistolas-de-comisaria-de.html>. [Consultada el 24 de mayo de 2014]

[29] dtw, [En línea], Disponible en: <http://cran.r-project.org/web/packages/dtw/index.html> [Consultada el 29 de julio de 2014]

[30] e1071, [En línea], Disponible en: <http://cran.r-project.org/web/packages/e1071/index.html> [Consultada el 29 de julio de 2014]

[31] clValid, [En línea], Disponible en: <http://cran.r-project.org/web/packages/clValid/index.html> [Consultada el 9 de junio de 2014]

[32] R 3.0.2. [En línea], Disponible en: <http://cran.r-project.org/> [Consultada el 30 de enero de 2014]

[33] Rousseeuw, P.J.; Kaufman, L. (1990). *Finding Groups in Data: An Introduction to Cluster Analysis*. Wiley.

[34] James W. Cooley & John W. Tukey (1965): "An algorithm for the machine calculation of complex Fourier series", Math. Comput. **19**, 297–301.

[35] Jerzy Stefanowski. Data Mining – Clustering. [En línea]. Poznan, Poland. <http://www.cs.put.poznan.pl/jstefanowski/sed/DM-7clusteringnew.pdf>. [Consultada el 5 de julio de 2014]

Apéndice A

Códigos en C++

A.1 Código en C ++ para la obtención de frecuencias en escala logarítmica

```
#include<stdio.h>
#include<stdlib.h>
#include<iostream>
#include<math.h>
#include<string.h>
#include<fstream>

using namespace std;

int main()
{ // se inicializan variables
  double num[52770][2], suma=0, number, menor, suma2=0, max=-9999;
  double n=0, m=0;
  double promedio[500];
  char frec[100], leido1[20];
  char Preal[100], leido2[20];
  char Pimag[100], leido3[20];
  int ciclo;
  char Earchivo[15], Sarchivo[9];
  int i=0, j, tam=0, k=1, cont=0, z, a=0;
  double Pr, Pi;
  do{
    //nombre del archivo que se desea abrir
    cout<<"Nombre del Archivo de entrada: ";
    cin>>Earchivo;
    cout<<endl;

    ifstream archivo(Earchivo, ios::in);

    if(!archivo)
    {
      cout<<"no abrio el archivo"<<endl;
      exit(1);
    }

    while (!archivo.eof()) //comienza a leer el archivo
    {
      archivo >> frec >> Preal >> Pimag;
      Pr=atof(Preal);
      Pi=atof(Pimag);
      //almacenando en la matriz num la frecuencia y la suma
      de la parte imaginaria y real
      num[i][0]=atof(frec);
      num[i][1]=sqrt(pow(Pr,2)+pow(Pi,2));
    }
  }
}
```



```

        i=i+1;
    }
    tam=i;
    cout<<tam<<endl;
    cout<<i<<endl;
    while(k<tam)
    {
        n=n+0.1;
        m=n-0.1;
        number=pow(10,n);
        menor=pow(10,m);
        // realizando el promedio en escala logarítmica
        for( j=k;num[j][0]<number;j++)
        {
            suma=suma + num[j][1];
            cont=cont + 1;
        }
        promedio[a]=suma;
        a=a+1;
        suma2=0;
        suma=0;
        cont=0;
        k=j;
    }
    a=a-1;
    for(int b=0;b<a;b++)
    {
        if(max<promedio[b])
        {
            max=promedio[b];
        }
    }
    //almacenando el promedio en escala logarítmica en un archivo
    //de salida dado
    cout<<"Nombre del Archivo de Salida: ";
    cin>>Sarchivo;
    cout<<endl;

    for(int b=0;b<a;b++)
    {
        ofstream Elog(Sarchivo,ios::app);
        Elog <<promedio[b]/max<<endl;
        Elog.close();
    }
    archivo.close();
    for(int d=0;d<=tam;d++)
    {num[d][0];
    num[d][1];
    }
    //inicializando las variables utilizados a cero
    suma=0;
    number=0;
    menor=0;
    suma2=0;
    max=-9999;
    n=0,m=0;

```

```
memset(promedio,0,500);
memset(frec,0,100);
memset(Preal,0,100);
memset(Pimag,0,100);
memset(leido1,0,20);
memset(leido2,0,20);
memset(leido3,0,20);
memset(Earchivo,0,15);
memset(Sarchivo,0,9);
i=0;
j=0;
tam=0;
k=1;
cont=0;
z=0;
a=0;
Pr=0;
Pi=0;
cout<<endl<<"oprime 1 si quieres leer otro archivo: ";
cin>>ciclo;
cout<<endl;
system("cls");
}while(ciclo==1);//iniciando ciclo si se desea leer otro archivo
system("pause");
return 0;
}
```

A.2 Código en C++ para la obtención de frecuencias por bandas de frecuencia de 50 en 50 Hz

```
#include<stdio.h>
#include<stdlib.h>
#include<iostream>
#include<math.h>
#include<string.h>
#include<fstream>

using namespace std;

int main()
{    //inicializan variables
    double num[52770][2], suma=0, number, menor, suma2=0, max=-9999;
    double n=0, m=0;
    double promedio[500], promfrec[500];
    char frec[100], leido1[20];
    char Preal[100], leido2[20];
    char Pimag[100], leido3[20];
    int ciclo;
    char Earchivo[15], Sarchivo[9];
    int i=0, j, tam=0, k=1, cont=0, z, a=0;
    double Pr, Pi;
    do{
        // se nombra el archivo donde se encuentran las frecuencias
        cout<<"Nombre del Archivo de entrada: ";
        cin>>Earchivo;
        cout<<endl;

        ifstream archivo(Earchivo, ios::in);

        if(!archivo)
        {
            cout<<"no abrio el archivo"<<endl;
            exit(1);
        }
        //se lee el archivo
        while (!archivo.eof())
        {
            archivo >> frec >> Preal >> Pimag;
            Pr=atof(Preal);
            Pi=atof(Pimag);
            // se almacena la frecuencia en la matriz num
            num[i][0]=atof(frec);
            num[i][1]=sqrt(pow(Pr, 2)+pow(Pi, 2));
            i=i+1;
        }
        tam=i;
        cout<<tam<<endl;
        cout<<i<<endl;
        //inicia el promedio por bandas de frecuencia de 50 en 50 Hz
        while(k<tam)
        {
```

```

n=n+50;
for( j=k;num[j][0]<=n;j++)
{
    suma=suma + num[j][1];
    cont=cont + 1;
}
if(suma != 0)
{
    promedio[a]=suma;
    a=a+1;
}
suma2=0;
suma=0;
cont=0;
k=j;
}
for(int b=0;b<a;b++)
{
    if(max<promedio[b])
    {
        max=promedio[b];
    }
}
// se proporciona el nombre del archivo de salida con
//extension que almacenara los promedios de la frecuencia
cout<<"Nomre de Archivo de Salida: ";
cin>>Sarchivo;
cout<<endl;

for(int b=0;b<a;b++)
{
    ofstream Elog(Sarchivo,ios::app);
    Elog <<promedio[b]/max<<endl;
    Elog.close();
}
archivo.close();
// se inicializan las variables a cero
for(int d=0;d<=tam;d++)
{num[d][0];
num[d][1];
}
suma=0;
number=0;
menor=0;
suma2=0;
max=-9999;
n=0,m=0;
memset(promedio,0,500);
memset(frec,0,100);
memset(Preal,0,100);
memset(Pimag,0,100);
memset(leido1,0,20);
memset(leido2,0,20);
memset(leido3,0,20);
memset(promfrec,0,500);
memset(Earchivo,0,15);
memset(Sarchivo,0,9);

```

```
    i=0;
    j=0;
    tam=0;
    k=1;
    cont=0;
    z=0;
    a=0;
    Pr=0;
    Pi=0;
    cout<<endl<<"oprime 1 si quieres leer otro archivo: ";
    cin>>ciclo;
    cout<<endl;
    system("cls");
}while(ciclo==1);// se inicia el ciclo si se oprime 1
system("pause");
return 0;
```

```
}
```

Apéndice B

Códigos en R

B.1 Algoritmos de clustering

B.1.1 K-Means

- **K-Means con 2 clusters**

```
#Método K-Means
t<-proc.time()#inicia tiempo de ejecución
distk <-dist(Tsonidos,method="euclidean")#obtiene la matriz de distancias de similitud
(Reskmeans <- kmeans(distk,2))#implementación de K-Means con 2 clusters
proc.time()-t# finaliza tiempo de ejecución
table(datos$MuestraSonidos,Reskmeans$cluster)#muestra tabla de clustering
#grafica el clustering
clusplot(Tsonidos,Reskmeans$cluster,color=TRUE,shade=TRUE,main="K-medias con K=2")
#se obtiene índice "Silhoutte"
sil <- silhouette(Reskmeans$cluster,dist=distk)
sil
#gráfica el índice "Silhoutte"
plot(sil,main="Silueta kmeans con 2 clusters", border="blue")
```

- **K-Means con 3 clusters**

```
#Método K-Means
t<-proc.time()#inicia tiempo de ejecución
distk <-dist(Tsonidos,method="euclidean")#obtiene la matriz de distancias de similitud
(Reskmeans <- kmeans(distk,3))#implementación de K-Means con 3 clusters
proc.time()-t# finaliza tiempo de ejecución
table(datos$MuestraSonidos,Reskmeans$cluster)#muestra tabla de clustering
#grafica el clustering
clusplot(Tsonidos,Reskmeans$cluster,color=TRUE,shade=TRUE,main="K-medias con K=3")
#se obtiene índice "Silhoutte"
sil <- silhouette(Reskmeans$cluster,dist=distk)
sil
#gráfica el índice "Silhoutte"
plot(sil,main="Silueta kmeans con 3 clusters", border="blue")
```

- **K-Means con 4 clusters**

```
#Método K-Means
t<-proc.time()#inicia tiempo de ejecución
distk <-dist(Tsonidos,method="euclidean")#obtiene la matriz de distancias de similitud
```

```

(Reskmeans <- kmeans(distk,4))#implementación de K-Means con 4 clusters
proc.time()-t# finaliza tiempo de ejecución
table(datos$MuestraSonidos,Reskmeans$cluster)#muestra tabla de clustering
#grafica el clustering
clusplot(Tsonidos,Reskmeans$cluster,color=TRUE,shade=TRUE,main="K-medias con K=4")
#se obtiene índice "Silhoutte"
sil <- silhouette(Reskmeans$cluster,dist=distk)
sil
#gráfica el índice "Silhoutte"
plot(sil,main="Silueta kmeans con 4 clusters", border="blue")

```

- **K-Means con 5 clusters**

```

#Método K-Means
t<-proc.time()#inicia tiempo de ejecución
distk <-dist(Tsonidos,method="euclidean")#obtiene la matriz de distancias de similitud
(Reskmeans <- kmeans(distk,5))#implementación de K-Means con 5 clusters
proc.time()-t# finaliza tiempo de ejecución
table(datos$MuestraSonidos,Reskmeans$cluster)#muestra tabla de clustering
#grafica el clustering
clusplot(Tsonidos,Reskmeans$cluster,color=TRUE,shade=TRUE,main="K-medias con K=5")
#se obtiene índice "Silhoutte"
sil <- silhouette(Reskmeans$cluster,dist=distk)
sil
#gráfica el índice "Silhoutte"
plot(sil,main="Silueta kmeans con 5 clusters", border="blue")

```

- **K-Means con 6 clusters**

```

#Método K-Means
t<-proc.time()#inicia tiempo de ejecución
distk <-dist(Tsonidos,method="euclidean")#obtiene la matriz de distancias de similitud
(Reskmeans <- kmeans(distk,6))#implementación de K-Means con 6 clusters
proc.time()-t# finaliza tiempo de ejecución
table(datos$MuestraSonidos,Reskmeans$cluster)#muestra tabla de clustering
#grafica el clustering
clusplot(Tsonidos,Reskmeans$cluster,color=TRUE,shade=TRUE,main="K-medias con K=6")
#se obtiene índice "Silhoutte"
sil <- silhouette(Reskmeans$cluster,dist=distk)
sil
#gráfica el índice "Silhoutte"
plot(sil,main="Silueta kmeans con 6 clusters", border="blue")

```

- **K-Means con 7 clusters**

```

#Método K-Means
t<-proc.time()#inicia tiempo de ejecución
distk <-dist(Tsonidos,method="euclidean")#obtiene la matriz de distancias de similitud

```

```
(Reskmeans <- kmeans(distk,7))#implementación de K-Means con 7 clusters
proc.time()-t# finaliza tiempo de ejecución
table(datos$MuestraSonidos,Reskmeans$cluster)#muestra tabla de clustering
#grafica el clustering
clusplot(Tsonidos,Reskmeans$cluster,color=TRUE,shade=TRUE,main="K-medias con K=7")
#se obtiene índice "Silhoutte"
sil <- silhouette(Reskmeans$cluster,dist=distk)
sil
#gráfica el índice "Silhoutte"
plot(sil,main="Silueta kmeans con 7 clusters", border="blue")
```

- **K-Means con 8 clusters**

```
#Método K-Means
t<-proc.time()#inicia tiempo de ejecución
distk <-dist(Tsonidos,method="euclidean")#obtiene la matriz de distancias de similitud
(Reskmeans <- kmeans(distk,8))#implementación de K-Means con 8 clusters
proc.time()-t# finaliza tiempo de ejecución
table(datos$MuestraSonidos,Reskmeans$cluster)#muestra tabla de clustering
#grafica el clustering
clusplot(Tsonidos,Reskmeans$cluster,color=TRUE,shade=TRUE,main="K-medias con K=8")
#se obtiene índice "Silhoutte"
sil <- silhouette(Reskmeans$cluster,dist=distk)
sil
#gráfica el índice "Silhoutte"
plot(sil,main="Silueta kmeans con 8 clusters", border="blue")
```

- **K-Means con 9 clusters**

```
#Método K-Means
t<-proc.time()#inicia tiempo de ejecución
distk <-dist(Tsonidos,method="euclidean")#obtiene la matriz de distancias de similitud
(Reskmeans <- kmeans(distk,9))#implementación de K-Means con 9 clusters
proc.time()-t# finaliza tiempo de ejecución
table(datos$MuestraSonidos,Reskmeans$cluster)#muestra tabla de clustering
#grafica el clustering
clusplot(Tsonidos,Reskmeans$cluster,color=TRUE,shade=TRUE,main="K-medias con K=9")
#se obtiene índice "Silhoutte"
sil <- silhouette(Reskmeans$cluster,dist=distk)
sil
#gráfica el índice "Silhoutte"
plot(sil,main="Silueta kmeans con 9 clusters", border="blue")
```

- **K-Means con 10 clusters**

```
#Método K-Means
t<-proc.time()#inicia tiempo de ejecución
distk <-dist(Tsonidos,method="euclidean")#obtiene la matriz de distancias de similitud
(Reskmeans <- kmeans(distk,10))#implementación de K-Means con 10 clusters
```



```

proc.time()-t# finaliza tiempo de ejecución
table(datos$MuestraSonidos,Reskmeans$cluster)#muestra tabla de clustering
#grafica el clustering
clusplot(Tsonidos,Reskmeans$cluster,color=TRUE,shade=TRUE,main="K-medias con K=10")
#se obtiene índice "Silhoutte"
sil <- silhouette(Reskmeans$cluster,dist=distk)
sil
#gráfica el índice "Silhoutte"
plot(sil,main="Silueta kmeans con 10 clusters", border="blue")

```

B.1.2 K-Medoids

- **K-Medoids con 2 cluster**

```

t<-proc.time()#inicia tiempo de ejecución
Rpam <- pam(Tsonidos,2,metric="euclidean")#implementación del método K-Medoids
proc.time()-t #finaliza tiempo de ejecución
table(datos$MuestraSonidos,Rpam$clustering)#muestra tabla de clustering
#grafica el clustering
clusplot(Tsonidos,Rpam$clustering,color=TRUE,shade=TRUE,main="K-Medoids con 2 clusters")
#se obtiene índice "Silhoutte"
sil <- silhouette(Rpam)
#gráfica el índice "Silhoutte"
plot(sil,main="Silueta K-Medoids 2", border="blue")
Rpam$clustering
sil

```

- **K-Medoids con 3 cluster**

```

t<-proc.time()#inicia tiempo de ejecución
Rpam <- pam(Tsonidos,3,metric="euclidean")#implementación del método K-Medoids
proc.time()-t #finaliza tiempo de ejecución
table(datos$MuestraSonidos,Rpam$clustering)#muestra tabla de clustering
#grafica el clustering
clusplot(Tsonidos,Rpam$clustering,color=TRUE,shade=TRUE,main="K-Medoids con 3 clusters")
#se obtiene índice "Silhoutte"
sil <- silhouette(Rpam)
#gráfica el índice "Silhoutte"
plot(sil,main="Silueta K-Medoids 3", border="blue")
Rpam$clustering
sil

```

- **K-Medoids con 4 cluster**

```

t<-proc.time()#inicia tiempo de ejecución
Rpam <- pam(Tsonidos,4,metric="euclidean")#implementación del método K-Medoids
proc.time()-t #finaliza tiempo de ejecución
table(datos$MuestraSonidos,Rpam$clustering)#muestra tabla de clustering
#grafica el clustering
clusplot(Tsonidos,Rpam$clustering,color=TRUE,shade=TRUE,main="K-Medoids con 4 clusters")

```

```
#se obtiene índice "Silhoutte"
sil <- silhouette(Rpam)
#gráfica el índice "Silhoutte"
plot(sil,main="Silueta K-Medoids 4", border="blue")
Rpam$clustering
sil
```

- **K-Medoids con 5 cluster**

```
t<-proc.time()#inicia tiempo de ejecución
Rpam <- pam(Tsonidos,5,metric="euclidean")#implementación del método K-Medoids
proc.time()-t #finaliza tiempo de ejecución
table(datos$MuestraSonidos,Rpam$clustering)#muestra tabla de clustering
#gráfica el clustering
clusplot(Tsonidos,Rpam$clustering,color=TRUE,shade=TRUE,main="K-Medoids con 5 clusters")
#se obtiene índice "Silhoutte"
sil <- silhouette(Rpam)
#gráfica el índice "Silhoutte"
plot(sil,main="Silueta K-Medoids 5", border="blue")
Rpam$clustering
sil
```

- **K-Medoids con 6 cluster**

```
t<-proc.time()#inicia tiempo de ejecución
Rpam <- pam(Tsonidos,6,metric="euclidean")#implementación del método K-Medoids
proc.time()-t #finaliza tiempo de ejecución
table(datos$MuestraSonidos,Rpam$clustering)#muestra tabla de clustering
#gráfica el clustering
clusplot(Tsonidos,Rpam$clustering,color=TRUE,shade=TRUE,main="K-Medoids con 6 clusters")
#se obtiene índice "Silhoutte"
sil <- silhouette(Rpam)
#gráfica el índice "Silhoutte"
plot(sil,main="Silueta K-Medoids 6", border="blue")
Rpam$clustering
sil
```

- **K-Medoids con 7 cluster**

```
t<-proc.time()#inicia tiempo de ejecución
Rpam <- pam(Tsonidos,7,metric="euclidean")#implementación del método K-Medoids
proc.time()-t #finaliza tiempo de ejecución
table(datos$MuestraSonidos,Rpam$clustering)#muestra tabla de clustering
#gráfica el clustering
clusplot(Tsonidos,Rpam$clustering,color=TRUE,shade=TRUE,main="K-Medoids con 7 clusters")
#se obtiene índice "Silhoutte"
sil <- silhouette(Rpam)
#gráfica el índice "Silhoutte"
plot(sil,main="Silueta K-Medoids 7", border="blue")
Rpam$clustering
sil
```

- **K-Medoids con 8 cluster**

```
t<-proc.time()#inicia tiempo de ejecución
Rpam <- pam(Tsonidos,8,metric="euclidean")#implementación del método K-Medoids
proc.time()-t #finaliza tiempo de ejecución
table(datos$MuestraSonidos,Rpam$clustering)#muestra tabla de clustering
#grafica el clustering
clusplot(Tsonidos,Rpam$clustering,color=TRUE,shade=TRUE,main="K-Medoids con 8 clusters")
#se obtiene índice "Silhoutte"
sil <- silhouette(Rpam)
#gráfica el índice "Silhoutte"
plot(sil,main="Silueta K-Medoids 8", border="blue")
Rpam$clustering
sil
```

- **K-Medoids con 9 cluster**

```
t<-proc.time()#inicia tiempo de ejecución
Rpam <- pam(Tsonidos,9,metric="euclidean")#implementación del método K-Medoids
proc.time()-t #finaliza tiempo de ejecución
table(datos$MuestraSonidos,Rpam$clustering)#muestra tabla de clustering
#grafica el clustering
clusplot(Tsonidos,Rpam$clustering,color=TRUE,shade=TRUE,main="K-Medoids con 9 clusters")
#se obtiene índice "Silhoutte"
sil <- silhouette(Rpam)
#gráfica el índice "Silhoutte"
plot(sil,main="Silueta K-Medoids 9", border="blue")
Rpam$clustering
sil
```

- **K-Medoids con 10 cluster**

```
t<-proc.time()#inicia tiempo de ejecución
Rpam <- pam(Tsonidos,10,metric="euclidean")#implementación del método K-Medoids
proc.time()-t #finaliza tiempo de ejecución
table(datos$MuestraSonidos,Rpam$clustering)#muestra tabla de clustering
#grafica el clustering
clusplot(Tsonidos,Rpam$clustering,color=TRUE,shade=TRUE,main="K-Medoids con 5 clusters")
#se obtiene índice "Silhoutte"
sil <- silhouette(Rpam)
#gráfica el índice "Silhoutte"
plot(sil,main="Silueta K-Medoids 10", border="blue")
Rpam$clustering
sil
```

B.1.3 Método Jerárquico

- **Jerárquico con 2 clusters**

```
# Método jerárquico usando método WARD
```

```

t<-proc.time()#inicia tiempo de ejecución
distancia <-daisy(Tsonidos,metric="euclidean")#obtiene la matriz de distancias de similitud
Rjerarquico <- hclust(distancia,method="ward.D")#implementación del método Jerárquico
groups <- cutree(Rjerarquico,k=2)#formacion de 2 grupos
proc.time()-t#finaliza el tiempo de ejecución
#grafica del conglomerado
clusplot(Tsonidos,groups,color=TRUE,shade=TRUE,main="Hierarchical con 2 clusters")
# graficando dendograma
plot(Rjerarquico,labels=datos$MuestraSonidos,cex=0.3,col=as.numeric(datos$MuestraSonidos),ma
in="Hierarchical con 2 cluster")
#cortando el dendograma en los grupos
x <- rect.hclust(Rjerarquico,k=2,border="red")
table(datos$MuestraSonidos,groups)#muestra tabla de clustering
table(groups)#tamaño de los clusters
x#muestra las observaciones que corresponden a cierto grupos
sil <- silhouette(groups,dist=distancia)#obtiene índice silueta
sil#imprime valores del índice silueta
plot(sil,main="Silueta Jerárquico 2", border="blue")#gráfica el índice silueta

```

- **Jerárquico con 3 clusters**

Método jerárquico usando método WARD

```

t<-proc.time()#inicia tiempo de ejecución
distancia <-daisy(Tsonidos,metric="euclidean")#obtiene la matriz de distancias de similitud
Rjerarquico <- hclust(distancia,method="ward.D")#implementación del método Jerárquico
groups <- cutree(Rjerarquico,k=3)#formacion de 3 grupos
proc.time()-t#finaliza el tiempo de ejecución
#grafica del conglomerado
clusplot(Tsonidos,groups,color=TRUE,shade=TRUE,main="Hierarchical con 3 clusters")
# graficando dendograma
plot(Rjerarquico,labels=datos$MuestraSonidos,cex=0.3,col=as.numeric(datos$MuestraSonidos),ma
in="Hierarchical con 3 cluster")
#cortando el dendograma en los grupos
x <- rect.hclust(Rjerarquico,k=3,border="red")
table(datos$MuestraSonidos,groups)#muestra tabla de clustering
table(groups)#tamaño de los clusters
x#muestra las observaciones que corresponden a cierto grupos
sil <- silhouette(groups,dist=distancia)#obtiene índice silueta
sil#imprime valores del índice silueta
plot(sil,main="Silueta Jerárquico 3", border="blue")#gráfica el índice silueta

```

- **Jerárquico con 4 clusters**

Método jerárquico usando método WARD

```

t<-proc.time()#inicia tiempo de ejecución
distancia <-daisy(Tsonidos,metric="euclidean")#obtiene la matriz de distancias de similitud
Rjerarquico <- hclust(distancia,method="ward.D")#implementación del método Jerárquico
groups <- cutree(Rjerarquico,k=4)#formacion de 4 grupos

```

```

proc.time()-t##finaliza el tiempo de ejecución
#grafica del conglomerado
clusplot(Tsonidos,groups,color=TRUE,shade=TRUE,main="Hierarchical con 4 clusters")
# graficando dendograma
plot(Rjerarquico,labels=datos$MuestraSonidos,cex=0.3,col=as.numeric(datos$MuestraSonidos),main="Hierarchical con 4 cluster")
#cortando el dendograma en los grupos
x <- rect.hclust(Rjerarquico,k=4,border="red")
table(datos$MuestraSonidos,groups)#muestra tabla de clustering
table(groups)#tamaño de los clusters
x##muestra las observaciones que corresponden a cierto grupos
sil <- silhouette(groups,dist=distancia)#obtiene índice silueta
sil##imprime valores del índice silueta
plot(sil,main="Silueta Jerárquico 4", border="blue")#gráfica el índice silueta

```

- **Jerárquico con 5 clusters**

Método jerárquico usando método WARD

```

t<-proc.time()#inicia tiempo de ejecución
distancia <-daisy(Tsonidos,metric="euclidean")#obtiene la matriz de distancias de similitud
Rjerarquico <- hclust(distancia,method="ward.D")#implementación del método Jerárquico
groups <- cutree(Rjerarquico,k=5)#formacion de 5 grupos
proc.time()-t##finaliza el tiempo de ejecución
#grafica del conglomerado
clusplot(Tsonidos,groups,color=TRUE,shade=TRUE,main="Hierarchical con 5 clusters")
# graficando dendograma
plot(Rjerarquico,labels=datos$MuestraSonidos,cex=0.3,col=as.numeric(datos$MuestraSonidos),main="Hierarchical con 5 cluster")
#cortando el dendograma en los grupos
x <- rect.hclust(Rjerarquico,k=5,border="red")
table(datos$MuestraSonidos,groups)#muestra tabla de clustering
table(groups)#tamaño de los clusters
x##muestra las observaciones que corresponden a cierto grupos
sil <- silhouette(groups,dist=distancia)#obtiene índice silueta
sil##imprime valores del índice silueta
plot(sil,main="Silueta Jerárquico 5", border="blue")#gráfica el índice silueta

```

- **Jerárquico con 6 clusters**

Método jerárquico usando método WARD

```

t<-proc.time()#inicia tiempo de ejecución
distancia <-daisy(Tsonidos,metric="euclidean")#obtiene la matriz de distancias de similitud
Rjerarquico <- hclust(distancia,method="ward.D")#implementación del método Jerárquico
groups <- cutree(Rjerarquico,k=6)#formacion de 6 grupos
proc.time()-t##finaliza el tiempo de ejecución
#grafica del conglomerado
clusplot(Tsonidos,groups,color=TRUE,shade=TRUE,main="Hierarchical con 6 clusters")
# graficando dendograma
plot(Rjerarquico,labels=datos$MuestraSonidos,cex=0.3,col=as.numeric(datos$MuestraSonidos),main="Hierarchical con 6 cluster")
#cortando el dendograma en los grupos
x <- rect.hclust(Rjerarquico,k=6,border="red")
table(datos$MuestraSonidos,groups)#muestra tabla de clustering

```

```

table(groups)#tamaño de los clusters
x#muestra las observaciones que corresponden a cierto grupos
sil <- silhouette(groups,dist=distancia)#obtiene índice silueta
sil#imprime valores del índice silueta
plot(sil,main="Silueta Jerárquico 6", border="blue")#gráfica el índice silueta

```

- **Jerárquico con 7 clusters**

```
# Método jerárquico usando método WARD
```

```

t<-proc.time()#inicia tiempo de ejecución
distancia <-daisy(Tsonidos,metric="euclidean")#obtiene la matriz de distancias de similitud
Rjerarquico <- hclust(distancia,method="ward.D")#implementación del método Jerárquico
groups <- cutree(Rjerarquico,k=7)#formacion de 7 grupos
proc.time()-t#finaliza el tiempo de ejecución
#grafica del conglomerado
clusplot(Tsonidos,groups,color=TRUE,shade=TRUE,main="Hierarchical con 7 clusters")
# graficando dendograma
plot(Rjerarquico,labels=datos$MuestraSonidos,cex=0.3,col=as.numeric(datos$MuestraSonidos),ma
in="Hierarchical con 7 cluster")
#cortando el dendograma en los grupos
x <- rect.hclust(Rjerarquico,k=7,border="red")
table(datos$MuestraSonidos,groups)#muestra tabla de clustering
table(groups)#tamaño de los clusters
x#muestra las observaciones que corresponden a cierto grupos
sil <- silhouette(groups,dist=distancia)#obtiene índice silueta
sil#imprime valores del índice silueta
plot(sil,main="Silueta Jerárquico 7", border="blue")#gráfica el índice silueta

```

- **Jerárquico con 8 clusters**

```
# Método jerárquico usando método WARD
```

```

t<-proc.time()#inicia tiempo de ejecución
distancia <-daisy(Tsonidos,metric="euclidean")#obtiene la matriz de distancias de similitud
Rjerarquico <- hclust(distancia,method="ward.D")#implementación del método Jerárquico
groups <- cutree(Rjerarquico,k=8)#formacion de 8 grupos
proc.time()-t#finaliza el tiempo de ejecución
#grafica del conglomerado
clusplot(Tsonidos,groups,color=TRUE,shade=TRUE,main="Hierarchical con 8 clusters")
# graficando dendograma
plot(Rjerarquico,labels=datos$MuestraSonidos,cex=0.3,col=as.numeric(datos$MuestraSonidos),ma
in="Hierarchical con 8 cluster")
#cortando el dendograma en los grupos
x <- rect.hclust(Rjerarquico,k=8,border="red")
table(datos$MuestraSonidos,groups)#muestra tabla de clustering
table(groups)#tamaño de los clusters
x#muestra las observaciones que corresponden a cierto grupos
sil <- silhouette(groups,dist=distancia)#obtiene índice silueta
sil#imprime valores del índice silueta
plot(sil,main="Silueta Jerárquico 8", border="blue")#gráfica el índice silueta

```

- **Jerárquico con 9 clusters**

Método jerárquico usando método WARD

```
t<-proc.time()#inicia tiempo de ejecución
distancia <-daisy(Tsonidos,metric="euclidean")#obtiene la matriz de distancias de similitud
Rjerarquico <- hclust(distancia,method="ward.D")#implementación del método Jerárquico
groups <- cutree(Rjerarquico,k=9)#formacion de 9 grupos
proc.time()-t#finaliza el tiempo de ejecución
#grafica del conglomerado
clusplot(Tsonidos,groups,color=TRUE,shade=TRUE,main="Hierarchical con 9 clusters")
# graficando dendograma
plot(Rjerarquico,labels=datos$MuestraSonidos,cex=0.3,col=as.numeric(datos$MuestraSonidos),ma
in="Hierarchical con 9 cluster")
#cortando el dendograma en los grupos
x <- rect.hclust(Rjerarquico,k=9,border="red")
table(datos$MuestraSonidos,groups)#muestra tabla de clustering
table(groups)#tamaño de los clusters
x#muestra las observaciones que corresponden a cierto grupos
sil <- silhouette(groups,dist=distancia)#obtiene índice silueta
sil#imprime valores del índice silueta
plot(sil,main="Silueta Jerárquico 9", border="blue")#gráfica el índice silueta
```

- **Jerárquico con 10 clusters**

Método jerárquico usando método WARD

```
t<-proc.time()#inicia tiempo de ejecución
distancia <-daisy(Tsonidos,metric="euclidean")#obtiene la matriz de distancias de similitud
Rjerarquico <- hclust(distancia,method="ward.D")#implementación del método Jerárquico
groups <- cutree(Rjerarquico,k=10)#formacion de 10 grupos
proc.time()-t#finaliza el tiempo de ejecución
#grafica del conglomerado
clusplot(Tsonidos,groups,color=TRUE,shade=TRUE,main="Hierarchical con 510clusters")
# graficando dendograma
plot(Rjerarquico,labels=datos$MuestraSonidos,cex=0.3,col=as.numeric(datos$MuestraSonidos),ma
in="Hierarchical con 10 cluster")
#cortando el dendograma en los grupos
x <- rect.hclust(Rjerarquico,k=10,border="red")
table(datos$MuestraSonidos,groups)#muestra tabla de clustering
table(groups)#tamaño de los clusters
x#muestra las observaciones que corresponden a cierto grupos
sil <- silhouette(groups,dist=distancia)#obtiene índice silueta
sil#imprime valores del índice silueta
plot(sil,main="Silueta Jerárquico 10", border="blue")#gráfica el índice silueta
```

B.1.4 Jerárquico con distancia “DTW”

- **Jerárquico con 2 clusters**

```
#DTW
```

```
t<-proc.time()#inicia tiempo de ejecución
distMatrix <- dist(Tsonidos,method="DTW")#se implementa DTW para obtener una matriz de
#distancia de similaridad
hcdtw <- hclust(distMatrix,method="ward.D")#aplicación de algoritmo jerárquico
groups <- cutree(hcdtw,k=2)#aglomeración en 2 clusters
proc.time()-t#finaliza el tiempo de ejecución
#grafican los clusters
clusplot(Tsonidos,groups,color=TRUE,shade=TRUE,main="Hierarchical con DTW agrupando en 2
clusters")
#gráfica el dendograma
plot(hcdtw,labels=datos$MuestraSonidos,cex=0.3,col=as.numeric(datos$MuestraSonidos),main="H
ierarchical con DTW agrupando en 2 cluster")
x <- rect.hclust(hcdtw,k=2,border="red")#partición del dendograma en cluster
table(datos$MuestraSonidos,groups)#tabla de clasificación del clustering jerarquico
table(groups)#tamaño de los clusters
x# composición de los clusters
sil <- silhouette(groups,distMatrix)#índice silueta
#gráfica la silueta
plot(sil,main="Silueta Hierarchical con DTW agrupando en 2 clusters", border="blue")
sil#valores del índice silueta
```

- **Jerárquico con 3 clusters**

```
#DTW
```

```
t<-proc.time()#inicia tiempo de ejecución
distMatrix <- dist(Tsonidos,method="DTW")#se implementa DTW para obtener una matriz de
#distancia de similaridad
hcdtw <- hclust(distMatrix,method="ward.D")#aplicación de algoritmo jerárquico
groups <- cutree(hcdtw,k=3)#aglomeración en 3 clusters
proc.time()-t#finaliza el tiempo de ejecución
#grafican los clusters
clusplot(Tsonidos,groups,color=TRUE,shade=TRUE,main="Hierarchical con DTW agrupando en 3
clusters")
#gráfica el dendograma
plot(hcdtw,labels=datos$MuestraSonidos,cex=0.3,col=as.numeric(datos$MuestraSonidos),main="H
ierarchical con DTW agrupando en 3 cluster")
x <- rect.hclust(hcdtw,k=3,border="red")#partición del dendograma en cluster
table(datos$MuestraSonidos,groups)#tabla de clasificación del clustering jerarquico
table(groups)#tamaño de los clusters
x# composición de los clusters
sil <- silhouette(groups,distMatrix)#índice silueta
#gráfica la silueta
plot(sil,main="Silueta Hierarchical con DTW agrupando en 3 clusters", border="blue")
sil#valores del índice silueta
```

- **Jerárquico con 4 clusters**

```
#DTW
```



```

t<-proc.time()#inicia tiempo de ejecución
distMatrix <- dist(Tsonidos,method="DTW")#se implementa DTW para obtener una matriz de
#distancia de similaridad
hcdtw <- hclust(distMatrix,method="ward.D")#aplicación de algoritmo jerárquico
groups <- cutree(hcdtw,k=4)#aglomeración en 4 clusters
proc.time()-t#finaliza el tiempo de ejecución
#grafican los clusters
clusplot(Tsonidos,groups,color=TRUE,shade=TRUE,main="Hierarchical con DTW agrupando en 4
clusters")
#gráfica el dendograma
plot(hcdtw,labels=datos$MuestraSonidos,cex=0.3,col=as.numeric(datos$MuestraSonidos),main="H
ierarchical con DTW agrupando en 4 cluster")
x <- rect.hclust(hcdtw,k=4,border="red")#partición del dendograma en cluster
table(datos$MuestraSonidos,groups)#tabla de clasificación del clustering jerarquico
table(groups)#tamaño de los clusters
x# composición de los clusters
sil <- silhouette(groups,distMatrix)#índice silueta
#gráfica la silueta
plot(sil,main="Silueta Hierarchical con DTW agrupando en 4 clusters", border="blue")
sil#valores del índice silueta

```

- **Jerárquico con 5 clusters**

```
#DTW
```

```

t<-proc.time()#inicia tiempo de ejecución
distMatrix <- dist(Tsonidos,method="DTW")#se implementa DTW para obtener una matriz de
#distancia de similaridad
hcdtw <- hclust(distMatrix,method="ward.D")#aplicación de algoritmo jerárquico
groups <- cutree(hcdtw,k=5)#aglomeración en 5 clusters
proc.time()-t#finaliza el tiempo de ejecución
#grafican los clusters
clusplot(Tsonidos,groups,color=TRUE,shade=TRUE,main="Hierarchical con DTW agrupando en 5
clusters")
#gráfica el dendograma
plot(hcdtw,labels=datos$MuestraSonidos,cex=0.3,col=as.numeric(datos$MuestraSonidos),main="H
ierarchical con DTW agrupando en 5 cluster")
x <- rect.hclust(hcdtw,k=5,border="red")#partición del dendograma en cluster
table(datos$MuestraSonidos,groups)#tabla de clasificación del clustering jerarquico
table(groups)#tamaño de los clusters
x# composición de los clusters
sil <- silhouette(groups,distMatrix)#índice silueta
#gráfica la silueta
plot(sil,main="Silueta Hierarchical con DTW agrupando en 5 clusters", border="blue")
sil#valores del índice silueta

```

- **Jerárquico con 6 clusters**

```
#DTW
```

```

t<-proc.time()#inicia tiempo de ejecución
distMatrix <- dist(Tsonidos,method="DTW")#se implementa DTW para obtener una matriz de

```

```

#distancia de similaridad
hcdtw <- hclust(distMatrix,method="ward.D")#aplicación de algoritmo jerárquico
groups <- cutree(hcdtw,k=6)#aglomeración en 6 clusters
proc.time()-t#finaliza el tiempo de ejecución
#grafican los clusters
clusplot(Tsonidos,groups,color=TRUE,shade=TRUE,main="Hierarchical con DTW agrupando en 6 clusters")
#gráfica el dendograma
plot(hcdtw,labels=datos$MuestraSonidos,cex=0.3,col=as.numeric(datos$MuestraSonidos),main="Hierarchical con DTW agrupando en 6 cluster")
x <- rect.hclust(hcdtw,k=6,border="red")#partición del dendograma en cluster
table(datos$MuestraSonidos,groups)#tabla de clasificación del clustering jerarquico
table(groups)#tamaño de los clusters
x# composición de los clusters
sil <- silhouette(groups,distMatrix)#índice silueta
#gráfica la silueta
plot(sil,main="Silueta Hierarchical con DTW agrupando en 6 clusters", border="blue")
sil#valores del índice silueta

```

- **Jerárquico con 7 clusters**

```

#DTW

t<-proc.time()#inicia tiempo de ejecución
distMatrix <- dist(Tsonidos,method="DTW")#se implementa DTW para obtener una matriz de
#distancia de similaridad
hcdtw <- hclust(distMatrix,method="ward.D")#aplicación de algoritmo jerárquico
groups <- cutree(hcdtw,k=7)#aglomeración en 7 clusters
proc.time()-t#finaliza el tiempo de ejecución
#grafican los clusters
clusplot(Tsonidos,groups,color=TRUE,shade=TRUE,main="Hierarchical con DTW agrupando en 7 clusters")
#gráfica el dendograma
plot(hcdtw,labels=datos$MuestraSonidos,cex=0.3,col=as.numeric(datos$MuestraSonidos),main="Hierarchical con DTW agrupando en 7 cluster")
x <- rect.hclust(hcdtw,k=7,border="red")#partición del dendograma en cluster
table(datos$MuestraSonidos,groups)#tabla de clasificación del clustering jerarquico
table(groups)#tamaño de los clusters
x# composición de los clusters
sil <- silhouette(groups,distMatrix)#índice silueta
#gráfica la silueta
plot(sil,main="Silueta Hierarchical con DTW agrupando en 7 clusters", border="blue")
sil#valores del índice silueta

```

- **Jerárquico con 8 clusters**

```

#DTW

t<-proc.time()#inicia tiempo de ejecución
distMatrix <- dist(Tsonidos,method="DTW")#se implementa DTW para obtener una matriz de
#distancia de similaridad
hcdtw <- hclust(distMatrix,method="ward.D")#aplicación de algoritmo jerárquico

```

```

groups <- cutree(hcdtw,k=8)#aglomeración en 8 clusters
proc.time()-t#finaliza el tiempo de ejecución
#grafican los clusters
clusplot(Tsonidos,groups,color=TRUE,shade=TRUE,main="Hierarchical con DTW agrupando en 8 clusters")
#gráfica el dendograma
plot(hcdtw,labels=datos$MuestraSonidos,cex=0.3,col=as.numeric(datos$MuestraSonidos),main="Hierarchical con DTW agrupando en 8 cluster")
x <- rect.hclust(hcdtw,k=8,border="red")#partición del dendograma en cluster
table(datos$MuestraSonidos,groups)#tabla de clasificación del clustering jerarquico
table(groups)#tamaño de los clusters
x# composición de los clusters
sil <- silhouette(groups,distMatrix)#índice silueta
#gráfica la silueta
plot(sil,main="Silueta Hierarchical con DTW agrupando en 8 clusters", border="blue")
sil#valores del índice silueta

```

- **Jerárquico con 9 clusters**

```
#DTW
```

```

t<-proc.time()#inicia tiempo de ejecución
distMatrix <- dist(Tsonidos,method="DTW")#se implementa DTW para obtener una matriz de
#distancia de similaridad
hcdtw <- hclust(distMatrix,method="ward.D")#aplicación de algoritmo jerárquico
groups <- cutree(hcdtw,k=9)#aglomeración en 9 clusters
proc.time()-t#finaliza el tiempo de ejecución
#grafican los clusters
clusplot(Tsonidos,groups,color=TRUE,shade=TRUE,main="Hierarchical con DTW agrupando en 9 clusters")
#gráfica el dendograma
plot(hcdtw,labels=datos$MuestraSonidos,cex=0.3,col=as.numeric(datos$MuestraSonidos),main="Hierarchical con DTW agrupando en 9 cluster")
x <- rect.hclust(hcdtw,k=9,border="red")#partición del dendograma en cluster
table(datos$MuestraSonidos,groups)#tabla de clasificación del clustering jerarquico
table(groups)#tamaño de los clusters
x# composición de los clusters
sil <- silhouette(groups,distMatrix)#índice silueta
#gráfica la silueta
plot(sil,main="Silueta Hierarchical con DTW agrupando en 9 clusters", border="blue")
sil#valores del índice silueta

```

- **Jerárquico con 10 clusters**

```
#DTW
```

```

t<-proc.time()#inicia tiempo de ejecución
distMatrix <- dist(Tsonidos,method="DTW")#se implementa DTW para obtener una matriz de
#distancia de similaridad
hcdtw <- hclust(distMatrix,method="ward.D")#aplicación de algoritmo jerárquico
groups <- cutree(hcdtw,k=10)#aglomeración en 10 clusters
proc.time()-t#finaliza el tiempo de ejecución

```

```

#grafican los clusters
clusplot(Tsonidos,groups,color=TRUE,shade=TRUE,main="Hierarchical con DTW agrupando en 3 clusters")
#gráfica el dendograma
plot(hcdtw,labels=datos$MuestraSonidos,cex=0.3,col=as.numeric(datos$MuestraSonidos),main="Hierarchical con DTW agrupando en 10 cluster")
x <- rect.hclust(hcdtw,k=10,border="red")#partición del dendograma en cluster
table(datos$MuestraSonidos,groups)#tabla de clasificación del clustering jerárquico
table(groups)#tamaño de los clusters
x# composición de los clusters
sil <- silhouette(groups,distMatrix)#índice silueta
#gráfica la silueta
plot(sil,main="Silueta Hierarchical con DTW agrupando en 10 clusters", border="blue")
sil#valores del índice silueta

```

B.2 Clasificación con SVM del inglés Support Vector Machines

B.2.1 SVM con kernel lineal

```

# SVM LINEAL

t<-proc.time()#inicializa el tiempo de ejecución
#se obtiene el modelo aplicando la clasificación SVM
modelo <- svm(MuestraSonidos~.,data=Tsonido,method="C-classification",kernel="linear")
#se imprime sus valores
print(modelo)
summary(modelo)
pred <- fitted(modelo)
table(Tsonido$MuestraSonidos,pred)#matriz de confusión
proc.time()-t#finaliza tiempo de ejecución
error=sum(pred != Tsonidos$MuestraSonidos)/nrow(Tsonidos)
error#imprime el error
acierto=(1-error)*100#calcula la exactitud de clasificación del método svm
acierto#imprime la exactitud

```

B.2.2 SVM con kernel polynomial

```

# SVM POLINOMIAL

t<-proc.time()#inicializa el tiempo de ejecución
#se obtiene el modelo aplicando la clasificación SVM
modelo <- svm(MuestraSonidos~.,data=Tsonido,method="C-classification",kernel="polynomial")
#se imprime sus valores
print(modelo)
summary(modelo)
pred <- fitted(modelo)
table(Tsonido$MuestraSonidos,pred)#matriz de confusión
proc.time()-t#finaliza tiempo de ejecución
error=sum(pred != Tsonidos$MuestraSonidos)/nrow(Tsonidos)

```

```
error#imprime el error
acierto=(1-error)*100#calcula la exactitud de clasificación del método svm
acierto#imprime la exactitud
```

B.2.3 SVM con kernel radial

```
# SVM RADIAL

t<-proc.time()#inicializa el tiempo de ejecución
#se obtiene el modelo aplicando la clasificación SVM
modelo <- svm(MuestraSonidos~.,data=Tsonido,method="C-classification",kernel="radial")
#se imprime sus valores
print(modelo)
summary(modelo)
pred <- fitted(modelo)
table(Tsonido$MuestraSonidos,pred)#matriz de confusión
proc.time()-t#finaliza tiempo de ejecución
error=sum(pred != Tsonidos$MuestraSonidos)/nrow(Tsonidos)
error#imprime el error
acierto=(1-error)*100#calcula la exactitud de clasificación del método svm
acierto#imprime la exactitud
```

B.2.4 SVM con kernel sigmoidal

```
# SVM SIGMOIDAL

t<-proc.time()#inicializa el tiempo de ejecución
#se obtiene el modelo aplicando la clasificación SVM
modelo <- svm(MuestraSonidos~.,data=Tsonido,method="C-classification",kernel="sigmoid")
#se imprime sus valores
print(modelo)
summary(modelo)
pred <- fitted(modelo)
table(Tsonido$MuestraSonidos,pred)#matriz de confusión
proc.time()-t#finaliza tiempo de ejecución
error=sum(pred != Tsonidos$MuestraSonidos)/nrow(Tsonidos)
error#imprime el error
acierto=(1-error)*100#calcula la exactitud de clasificación del método svm
acierto#imprime la exactitud
```

B.3 Validación interna

```
#asignación de Tsonidos a Ts
Ts <- Tsonidos
```

```

Ts$MuestraSonidos <- NULL
#aplicación de validación Interna
Interna <- clValid(Ts,2:10,clMethods=c("hierarchical","kmeans","pam"), validation="internal")
#se imprimen los valores de la validación interna
summary(Interna)

```

B.4 Lectura de archivos

B.4.1 Lectura del archivo con 280 observaciones

```
#lectura del archivo
```

```

datos <- read.table("C:/Users/-
Administrador/Documents/disparos.txt",header=FALSE,sep="\t",dec=".")    colnames(dato) <-
c("ID","frecuencia1","frecuencia2", "frecuencia3","frecuencia4",
"frecuencia5","frecuencia6","frecuencia7","frecuencia8","frecuencia9","frecuencia10",
"frecuencia11","frecuencia12","frecuencia13","frecuencia14", "frecuencia15","frecuencia16",
"frecuencia17","frecuencia18", "frecuencia19","frecuencia20","frecuencia21","frecuencia22",
"frecuencia23","frecuencia24", "frecuencia25","frecuencia26", "frecuencia27","frecuencia28",
"frecuencia29","frecuencia30","frecuencia31","frecuencia32","frecuencia33","frecuencia34",
"frecuencia35","frecuencia36","frecuencia37","frecuencia38","frecuencia39","frecuencia40",
"frecuencia41","frecuencia42","frecuencia43","MuestraSonidos")

```

```
#almacena en Tsonidos dando el nombre de las columnas
```

```

Tsonidos<- datos[,c("frecuencia1","frecuencia2", "frecuencia3","frecuencia4","frecuencia5",
"frecuencia6","frecuencia7","frecuencia8","frecuencia9","frecuencia10","frecuencia11",
"frecuencia12","frecuencia13","frecuencia14", "frecuencia15","frecuencia16","frecuencia17",
"frecuencia18", "frecuencia19","frecuencia20","frecuencia21","frecuencia22","frecuencia23",
"frecuencia24","frecuencia25","frecuencia26","frecuencia27","frecuencia28","frecuencia29",
"frecuencia30","frecuencia31","frecuencia32", "frecuencia33","frecuencia34","frecuencia35",
"frecuencia36", "frecuencia37","frecuencia38","frecuencia39","frecuencia40","frecuencia41",
"frecuencia42","frecuencia43","MuestraSonidos")]

```

```
#asigna nombre de las filas a Tsonidos
```

```
rownames(Tsonidos) <- dato$ID
```

```
Tsonidos$MuestraSonidos <- NULL
```

B.4.1 Lectura del archivo con 301 observaciones

```

datos <- read.table("C:/Users/-
Administrador/Documents/disparos2.txt",header=FALSE,sep="\t",dec=".")

```

```

colnames(datos) <- c("ID","frecuencia1","frecuencia1","frecuencia3","frecuencia4",
"frecuencia5","frecuencia6","frecuencia7","frecuencia8",

```

"frecuencia9","frecuencia10","frecuencia11","frecuencia12",
"frecuencia13","frecuencia14","frecuencia15","frecuencia16",
"frecuencia17","frecuencia18","frecuencia19","frecuencia20",
"frecuencia21","frecuencia22","frecuencia23","frecuencia24",
"frecuencia25","frecuencia26","frecuencia27","frecuencia28",
"frecuencia29","frecuencia30","frecuencia31","frecuencia32",
"frecuencia33","frecuencia34","frecuencia35","frecuencia36",
"frecuencia37","frecuencia38","frecuencia39","frecuencia40",
"frecuencia41","frecuencia42","frecuencia43","frecuencia44",
"frecuencia45","frecuencia46","frecuencia47","frecuencia48",
"frecuencia49","frecuencia50","frecuencia51","frecuencia52",
"frecuencia53","frecuencia54","frecuencia55","frecuencia56",
"frecuencia57","frecuencia58","frecuencia59","frecuencia60",
"frecuencia61","frecuencia62","frecuencia63","frecuencia64",
"frecuencia65","frecuencia66","frecuencia67","frecuencia68",
"frecuencia69","frecuencia70","frecuencia71","frecuencia72",
"frecuencia73","frecuencia74","frecuencia75","frecuencia76",
"frecuencia77","frecuencia78","frecuencia79","frecuencia80",
"frecuencia81","frecuencia82","frecuencia83","frecuencia84",
"frecuencia85","frecuencia86","frecuencia87","frecuencia88",
"frecuencia89","frecuencia90","frecuencia91","frecuencia92",
"frecuencia93","frecuencia94","frecuencia95","frecuencia96",
"frecuencia97","frecuencia98","frecuencia99","frecuencia100",
"frecuencia101","frecuencia102","frecuencia103","frecuencia104",
"frecuencia105","frecuencia106","frecuencia107","frecuencia108",
"frecuencia109","frecuencia110","frecuencia111","frecuencia112",
"frecuencia113","frecuencia114","frecuencia115","frecuencia116",
"frecuencia117","frecuencia118","frecuencia119","frecuencia120",
"frecuencia121","frecuencia122","frecuencia123","frecuencia124",
"frecuencia125","frecuencia126","frecuencia127","frecuencia128",
"frecuencia129","frecuencia130","frecuencia131","frecuencia132",
"frecuencia133","frecuencia134","frecuencia135","frecuencia136",
"frecuencia137","frecuencia138","frecuencia139","frecuencia140",
"frecuencia141","frecuencia142","frecuencia143","frecuencia144",
"frecuencia145","frecuencia146","frecuencia147","MuestraSonidos")

Tsonidos <- datos[,c("frecuencia1","frecuencia1","frecuencia3","frecuencia4",
"frecuencia5","frecuencia6","frecuencia7","frecuencia8",
"frecuencia9","frecuencia10","frecuencia11","frecuencia12",
"frecuencia13","frecuencia14","frecuencia15","frecuencia16",
"frecuencia17","frecuencia18","frecuencia19","frecuencia20",
"frecuencia21","frecuencia22","frecuencia23","frecuencia24",
"frecuencia25","frecuencia26","frecuencia27","frecuencia28",
"frecuencia29","frecuencia30","frecuencia31","frecuencia32",
"frecuencia33","frecuencia34","frecuencia35","frecuencia36",
"frecuencia37","frecuencia38","frecuencia39","frecuencia40",
"frecuencia41","frecuencia42","frecuencia43","frecuencia44",
"frecuencia45","frecuencia46","frecuencia47","frecuencia48",
"frecuencia49","frecuencia50","frecuencia51","frecuencia52",
"frecuencia53","frecuencia54","frecuencia55","frecuencia56",
"frecuencia57","frecuencia58","frecuencia59","frecuencia60",
"frecuencia61","frecuencia62","frecuencia63","frecuencia64",
"frecuencia65","frecuencia66","frecuencia67","frecuencia68",
"frecuencia69","frecuencia70","frecuencia71","frecuencia72",
"frecuencia73","frecuencia74","frecuencia75","frecuencia76",

```
"frecuencia77","frecuencia78","frecuencia79","frecuencia80",  
"frecuencia81","frecuencia82","frecuencia83","frecuencia84",  
"frecuencia85","frecuencia86","frecuencia87","frecuencia88",  
"frecuencia89","frecuencia90","frecuencia91","frecuencia92",  
"frecuencia93","frecuencia94","frecuencia95","frecuencia96",  
"frecuencia97","frecuencia98","frecuencia99","frecuencia100",  
"frecuencia101","frecuencia102","frecuencia103","frecuencia104",  
"frecuencia105","frecuencia106","frecuencia107","frecuencia108",  
"frecuencia109","frecuencia110","frecuencia111","frecuencia112",  
"frecuencia113","frecuencia114","frecuencia115","frecuencia116",  
"frecuencia117","frecuencia118","frecuencia119","frecuencia120",  
"frecuencia121","frecuencia122","frecuencia123","frecuencia124",  
"frecuencia125","frecuencia126","frecuencia127","frecuencia128",  
"frecuencia129","frecuencia130","frecuencia131","frecuencia132",  
"frecuencia133","frecuencia134","frecuencia135","frecuencia136",  
"frecuencia137","frecuencia138","frecuencia139","frecuencia140",  
"frecuencia141","frecuencia142","frecuencia143","frecuencia144",  
"frecuencia145","frecuencia146","frecuencia147","MuestraSonidos")]
```

```
rownames(Tsonidos) <- datos$ID
```


Apéndice C

Resultados obtenidos por el método K-Means

C.1. Resultados obtenidos al implementar K-Means sobre las 280 observaciones por bandas de frecuencia promediadas en escala logarítmica

- **K-Means con 2 clusters**

La *Tabla C.1* exhibe el resultado obtenido al agrupar con 2 cluster sin importar que se iterara 10 veces se obtuvo el mismo resultado, en la *Figura C.1* se puede ver los cluster traslapados

Tabla C.1: K-Means con 2 clusters

1 Iteración k.means con 2 clusters de tamaño: 140, 140		
Muestras	Grupos	
	1	2
DisparoReal	62	75
DisparoSalva	23	21
EfectoSonido	37	12
Explosivo	18	32
Tiempo de ejecución 2,35 segundos		

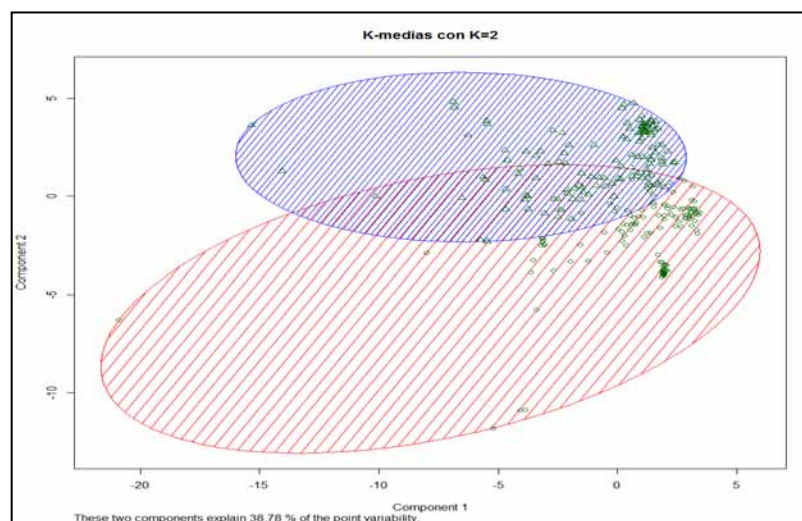


Figura C.1: K-Means con 2 clusters

- **K-Means con 3 clusters**

La *Tabla C.2* expone los resultados obtenidos por la iteración 1,3 y 7, también se puede ver que las tres iteraciones obtienen el mismo resultado, en la *Figura C.2* expone el resultado de la iteración 1 y se puede ver traslapación entre los *clusters*.

Tabla C.2: Iteración 1,3 y 7, Kmeans con 3 clusters

1 Iteración k.means con 3 clusters de tamaño: 57, 105, 118			
Muestras	Grupos		
	1	2	3
DisparoReal	34	47	56
DisparoSalva	0	26	18
EfectoSonido	15	1	33
Explosivo	8	31	11
Tiempo de ejecución 0.49 segundos			
3 Iteración k.means con 3 clusters de tamaño: 105, 118, 57			
Muestras	Grupos		
	1	2	3
DisparoReal	47	56	34
DisparoSalva	26	18	0
EfectoSonido	1	33	15
Explosivo	31	11	8
Tiempo de ejecución 0.55 segundos			
7 Iteración k.means con 3 clusters de tamaño: 105, 118, 57			
Muestras	Grupos		
	1	2	3
DisparoReal	47	56	34
DisparoSalva	26	18	0
EfectoSonido	1	33	15
Explosivo	31	11	8
Tiempo de ejecución 0.53 segundos			

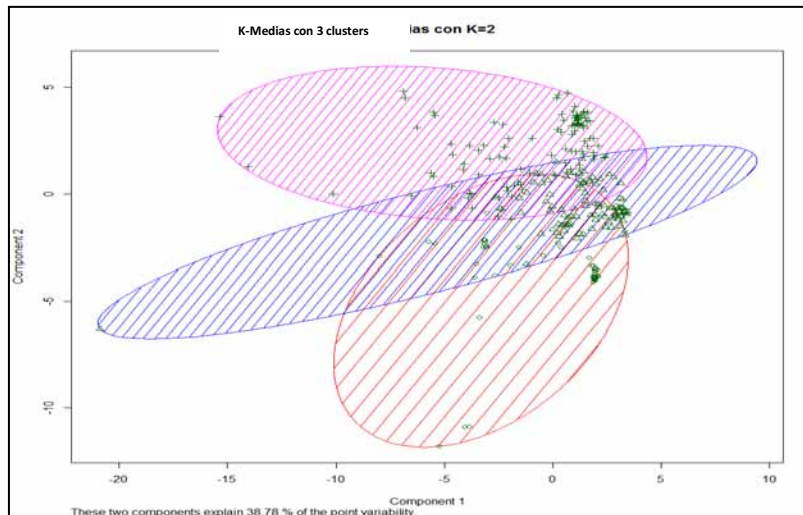


Figura C.2: Iteración 1, K-Means con 3 clusters

La *Tabla C.3* presenta similitud entre las iteraciones 2 y 6, también se puede ver la dificultad para distinguir entre las muestras

Tabla C.3 Iteración 1,3 y 7, Kmeans con 3 clusters

2 Iteración k.means con 3 clusters de tamaño: 98, 107, 75			
Muestras	Grupos		
	1	2	3
DisparoReal	21	69	47
DisparoSalva	20	17	7
EfectoSonido	27	3	19
Explosivo	30	18	2
Tiempo de ejecución 0.50 segundos			
6 Iteración k.means con 3 clusters de tamaño: 98, 107, 75			
Muestras	Grupos		
	1	2	3
DisparoReal	21	69	47
DisparoSalva	20	17	7
EfectoSonido	27	3	19
Explosivo	30	18	2
Tiempo de ejecución 0.48 segundos			

La *Tabla C.4* muestra que se obtuvieron los mismos resultados en la iteración 4,5,9 y 10, también se puede observar dificultad para distinguir entre las observaciones

Tabla C.4: Iteración 4,5,9 y 10, KMeans con 3 clusters

4 Iteración k.means con 3 clusters de tamaño: 107, 119, 54			
Muestras	Grupos		
	1	2	3
DisparoReal	50	56	31
DisparoSalva	25	19	0
EfectoSonido	1	33	15
Explosivo	31	11	8
Tiempo de ejecución 0.60 segundos			
5 Iteración k.means con 3 clusters de tamaño: 119,54, 107			
Muestras	Grupos		
	1	2	3
DisparoReal	56	31	50
DisparoSalva	19	0	25
EfectoSonido	33	15	1
Explosivo	11	8	31
Tiempo de ejecución 0.50 segundos			
9 Iteración k.means con 3 clusters de tamaño: 119, 54, 107			
Muestras	Grupos		
	1	2	3
DisparoReal	56	31	50
DisparoSalva	19	0	25
EfectoSonido	33	15	1
Explosivo	11	8	31
Tiempo de ejecución 0.57 segundos			
10 Iteración k.means con 3 clusters de tamaño: 119, 54, 107			
Muestras	Grupos		
	1	2	3
DisparoReal	56	31	50
DisparoSalva	19	0	25
EfectoSonido	33	15	1
Explosivo	11	8	31
Tiempo de ejecución 0.45 segundos			

- **K-Means con 4 clusters**

La *Tabla C.5* muestra los resultados obtenidos por las iteraciones 1,3,7 y 8, también puede verse que obtuvieron el mismo resultado y muestran que no hay una separación entre las observaciones

Tabla C.5: iteración 1,3,6,8 , Kmeans con 4 clusters

1 iteración con 4 clusters de tamaño: 73, 80,83,44				
Muestra	Grupos			
	1	2	3	4
DisparoReal	46	18	42	31
DisparoSalva	7	18	19	0
EfectoSonido	18	26	0	5
Explosivo	2	18	22	8
Tiempo de ejecución 0.60 segundos				
3 iteración con 4 clusters de tamaño: 83, 80, 73, 44				
Muestra	Grupos			
	1	2	3	4
DisparoReal	42	18	46	31
DisparoSalva	19	18	7	0
EfectoSonido	0	26	18	5
Explosivo	22	18	2	8
Tiempo de ejecución 0.64 segundos				
7 iteración con 4 clusters de tamaño: 44, 73, 80, 83				
Muestra	Grupos			
	1	2	3	4
DisparoReal	31	46	18	42
DisparoSalva	0	7	18	19
EfectoSonido	5	18	26	0
Explosivo	8	2	18	22
Tiempo de ejecución 3.58 segundos				
8 iteración con 4 clusters de tamaño: 44, 73, 83, 80				
Muestra	Grupos			
	1	2	3	4
DisparoReal	31	46	42	18
DisparoSalva	0	7	19	18
EfectoSonido	5	18	0	26
Explosivo	8	2	22	18
Tiempo de ejecución 3.76				

En la *Figura C.3* muestra los grupos formados por la iteración 7 y se puede ver que los grupos se superponen

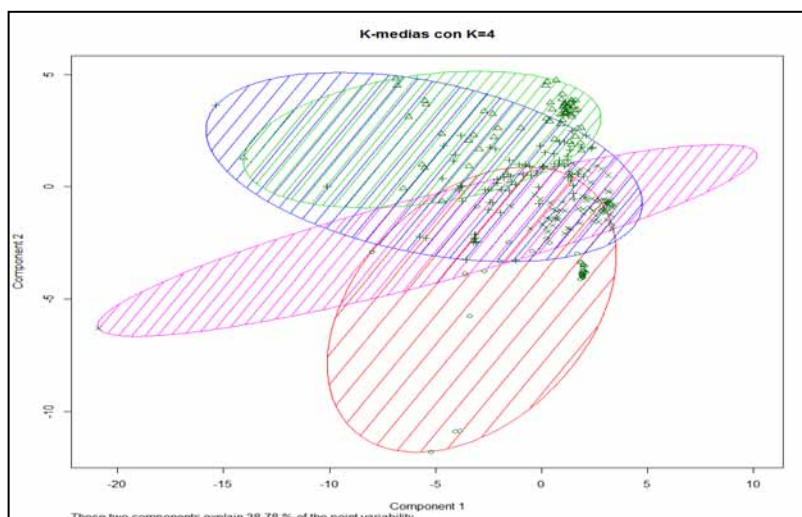


Figura C.3: Iteración 7, Kmeans con 4 cluster

La *Tabla C.6* muestra la iteración 2, y puede observarse que no se llegan a distinguir las observaciones

Tabla C.6: Iteración 2 kmeans con 4 clusters

2 iteración con 4 clusters de tamaño: 70, 30, 62, 118				
Muestra	Grupos			
	1	2	3	4
DisparoReal	22	29	17	69
DisparoSalva	19	1	6	18
EfectoSonido	5	0	32	12
Explosivo	24	0	7	19
Tiempo de ejecución 0.55 segundos				

La *Tabla C.7* muestra la iteración 9 y puede observarse que solo se llegan a separar 29 muestras aproximadamente 10% de los sonidos de disparo producidos por arma de fuego y la *Figura C.4* pueden observarse un grupo que llega a separarse de los demás grupos.

Tabla C.7: Iteración 9 con 4 clusters

9 iteración con 4 clusters de tamaño: 35, 107, 29, 109				
Muestra	Grupos			
	1	2	3	4
DisparoReal	2	56	29	50
DisparoSalva	1	18	0	25
EfectoSonido	23	25	0	1
Explosivo	9	8	0	33
Tiempo de ejecución 3.26 segundos				

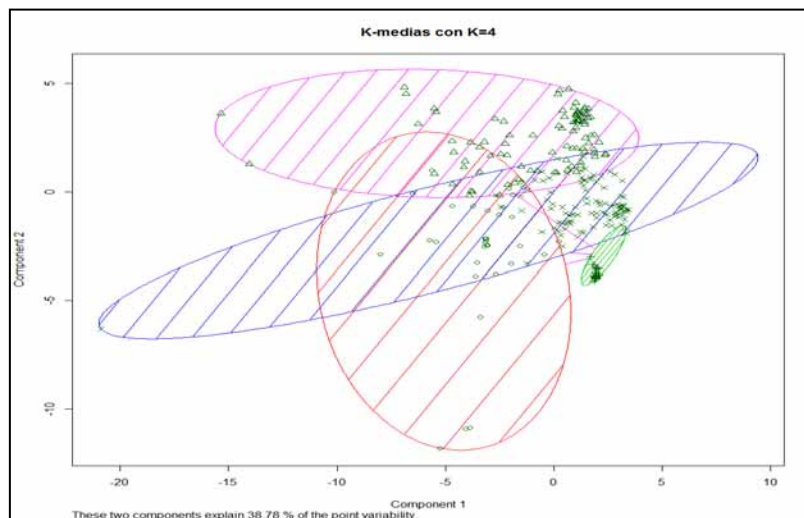


Figura C.4: iteración 9 con 4 clusters

La *Tabla C.8* nos muestra la iteración 4,6 y 10, también puede verse en estas iteraciones resultados similares

Tabla C.8: iteración 3,6 y 10, Kmeans con 4 cluters

4 iteración con 4 clusters de tamaño: 37, 75, 98, 70				
Muestra	Grupos			
	1	2	3	4
DisparoReal	3	46	66	22
DisparoSalva	2	7	17	18
EfectoSonido	23	21	0	5
Explosivo	9	1	15	25
Tiempo de ejecución 0.86 segundos				
6 iteración con 4 clusters de tamaño: 75, 37, 98, 70				
Muestra	Grupos			
	1	2	3	4
DisparoReal	46	3	66	22
DisparoSalva	7	2	17	18
EfectoSonido	21	23	0	5
Explosivo	1	9	15	25
Tiempo de ejecución 3.89 segundos				
10 iteración con 4 clusters de tamaño: 98, 37, 70, 75				
Muestra	Grupos			
	1	2	3	4
DisparoReal	66	3	22	46
DisparoSalva	17	2	18	7
EfectoSonido	0	23	5	21
Explosivo	15	9	25	1
Tiempo de ejecución 3.65 segundos				

- **K-Means con 5 clusters**

En la *Tabla C.9* se pueden ver las iteraciones con los mismos resultados, de igual forma puede verse que estos resultados muestran dificultad para separar las observaciones, en la *Figura C.5* se exponen los resultados obtenidos por la iteración 1 y se observa muy poca separación entre los *clusters*.

Tabla C.9; Iteración 1, 5 y 7, Kmeans con 5 clusters

1 Iteración k.means con 5 clusters de tamaño: 31, 71, 37, 82, 59					
Muestras	Grupos				
	1	2	3	4	5
DisparoReal	29	46	2	42	18
DisparoSalva	0	7	1	18	18
EfectoSonido	0	17	25	0	7
Explosivo	2	1	9	22	16
Tiempo de ejecución 4.02 segundos					
6 Iteración k.means con 5 clusters de tamaño: 71, 31, 59, 82, 37					
Muestras	Grupos				
	1	2	3	4	5
DisparoReal	46	20	18	42	2
DisparoSalva	7	0	18	18	1
EfectoSonido	17	0	7	0	25
Explosivo	1	2	16	22	9
Tiempo de ejecución 4.06 segundos					
7 Iteración k.means con 5 clusters de tamaño: 71, 82, 59, 37, 31					
Muestras	Grupos				
	1	2	3	4	5
DisparoReal	46	42	18	2	29
DisparoSalva	7	18	18	1	0
EfectoSonido	17	0	7	25	0
Explosivo	1	22	16	9	2
Tiempo de ejecución 4.53 segundos					

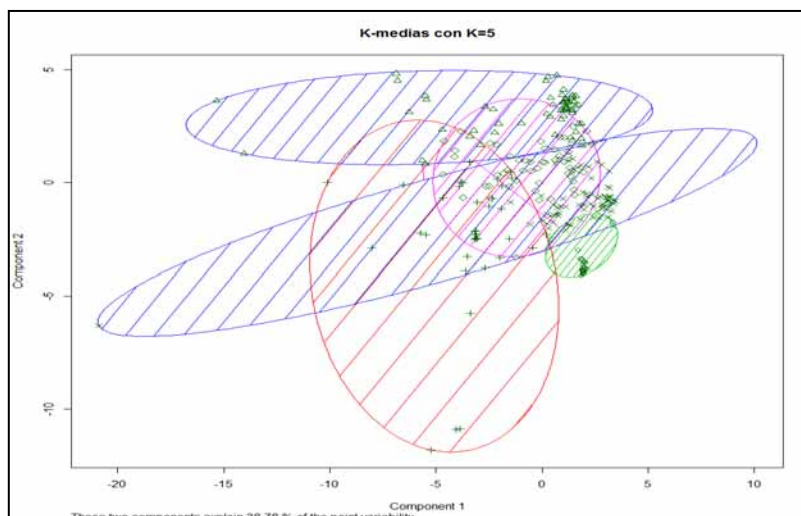


Figura C.5: Iteración 1, Kmeans con 5 clusters

La *Tabla C.10* muestra los resultados obtenidos por la iteración 2 y 3, en los cuáles puede observarse que no se logran separar las observaciones entre sí, sin embargo en la *Figura C.6* exhibe que la iteración 3 logra formar un grupo compuesto por 31 observaciones, de las cuáles 29 son sonidos de disparos reales, pero mostrando traslapamiento entre los *clusters*.

Tabla C.10: Iteraciones 2 y 3. Kmeans con 5 clusters

2 Iteración k.means con 5 clusters de tamaño: 96, 70, 45, 37, 32					
Muestras	Grupos				
	1	2	3	4	5
DisparoReal	64	22	22	3	26
DisparoSalva	17	18	6	1	2
EfectoSonido	0	5	16	24	4
Explosivo	15	25	1	9	0
Tiempo de ejecución 3.59 segundos					
3 Iteración k.means con 5 clusters de tamaño: 31, 73, 59, 35, 82					
Muestras	Grupos				
	1	2	3	4	5
DisparoReal	29	46	18	2	42
DisparoSalva	0	7	18	1	18
EfectoSonido	0	19	7	23	0
Explosivo	2	1	16	9	22
Tiempo de ejecución 4.42 segundos					

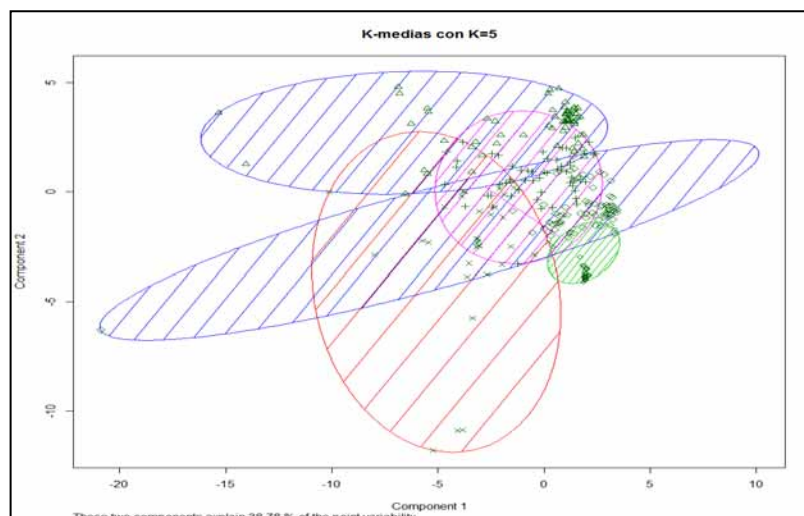


Figura C.6: Iteración 3. Kmeans con 5 clusters

La *Tabla C.11* muestra los resultados obtenidos por las iteraciones 4 y 10.

Tabla C.11: Iteración 4 y 10 , Kmeans con 5 clusters

4 Iteración k.means con 5 clusters de tamaño: 42, 78, 43, 81, 36					
Muestras	Grupos				
	1	2	3	4	5
DisparoReal	22	16	31	42	26
DisparoSalva	5	19	0	18	2
EfectoSonido	14	24	4	0	7
Explosivo	1	19	8	21	1
Tiempo de ejecución 3.73 segundos					
10 Iteración k.means con 5 clusters de tamaño: 36, 42, 78, 81, 43					
Muestras	Grupos				
	1	2	3	4	5
DisparoReal	26	22	16	42	31
DisparoSalva	2	5	19	18	0
EfectoSonido	7	14	24	0	4
Explosivo	1	1	19	21	8
Tiempo de ejecución 3.87 segundos					

La *Figura C.7* a exhibe los clústeres formados por la iteración 4, en la cual se puede observar que existe traslape entre los cluster

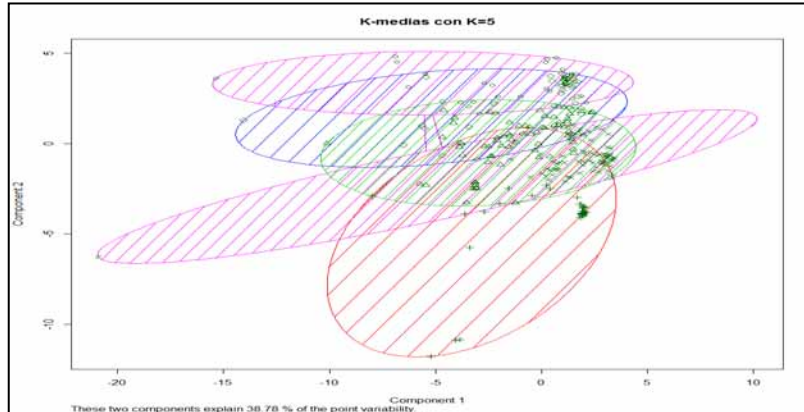


Figura C.7 iteración 4 con 5 clusters

La *Tabla C.12* muestra las iteraciones que obtuvieron resultados similares

Tabla C.12: Iteración 5,8 y 9, Kmeans con 5 clusters

5 Iteración k.means con 5 clusters de tamaño: 34, 82, 69, 37, 58					
Muestras	Grupos				
	1	2	3	4	5
DisparoReal	0	42	46	31	18
DisparoSalva	1	18	7	0	18
EfectoSonido	25	0	15	3	6
Explosivo	8	22	1	3	16
Tiempo de ejecución 3.57 segundos					
8 Iteración k.means con 5 clusters de tamaño: 58, 34, 82, 69, 37					
Muestras	Grupos				
	1	2	3	4	5
DisparoReal	18	0	42	46	31
DisparoSalva	18	1	18	7	0
EfectoSonido	6	25	0	15	3
Explosivo	16	8	22	1	3
Tiempo de ejecución 3.73 segundos					
9 Iteración k.means con 5 clusters de tamaño: 30, 64, 82, 31, 73					
Muestras	Grupos				
	1	2	3	4	5
DisparoReal	2	18	42	29	46
DisparoSalva	0	19	18	0	7
EfectoSonido	20	10	0	0	19
Explosivo	8	17	22	2	1
Tiempo de ejecución 5.28 segundos					

La Figura C.8 expone los resultados obtenidos por la iteración 9 en los cuáles puede verse poca separación entre los clusters

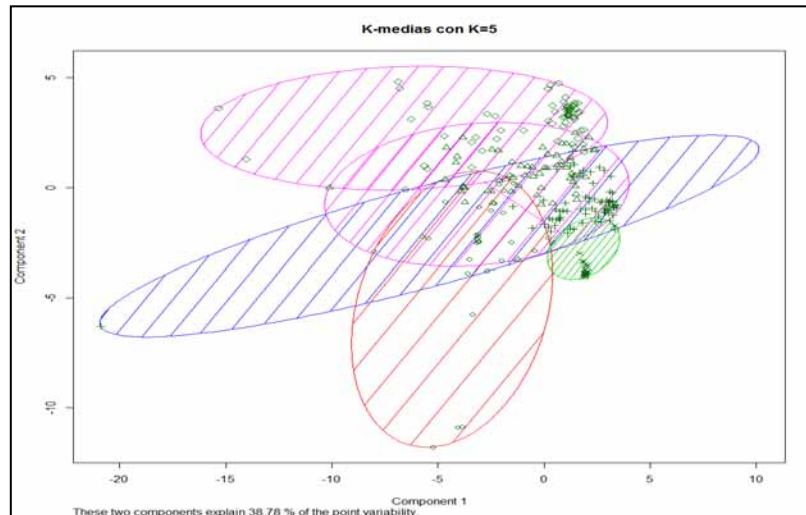


Figura C.8: Iteración 9, Kmeans con 5 clusters

- **K-Means con 6 clusters**

Al realizar el cálculo con 6 *cluster*, la primera, quinta, sexta y novena iteración fueron las únicas que lograron separar entre 28 y 29 sonidos de disparos, aproximadamente el 10% solo se consideraron como disparos reales, las demás iteraciones no lograron distinguir entre las muestras. Estos resultados pueden ser vistos en la *Tabla C.13*.

La Figura C.9 expone el resultado de la primera iteración con 6 *clusters*

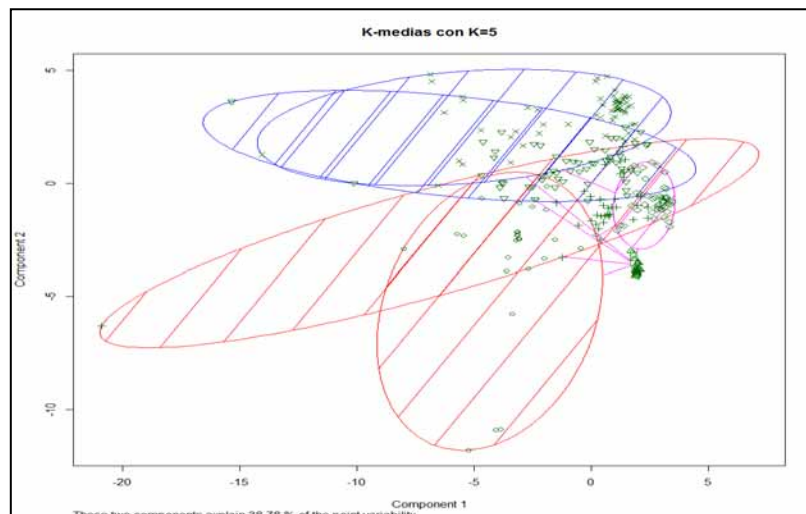


Figura C.9: Iteracion1 con 6 clusters

Tabla C.13: Iteraciones 1,2,3 y 4. Kmeans con 6 clusters

1 Iteración k.means con 6 clusters de tamaño: 28, 28, 37, 72, 53, 62						
Muestras	Grupos					
	1	2	3	4	5	6
DisparoReal	2	28	21	46	24	16
DisparoSalva	0	0	2	7	17	18
EfectoSonido	18	0	0	18	0	13
Explosivo	8	0	14	1	12	15
Tiempo de ejecución 0.72 segundos						
2 Iteración k.means con 6 clusters de tamaño: 82, 31, 57, 36, 32, 42						
Muestras	Grupos					
	1	2	3	4	5	6
DisparoReal	42	29	18	2	26	20
DisparoSalva	18	0	18	1	2	5
EfectoSonido	0	0	5	24	4	16
Explosivo	22	2	16	9	0	1
Tiempo de ejecución 0.94 segundos						
3 Iteración k.means con 6 clusters de tamaño: 57, 61, 25, 24, 84, 29						
Muestras	Grupos					
	1	2	3	4	5	6
DisparoReal	18	46	0	2	42	29
DisparoSalva	18	5	3	0	18	0
EfectoSonido	5	9	20	15	0	0
Explosivo	16	1	2	7	24	0
Tiempo de ejecución 0.69 segundos						
4 Iteración k.means con 6 clusters de tamaño: 62, 24, 33, 69, 58, 34						
Muestras	Grupos					
	1	2	3	4	5	6
DisparoReal	26	0	29	46	18	18
DisparoSalva	17	1	0	7	17	2
EfectoSonido	0	17	3	15	6	8
Explosivo	19	6	1	1	17	6
Tiempo de ejecución 0.71 segundos						

Tabla C.14: Iteraciones 5,6,7 y 8. Kmeans con 6 clusters

5 Iteración k.means con 6 clusters de tamaño: 27, 28, 60, 73, 35, 57						
Muestras	Grupos					
	1	2	3	4	5	6
DisparoReal	17	28	26	46	2	18
DisparoSalva	2	0	17	7	1	17
EfectoSonido	0	0	0	19	23	7
Explosivo	8	0	17	1	9	15
Tiempo de ejecución 0.68 segundos						
6 Iteración k.means con 6 clusters de tamaño: 30, 73, 63, 30, 28, 56						
Muestras	Grupos					
	1	2	3	4	5	6
DisparoReal	2	46	18	21	28	22
DisparoSalva	0	7	18	2	0	17
EfectoSonido	20	19	10	0	0	0
Explosivo	8	1	17	7	0	17
Tiempo de ejecución 0.70 segundos						
7 Iteración k.means con 6 clusters de tamaño: 34, 37, 29, 53, 58, 69						
Muestras	Grupos					
	1	2	3	4	5	6
DisparoReal	0	31	21	21	18	46
DisparoSalva	1	0	1	1	18	7
EfectoSonido	25	3	0	0	6	15
Explosivo	8	3	7	15	16	1
Tiempo de ejecución 0.71 segundos						
8 Iteración k.means con 6 clusters de tamaño: 68, 35, 9, 27, 98, 43						
Muestras	Grupos					
	1	2	3	4	5	6
DisparoReal	21	3	0	26	66	21
DisparoSalva	17	2	1	1	17	6
EfectoSonido	5	23	6	0	0	15
Explosivo	25	7	2	0	15	1
Tiempo de ejecución 0.75 segundos						

La Figura C.10 expone el resultado gráfico de la iteración 5

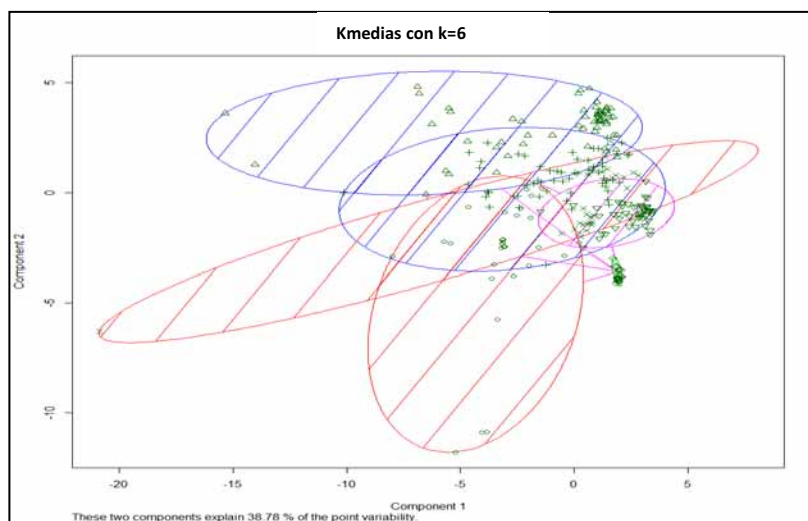


Figura C.10: Iteración 5, Kmeans con 6 clusters

Tabla C.15: Iteraciones 9 y 10. K-Means con 6 clusters

9 Iteración k.means con 6 clusters de tamaño: 84, 29, 32, 36, 57, 42						
Muestras	Grupos					
	1	2	3	4	5	6
DisparoReal	42	29	26	2	18	20
DisparoSalva	18	0	2	1	18	5
EfectoSonido	0	0	4	24	5	16
Explosivo	24	0	0	9	16	1
Tiempo de ejecución 0.67 segundos						
10 Iteración k.means con 6 clusters de tamaño: 31, 36, 57, 32, 42, 82						
Muestras	Grupos					
	1	2	3	4	5	6
DisparoReal	29	2	18	26	20	42
DisparoSalva	0	1	18	2	5	18
EfectoSonido	0	24	5	4	16	0
Explosivo	2	9	16	0	1	22
Tiempo de ejecución 0.67 segundos						

- **K-Means con 7 clusters**

La *Tabla C.16* expone las iteraciones que solo lograron distinguir el 10% de los sonidos de disparos reales. Gráficamente se puede ver la iteración 4 en la *Figura C.11*.

Tabla C.16: Iteraciones 3 y 4. Kmeans con 7 clusters

3 Iteración k.means con 7 clusters de tamaño: 29, 56, 41, 24, 27, 76, 27							
Muestras	Grupos						
	1	2	3	4	5	6	7
DisparoReal	29	18	20	0	7	37	25
DisparoSalva	0	18	5	2	1	17	1
EfectoSonido	0	5	15	17	12	0	0
Explosivo	0	15	1	5	7	22	0
Tiempo de ejecución 0.69 segundos							
4 Iteración k.means con 7 clusters de tamaño: 26, 80, 39, 37, 32, 29, 37							
Muestras	Grupos						
	1	2	3	4	5	6	7
DisparoReal	2	40	20	8	26	29	12
DisparoSalva	0	18	5	8	2	0	11
EfectoSonido	16	0	13	13	4	0	3
Explosivo	8	22	1	8	0	0	11
Tiempo de ejecución 0.74 segundos							

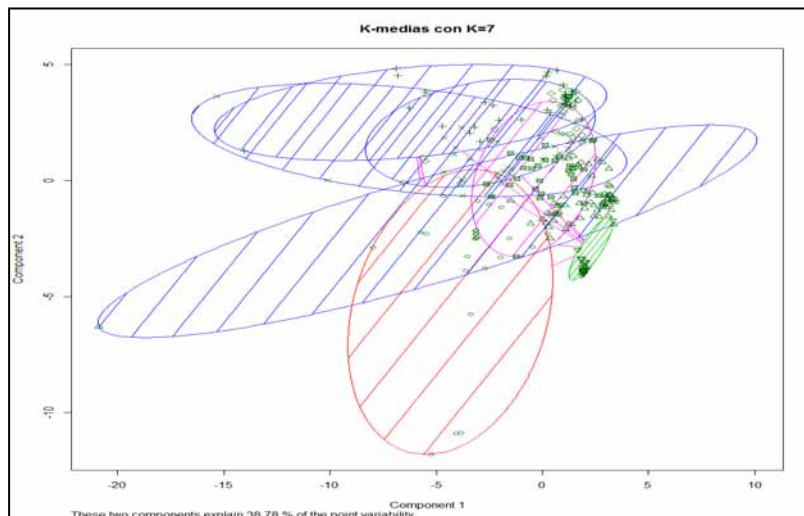


Figura C.11: Iteración 4. con 7 clusters

La *Tabla C.17* y la *Tabla C.18* exponen los resultados obtenidos por las demás iteraciones

Tabla C.17: Iteraciones 1, 2, 5 y 6. Kmeans con 7 clusters

1 Iteración k.means con 7 clusters de tamaño: 57, 28, 69, 13, 31, 29, 53							
Muestras	Grupos						
	1	2	3	4	5	6	7
DisparoReal	18	2	46	0	29	21	21
DisparoSalva	18	1	7	0	0	1	17
EfectoSonido	7	18	15	9	0	0	0
Explosivo	14	7	1	4	2	7	15
Tiempo de ejecución 0.70 segundos							
2 Iteración k.means con 7 clusters de tamaño: 27, 30, 56, 33, 30, 35, 69							
Muestras	Grupos						
	1	2	3	4	5	6	7
DisparoReal	0	7	18	28	19	19	46
DisparoSalva	1	16	17	0	2	1	7
EfectoSonido	20	0	6	3	5	0	15
Explosivo	6	7	15	2	4	15	1
Tiempo de ejecución 1.03 segundos							
5 Iteración k.means con 7 clusters de tamaño: 32, 42, 31, 57, 53, 29, 36							
Muestras	Grupos						
	1	2	3	4	5	6	7
DisparoReal	26	20	29	18	21	21	2
DisparoSalva	2	5	0	18	17	1	1
EfectoSonido	4	16	0	5	0	0	24
Explosivo	0	1	2	16	15	7	9
Tiempo de ejecución 0.89 segundos							
6 Iteración k.means con 7 clusters de tamaño: 31, 29, 53, 32, 42, 57, 36							
Muestras	Grupos						
	1	2	3	4	5	6	7
DisparoReal	29	21	21	26	20	18	2
DisparoSalva	0	1	17	2	5	18	1
EfectoSonido	0	0	0	4	16	5	24
Explosivo	2	7	15	0	1	16	9
Tiempo de ejecución 0.73 segundos							

Tabla C.18: Iteraciones 7,8,9 y 10. Kmeans con 7 clusters

7 Iteración k.means con 7 clusters de tamaño: 20, 27, 82, 57, 40, 31, 23							
Muestras	Grupos						
	1	2	3	4	5	6	7
DisparoReal	0	26	42	18	20	29	2
DisparoSalva	2	1	18	18	5	0	0
EfectoSonido	15	0	0	5	14	0	15
Explosivo	3	0	22	16	1	2	6
Tiempo de ejecución 0.96 segundos							
8 Iteración k.means con 7 clusters de tamaño: 32, 36, 42, 57, 53, 29, 31							
Muestras	Grupos						
	1	2	3	4	5	6	7
DisparoReal	26	2	20	18	21	21	29
DisparoSalva	2	1	5	18	17	1	0
EfectoSonido	4	24	16	5	0	0	0
Explosivo	0	9	1	16	15	7	2
Tiempo de ejecución 0.66 segundos							
9 Iteración k.means con 7 clusters de tamaño: 59, 25, 9, 82, 23, 51, 31							
Muestras	Grupos						
	1	2	3	4	5	6	7
DisparoReal	45	2	1	42	0	18	20
DisparoSalva	5	0	1	18	5	15	0
EfectoSonido	8	16	6	0	15	4	0
Explosivo	1	7	7	22	3	14	2
Tiempo de ejecución 1.17 segundos							
10 Iteración k.means con 7 clusters de tamaño: 28, 53, 27, 53, 57, 20, 42							
Muestras	Grupos						
	1	2	3	4	5	6	7
DisparoReal	0	25	20	20	21	30	21
DisparoSalva	0	2	6	15	4	0	17
EfectoSonido	24	0	17	5	0	3	0
Explosivo	7	0	1	15	8	4	15
Tiempo de ejecución 0.78 segundos							

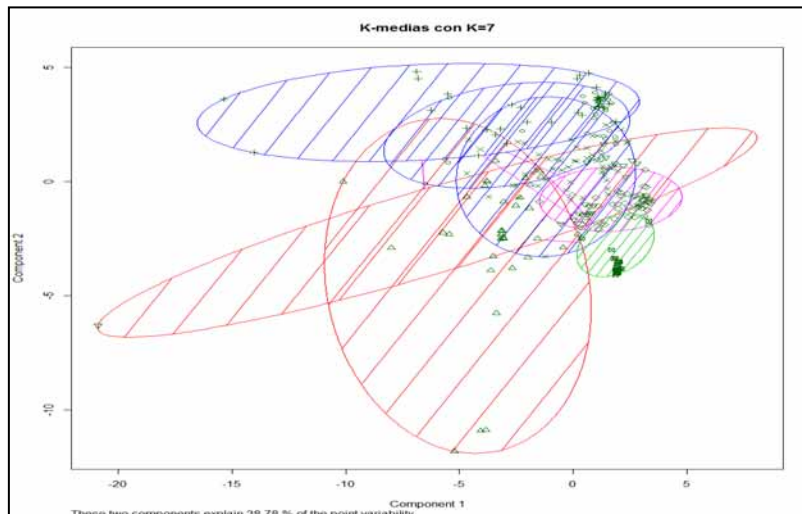


Figura C.12: Iteración 8. Kmeans con 7 clusters

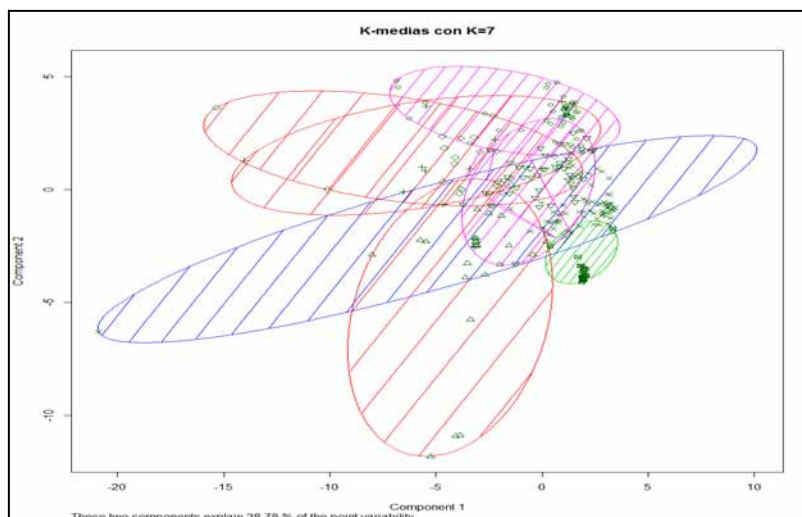


Figura C.13: Iteración 9. Kmeans con 7 clusters

- **K-Means con 8 clusters**

La *Tabla C.19* expone las iteraciones que distribuyeron aproximadamente el 10% de los sonidos de disparos reales

Tabla C.19: Iteraciones 1,3,6 y 8. Kmeans con 8 clusters

1 Iteración k.means con 8 clusters de tamaño: 57, 9, 29, 56, 41, 33, 27, 28								
Muestras	Grupos							
	1	2	3	4	5	6	7	8
DisparoReal	18	0	21	22	20	2	26	28
DisparoSalva	18	1	1	17	5	1	1	0
EfectoSonido	5	6	0	0	15	23	0	0
Explosivo	16	2	7	17	1	7	0	0
Tiempo de ejecución 6.30 segundos								
3 Iteración k.means con 8 clusters de tamaño: 30, 55, 42, 37, 28, 20, 32, 36								
Muestras	Grupos							
	1	2	3	4	5	6	7	8
DisparoReal	7	18	20	19	28	17	26	2
DisparoSalva	16	17	5	1	0	2	2	1
EfectoSonido	0	5	16	0	0	0	4	24
Explosivo	7	15	1	17	0	1	0	9
6 Iteración k.means con 8 clusters de tamaño: 9, 23, 30, 28, 44, 24, 56, 66								
Muestras	Grupos							
	1	2	3	4	5	6	7	8
DisparoReal	0	0	18	28	12	15	18	46
DisparoSalva	0	2	2	0	17	0	17	6
EfectoSonido	4	18	8	0	0	0	6	13
Explosivo	5	3	2	0	15	9	15	1
Tiempo de ejecución 1.11 segundos								
8 Iteración k.means con 8 clusters de tamaño: 28, 29, 23, 19, 6, 64, 55, 56								
Muestras	Grupos							
	1	2	3	4	5	6	7	8
DisparoReal	28	21	0	2	0	46	18	22
DisparoSalva	0	1	2	0	0	6	18	17
EfectoSonido	0	0	18	12	3	11	5	0
Explosivo	0	7	3	5	3	1	14	17
Tiempo de ejecución 0.67 segundos								

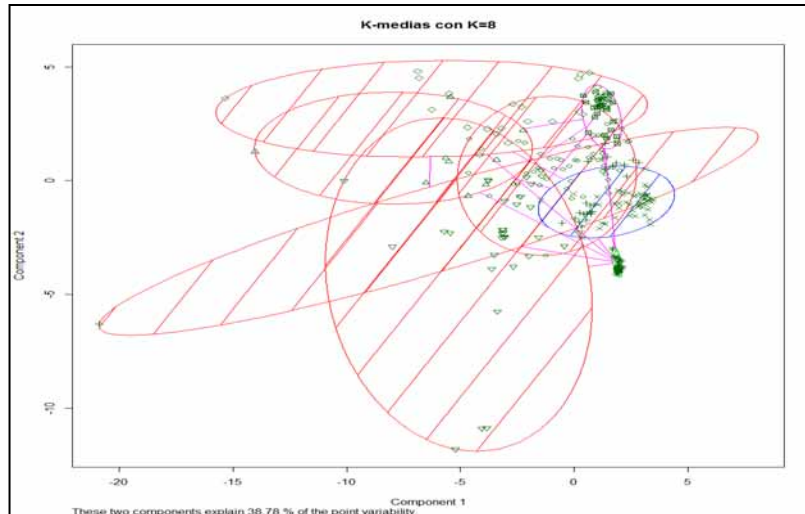


Figura C.14: Iteración 1. KMeans con 8 clusters

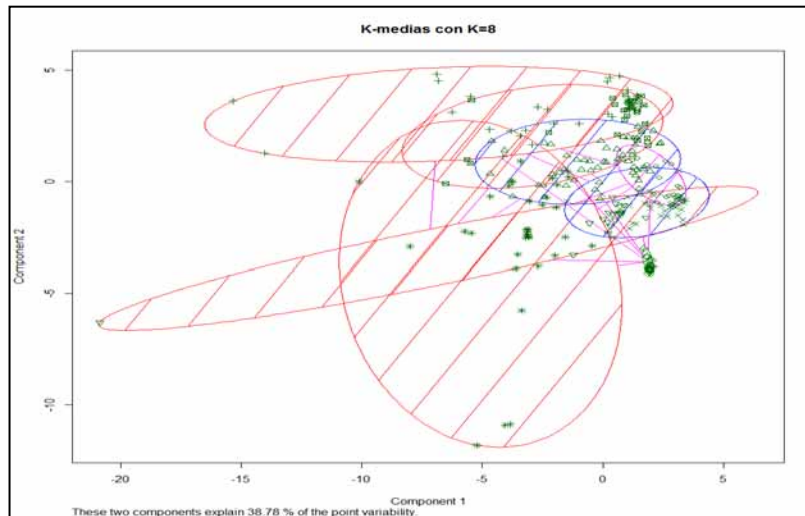


Figura C.15: Iteración 3. Kmeans con 8 clusters

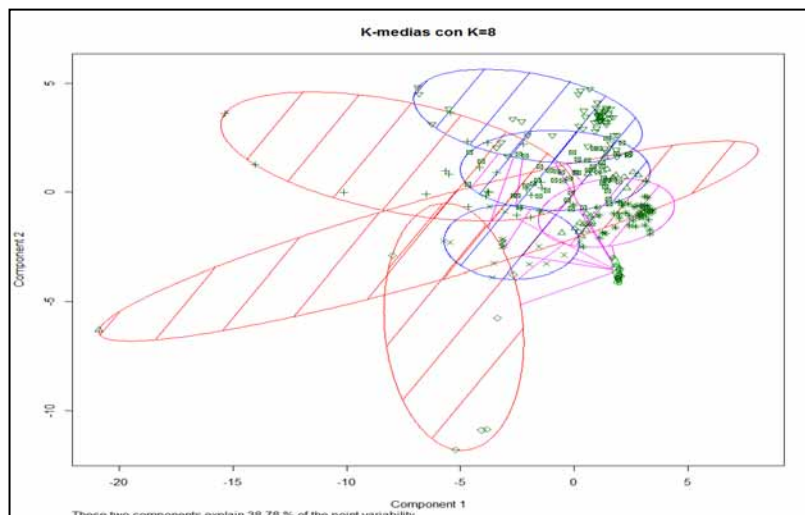


Figura C.16: Iteración 8. Kmeans con 8 clusters

La *Tabla C.20* y la *Tabla C.21* exponen los resultados de las iteraciones que no presentaron separación entre las observaciones.

Tabla C.20: Iteraciones 2, 4,,5 y 7. Kmeans con 8clusters

2 Iteración k.means con 8 clusters de tamaño: 39, 20, 27, 23, 31, 36, 27, 77								
Muestras	Grupos							
	1	2	3	4	5	6	7	8
DisparoReal	20	0	26	2	29	10	11	39
DisparoSalva	5	2	1	0	0	12	6	18
EfectoSonido	13	15	0	16	0	5	1	0
Explosivo	1	3	0	6	2	9	9	20
Tiempo de ejecución 4.96 segundos								
4 Iteración k.means con 8 clusters de tamaño: 40, 29, 59, 32, 23, 26, 38, 33								
Muestras	Grupos							
	1	2	3	4	5	6	7	8
DisparoReal	11	26	25	0	18	9	19	29
DisparoSalva	12	2	17	1	1	6	5	0
EfectoSonido	3	1	0	23	1	5	13	3
Explosivo	14	0	17	8	3	6	1	1
Tiempo de ejecución 0.79 segundos								
5 Iteración k.means con 8 clusters de tamaño: 53, 57, 27, 31, 23, 40, 20, 29								
Muestras	Grupos							
	1	2	3	4	5	6	7	8
DisparoReal	21	18	26	29	2	20	0	21
DisparoSalva	17	18	1	0	0	5	2	1
EfectoSonido	0	5	0	0	15	14	15	0
Explosivo	15	16	0	2	6	1	3	7
Tiempo de ejecución 0.73 segundos								
7 Iteración k.means con 8 clusters de tamaño: 29, 53, 40, 27, 23, 31, 57, 20								
Muestras	Grupos							
	1	2	3	4	5	6	7	8
DisparoReal	21	21	20	26	2	29	18	0
DisparoSalva	1	17	5	1	0	0	18	2
EfectoSonido	0	0	14	0	15	0	5	15
Explosivo	7	15	1	0	6	2	16	3
Tiempo de ejecución 0.73 segundos								

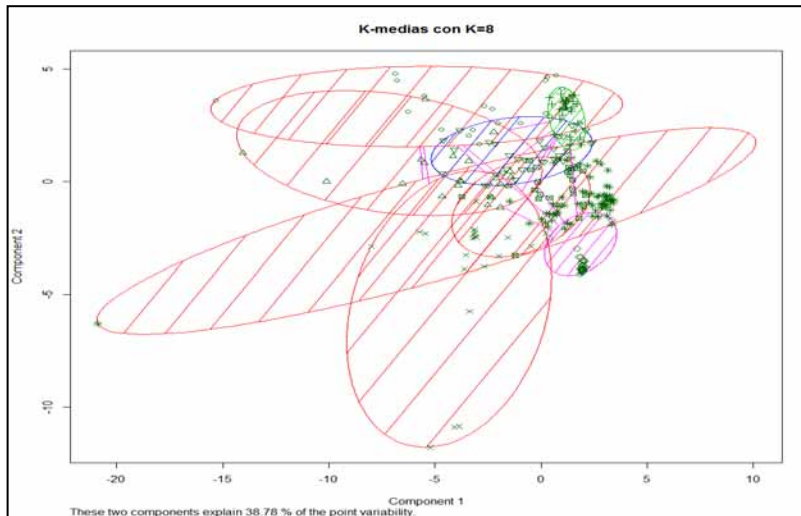


Figura C.17: Iteración 2. Kmeans con 8 clusters

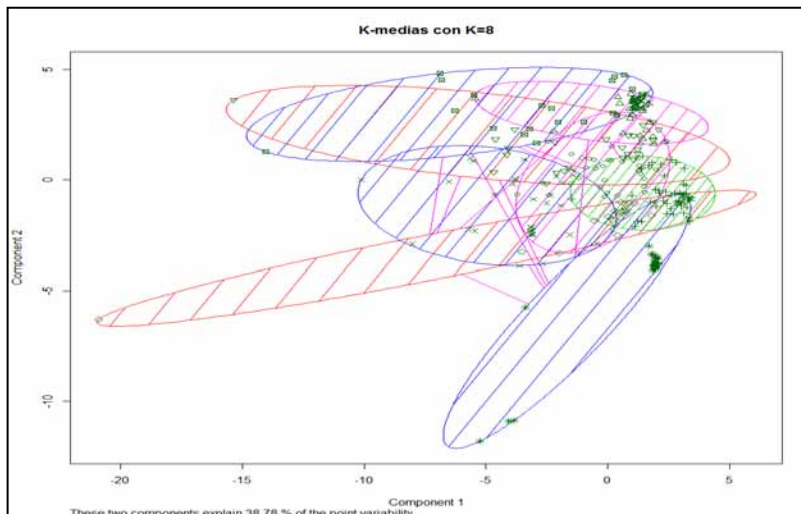


Figura C.18: K-Means con 8 clusters. Iteración 4

Tabla C.21: Iteraciones 9 y 10. Kmeans con 8 clusters

9 Iteración k.means con 8 clusters de tamaño: 40, 23, 27, 20, 29, 53, 31, 57								
Muestras	Grupos							
	1	2	3	4	5	6	7	8
DisparoReal	20	2	26	0	21	21	29	18
DisparoSalva	5	0	1	2	1	17	0	18
EfectoSonido	14	15	0	15	0	0	0	5
Explosivo	1	6	0	3	7	15	2	16
Tiempo de ejecución 0.85 segundos								
10 Iteración k.means con 8 clusters de tamaño: 32, 23, 55, 30, 29, 42, 32, 37								
Muestras	Grupos							
	1	2	3	4	5	6	7	8
DisparoReal	0	19	18	7	26	20	28	19
DisparoSalva	1	2	17	16	2	5	0	19
EfectoSonido	23	1	5	0	1	16	3	0
Explosivo	8	1	15	7	0	1	1	17
Tiempo de ejecución 0.89 segundos								

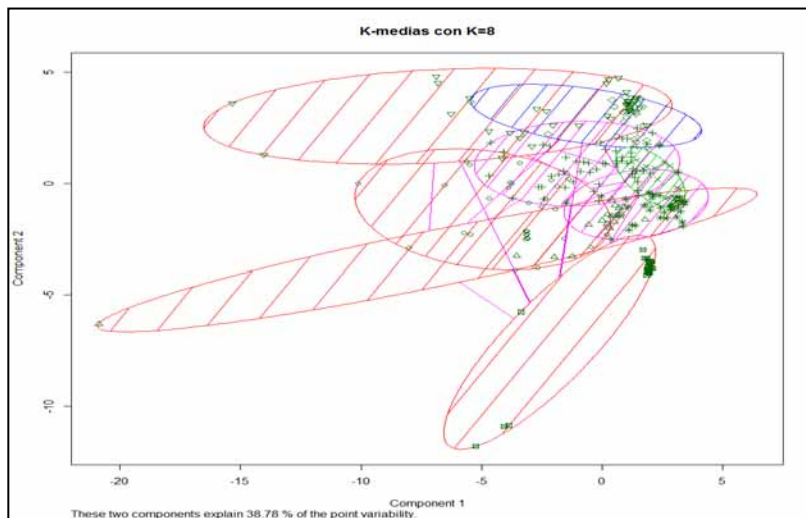


Figura C.19: Iteración 10, Kmeans con 8 clusters

- **K-Means con 9 clusters**

La *Tabla C.22* muestra las iteraciones que reconocieron aproximadamente entre el 7 y 10% de los sonidos de disparos reales

Tabla C.22: Iteraciones 1,5,7,8 y 9. Kmeans con 9 clusters

1 Iteración k.means con 9 clusters de tamaño: 12, 57, 20, 27, 31, 9, 21, 81, 22									
Muestras	Grupos								
	1	2	3	4	5	6	7	8	9
DisparoReal	0	18	20	26	29	20	18	0	28
DisparoSalva	1	18	0	1	0	4	13	1	0
EfectoSonido	9	5	0	0	0	8	4	7	0
Explosivo	2	16	0	0	2	1	12	1	2
Tiempo de ejecución 0.73 segundos									
5 Iteración k.means con 9 clusters de tamaño: 26, 57, 15, 40, 40, 27, 37, 9, 29									
Muestras	Grupos								
	1	2	3	4	5	6	7	8	9
DisparoReal	2	44	11	9	11	27	11	1	21
DisparoSalva	0	4	0	9	17	0	12	1	1
EfectoSonido	17	8	0	15	0	0	3	6	0
Explosivo	7	1	4	7	12	0	11	1	7
7 Iteración k.means con 7 clusters de tamaño: 38, 23, 60, 8, 27, 55, 14, 27, 28									
Muestras	Grupos								
	1	2	3	4	5	6	7	8	9
DisparoReal	20	2	26	0	17	18	0	26	28
DisparoSalva	5	0	17	1	2	17	1	1	0
EfectoSonido	12	15	0	6	0	5	11	0	0
Explosivo	1	6	17	1	8	15	2	0	0
Tiempo de ejecución 0.97 segundos									
8 Iteración k.means con 9 clusters de tamaño: 54, 27, 38, 35, 36, 29, 9, 24, 28									
Muestras	Grupos								
	1	2	3	4	5	6	7	8	9
DisparoReal	21	26	20	8	11	21	0	2	28
DisparoSalva	17	1	5	7	12	1	1	0	0
EfectoSonido	0	0	12	12	3	0	7	15	0
Explosivo	15	0	1	8	10	7	1	7	0
Tiempo de ejecución 1.07 segundos									
9 Iteración k.means con 9 clusters de tamaño: 52, 37, 23, 38, 28, 27, 28, 27, 20									
Muestras	Grupos								
	1	2	3	4	5	6	7	8	9
DisparoReal	20	19	2	11	28	11	20	26	0
DisparoSalva	17	5	0	12	0	6	1	1	2
EfectoSonido	0	12	15	6	0	1	0	0	15
Explosivo	15	1	6	9	0	9	7	0	3

Tiempo de ejecución 0.92 segundos

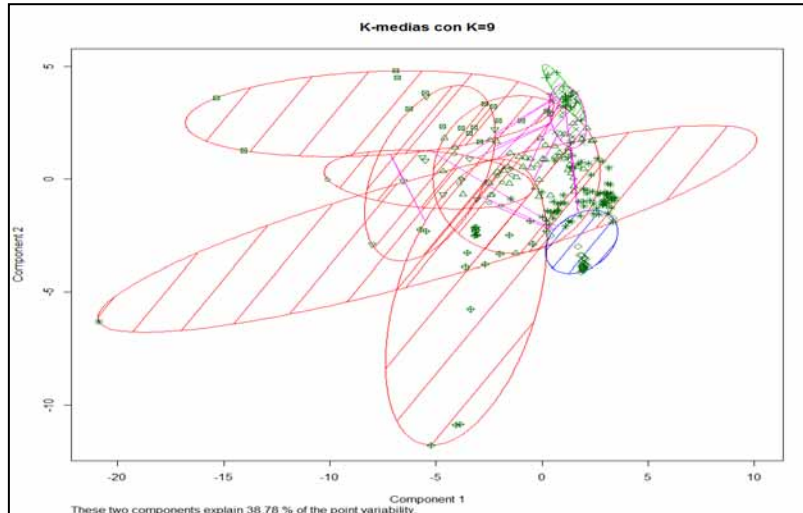


Figura C.20: Iteración 1. Kmeans con 9 clusters

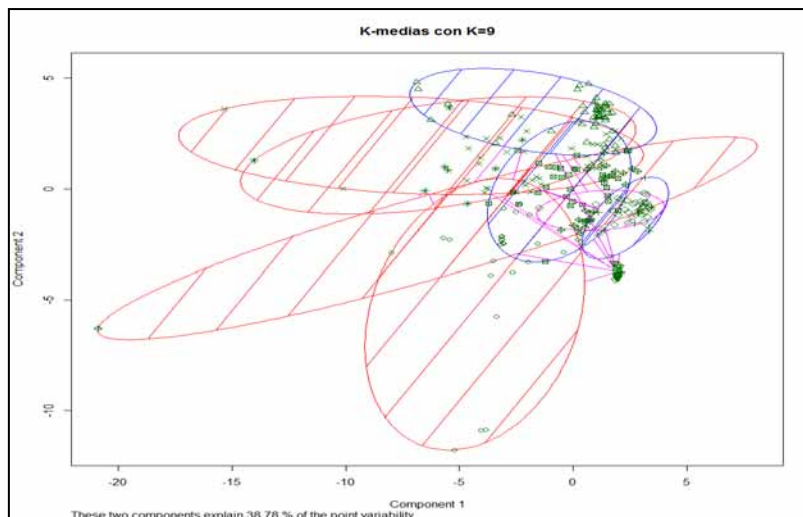


Figura C.21: Iteración 5. Kmeans con 8 clusters

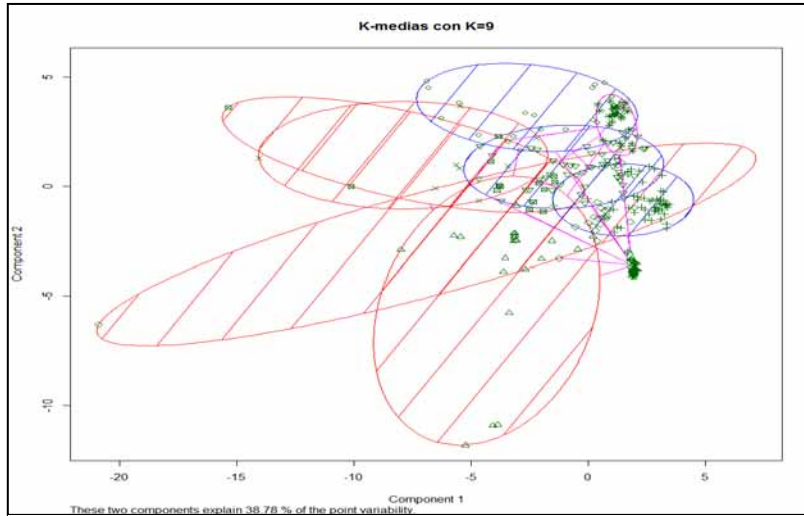


Figura C.22: Iteración 7. Kmeans con 9 clusters

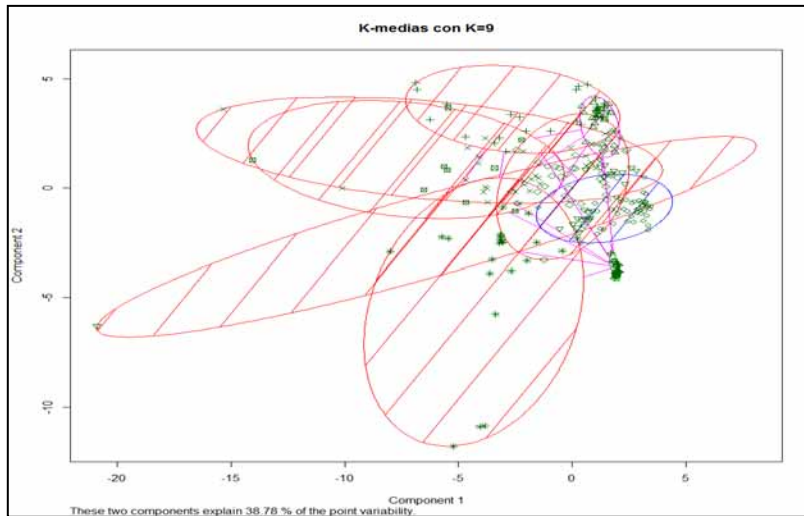


Figura C.23: Iteración 8. Kmeans con 9 clusters

Tabla C.23: Iteración 2,3,4,6 y 10. Kmeans con 9 clusters

2 Iteración k.means con 9 clusters de tamaño: 8, 53, 14, 23, 27, 31, 29, 38, 57									
Muestras	Grupos								
	1	2	3	4	5	6	7	8	9
DisparoReal	0	21	0	2	26	29	21	21	18
DisparoSalva	1	17	1	0	1	0	1	1	18
EfectoSonido	6	0	11	15	0	0	0	12	5
Explosivo	1	15	2	6	0	2	7	1	16
Tiempo de ejecución 0.72 segundos									
3 Iteración k.means con 9 clusters de tamaño: 29, 24, 52, 29, 27, 33, 47, 9, 30									
Muestras	Grupos								
	1	2	3	4	5	6	7	8	9
DisparoReal	1	2	21	21	26	20	18	0	28
DisparoSalva	7	0	17	1	1	4	13	1	0
EfectoSonido	15	15	0	0	0	8	4	7	0
Explosivo	6	7	14	7	0	1	12	1	2
Tiempo de ejecución 0.81 segundos									
4 Iteración k.means con 9 clusters de tamaño: 35, 27, 38, 14, 77, 8, 31, 27, 23									
Muestras	Grupos								
	1	2	3	4	5	6	7	8	9
DisparoReal	10	26	20	0	39	0	29	11	2
DisparoSalva	12	1	5	1	18	1	0	6	0
EfectoSonido	4	0	12	11	0	6	0	1	15
Explosivo	9	0	1	2	20	1	2	9	6
Tiempo de ejecución 0.80 segundos									
6 Iteración k.means con 9 clusters de tamaño: 37, 17, 37, 30, 13, 29, 29, 36, 52									
Muestras	Grupos								
	1	2	3	4	5	6	7	8	9
DisparoReal	11	2	20	28	0	21	26	8	21
DisparoSalva	12	0	4	0	0	1	2	8	17
EfectoSonido	3	11	12	0	9	0	1	13	0
Explosivo	11	4	1	2	4	7	0	7	14
Tiempo de ejecución 1.03 segundos									
10 Iteración k.means con 9 clusters de tamaño: 30, 27, 28, 20, 38, 27, 50, 23, 37									
Muestras	Grupos								
	1	2	3	4	5	6	7	8	9
DisparoReal	28	11	20	0	11	26	20	2	19
DisparoSalva	0	6	1	2	12	1	17	0	5
EfectoSonido	0	1	0	15	6	0	0	15	12

Explosivo	2	9	7	3	9	0	13	6	1
Tiempo de ejecución 0.93 segundos									

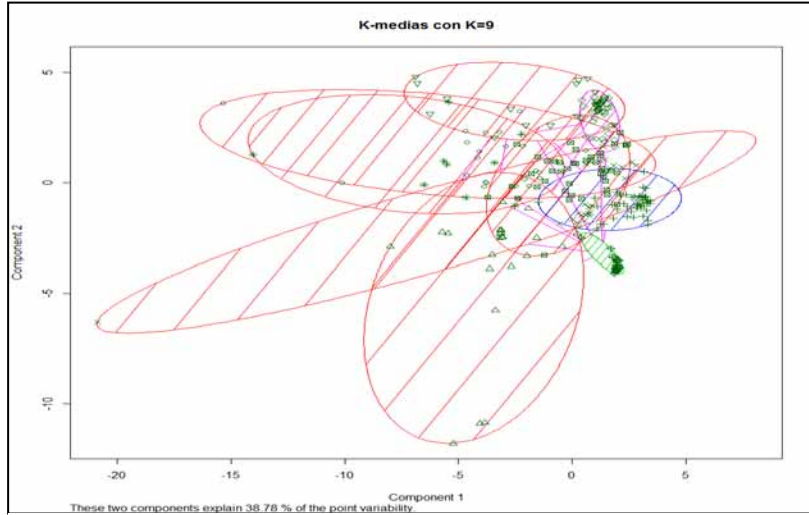


Figura C.24: Iteración 3. Kmeans con 9 clusters

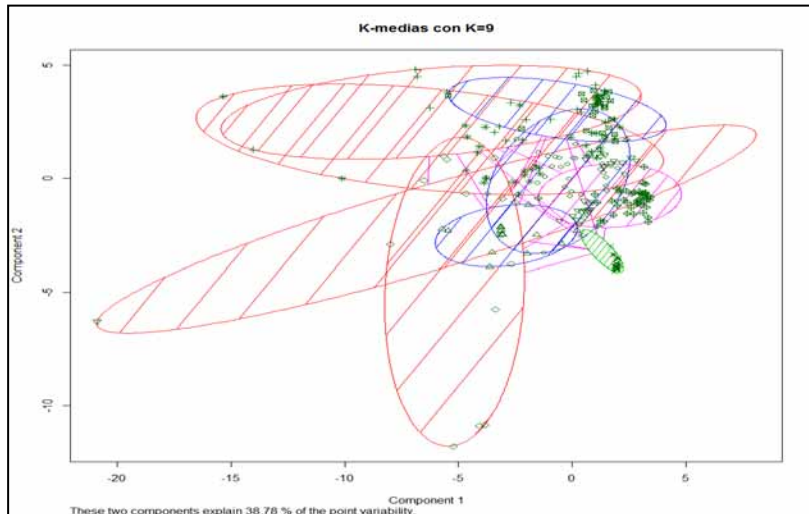


Figura C.25: Iteración 6. Kmeans con 9 clusters

- **K-Means con 10 Clusters**

Tabla C.24: Iteraciones 2,3,4 y 5. Kmeans con 10 clusters

2 Iteración k.means con 10 clusters de tamaño: 44, 26, 27, 27, 24, 27, 15, 48, 9, 33										
Muestras	Grupos									
	1	2	3	4	5	6	7	8	9	10
DisparoReal	14	1	26	19	2	27	11	17	0	20
DisparoSalva	17	7	1	1	0	0	0	13	1	4
EfectoSonido	0	15	0	0	15	0	0	4	7	8
Explosivo	13	3	0	7	7	0	4	14	1	1
Tiempo de ejecución 0,73 segundos										
3 Iteración k.means con 10 clusters de tamaño: 14, 43, 23, 20, 40, 52, 28, 27, 16, 17										
Muestras	Grupos									
	1	2	3	4	5	6	7	8	9	10
DisparoReal	6	18	2	0	20	16	28	26	14	7
DisparoSalva	1	17	0	2	5	17	0	1	0	1
EfectoSonido	0	0	15	15	14	5	0	0	0	0
Explosivo	7	8	3	1	1	14	0	0	2	9
Tiempo de ejecución 0.69 segundos										
4 Iteración k.means con 10 clusters de tamaño: 42, 38, 27, 35, 32, 27, 15, 23, 14, 27										
Muestras	Grupos									
	1	2	3	4	5	6	7	8	9	10
DisparoReal	13	19	27	11	26	9	11	0	2	19
DisparoSalva	17	5	0	11	2	7	0	1	0	1
EfectoSonido	0	13	0	3	4	5	0	19	5	0
Explosivo	12	1	0	10	0	6	4	3	7	7
Tiempo de ejecución 0.62 segundos										
5 Iteración k.means con 10 clusters de tamaño: 27, 9, 24, 29, 46, 43, 27, 33, 27, 15										
Muestras	Grupos									
	1	2	3	4	5	6	7	8	9	10
DisparoReal	19	0	2	1	17	14	26	20	27	11
DisparoSalva	1	1	0	7	13	17	1	4	0	0
EfectoSonido	0	7	15	15	4	0	0	8	0	0
Explosivo	7	1	7	6	12	12	0	1	0	4
Tiempo de ejecución 0.87 segundos										

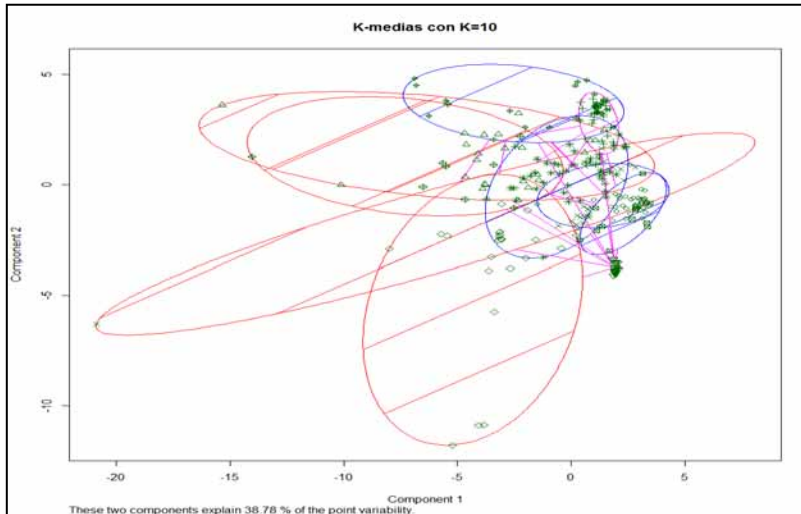


Figura C.26: Iteración 2. Kmeans con 10 clusters

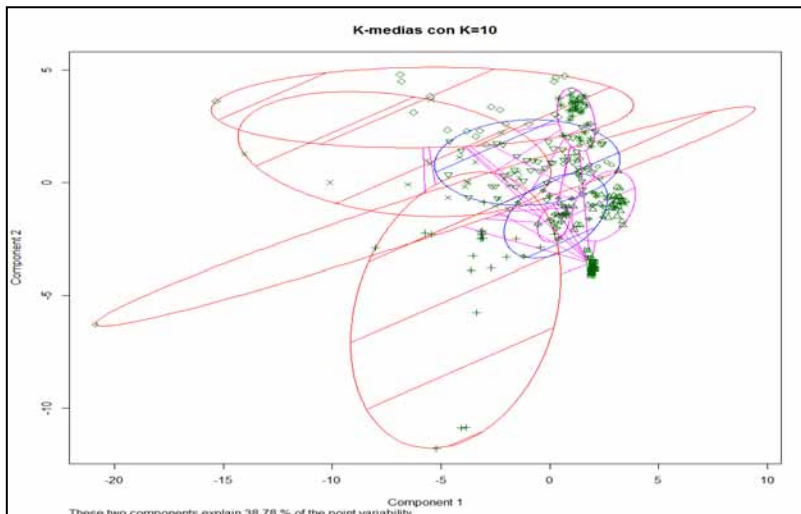


Figura C.27: Iteración 3. Kmeans con 10 clusters

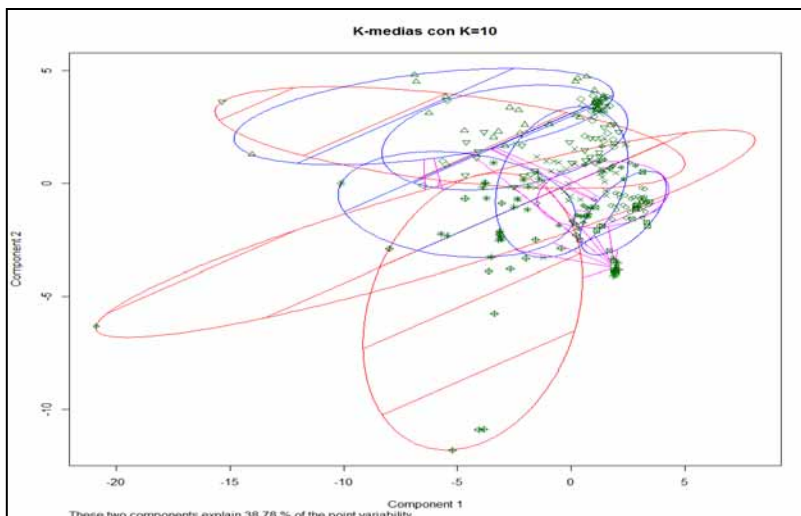


Figura C.28: Iteración 4. Kmeans con 10 clusters

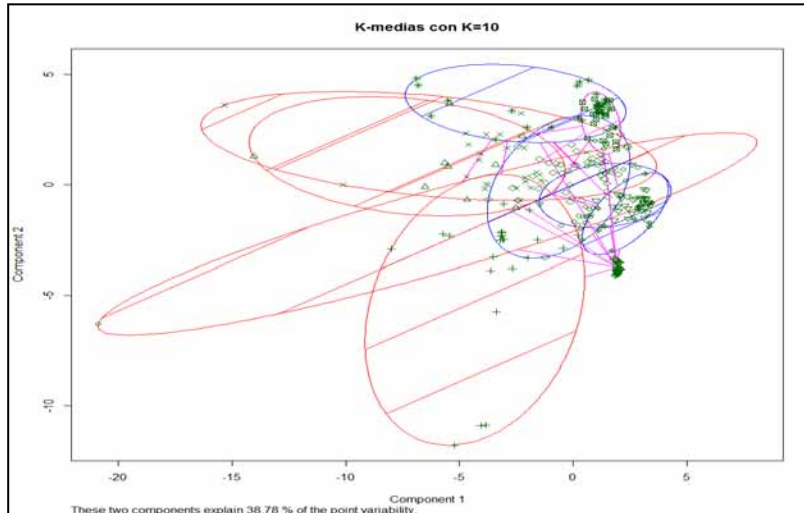


Figura C.29: Iteración 5. Kmeans con 10 clusters

Tabla C.25: Iteración 6 y 7 . Kmeans con 10 clusters

6 Iteración k.means con 10 clusters de tamaño: 9, 25, 20, 27, 17, 53, 69, 20, 27, 13										
Muestras	Grupos									
	1	2	3	4	5	6	7	8	9	10
DisparoReal	6	7	14	2	7	16	46	12	27	0
DisparoSalva	0	16	1	1	1	17	0	1	0	0
EfectoSonido	0	0	0	18	0	7	14	0	0	9
Explosivo	3	2	5	6	9	13	6	7	0	4
Tiempo de ejecución 0.79 segundos										
7 Iteración k.means con 10 clusters de tamaño: 40, 32, 23, 27, 27, 9, 27, 20, 20, 55										
Muestras	Grupos									
	1	2	3	4	5	6	7	8	9	10
DisparoReal	20	14	2	26	27	6	7	17	0	18
DisparoSalva	5	4	0	1	0	0	13	2	2	17
EfectoSonido	14	0	15	0	0	0	0	0	15	5
Explosivo	1	14	6	0	0	3	7	1	3	15
Tiempo de ejecución 0.61 segundos										

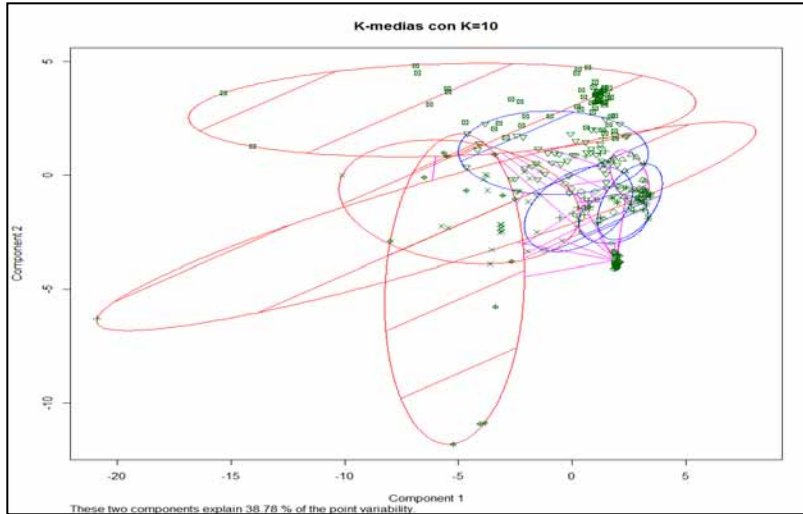


Figura C.30: Iteración 6. Kmeans con 10 clusters

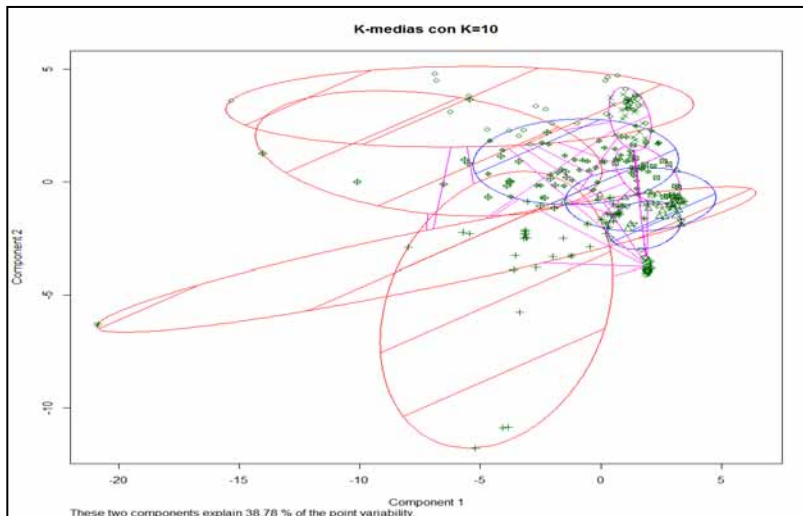


Figura C.31: Iteración 7. Kmeans con 10 clusters

Tabla C.26: Iteraciones 8,9 y 10. Kmeans con 10 clusters

8 Iteración k.means con 10 clusters de tamaño: 27, 38, 28, 27, 23, 35, 8, 50, 14, 30										
Muestras	Grupos									
	1	2	3	4	5	6	7	8	9	10
DisparoReal	26	20	20	11	2	10	0	20	0	28
DisparoSalva	1	5	1	6	0	12	1	17	1	0
EfectoSonido	0	12	0	1	15	4	6	0	11	0
Explosivo	0	1	7	9	6	9	1	13	2	2
Tiempo de ejecución 0.61 segundos										
9 Iteración k.means con 10 clusters de tamaño: 42, 23, 28, 27, 8, 49, 14, 28, 23, 38										
Muestras	Grupos									
	1	2	3	4	5	6	7	8	9	10
DisparoReal	11	2	20	26	0	18	0	28	12	20
DisparoSalva	15	0	1	1	1	17	1	0	3	5
EfectoSonido	5	15	0	0	6	0	11	0	0	12
Explosivo	11	6	7	0	1	14	2	0	8	1
Tiempo de ejecución 0.68 segundos										
10 Iteración k.means con 10 clusters de tamaño: 41, 22, 17, 51, 28, 43, 17, 6, 27, 28										
Muestras	Grupos									
	1	2	3	4	5	6	7	8	9	10
DisparoReal	20	0	2	16	28	18	7	0	26	20
DisparoSalva	5	2	0	17	0	17	1	0	1	1
EfectoSonido	15	17	9	5	0	0	0	3	0	0
Explosivo	1	3	6	13	0	8	9	3	0	7
Tiempo de ejecución 0.80										

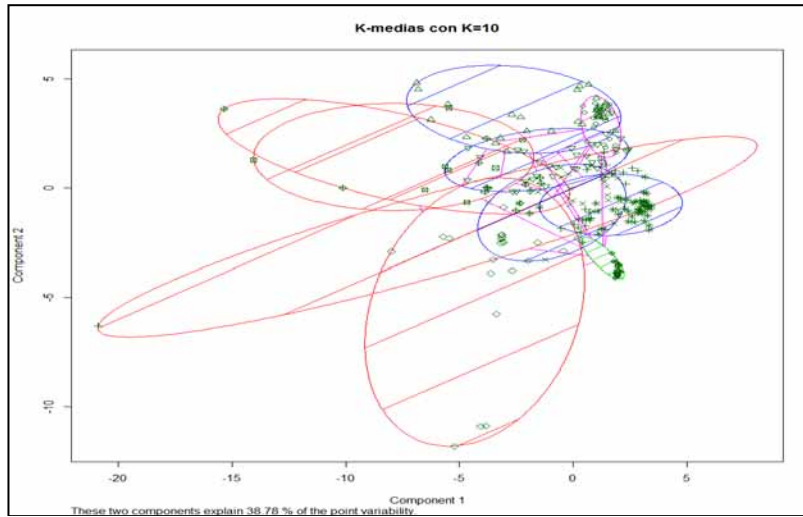


Figura C.32: Iteración 8. Kmeans con 10 clusters

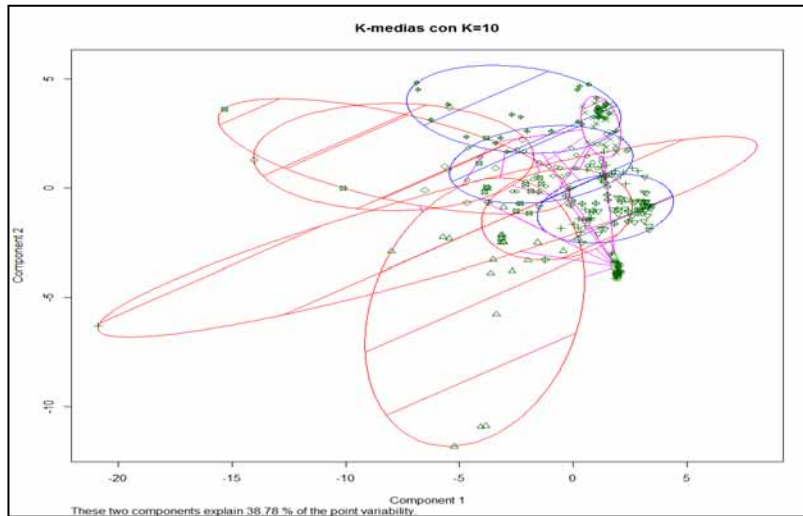


Figura C.33: Iteración 9. Kmeans con 10 clusters

C.2 Resultados obtenidos al implementar K-Means sobre 301 observaciones promediadas por bandas de frecuencia de 50 en 50 Hz

- **K-Means con 2 clusters**

La *Tabla C.27*, expone el resultado al particionar en 2 cluster y es el mismo resultado obtenido al iterar 10 veces.

Tabla C.27: Resultados obtenidos del algoritmo k-medias con k=2. Iteraciones de 1 a 10

1 iteración k-means con 2 clusters de tamaño: 178, 123		
Muestras	Grupos	
	1	2
DisparoReal	96	56
DisparoSalva	45	5
EfectoSonido	7	42
Explosivo	30	20
Tiempo de ejecución 0.64 segundos		

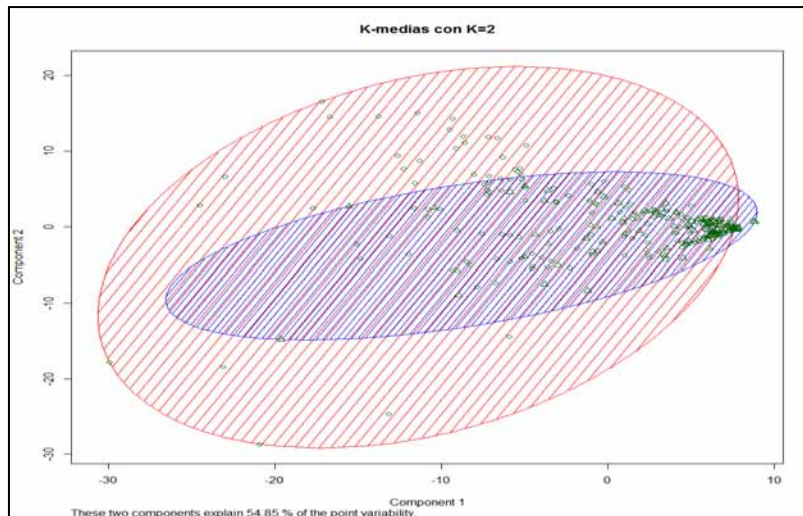


Figura C.34: Iteración 1. Kmeans con 2 clusters

- **K-Means con 3 clusters**

La *Tabla C.28*, expone el resultado al particionar en 3 cluster y es el mismo resultado obtenido en cada unas de las iteraciones. La *Tabla C.29* muestra los tiempos de ejecución por iteración.

Tabla C.28: K-Means con 3 clusters

k-means con 3 clusters de tamaño: 141, 48, 112			
Muestras	Grupos		
	1	2	3
DisparoReal	68	34	50
DisparoSalva	41	6	3
EfectoSonido	6	3	40
Explosivo	26	5	19

Tabla C.29: Tiempo de ejecución por cada unas de las iteraciones

Número de Iteración	Tiempo de ejecución
1	0.86 segundos
2	0.71 segundos
3	0.67 segundos
4	0.81 segundos
5	0.70 segundos
6	0.72 segundos
7	0.64 segundos
8	0.74 segundos
9	0.67 segundos
10	0.69 segundos

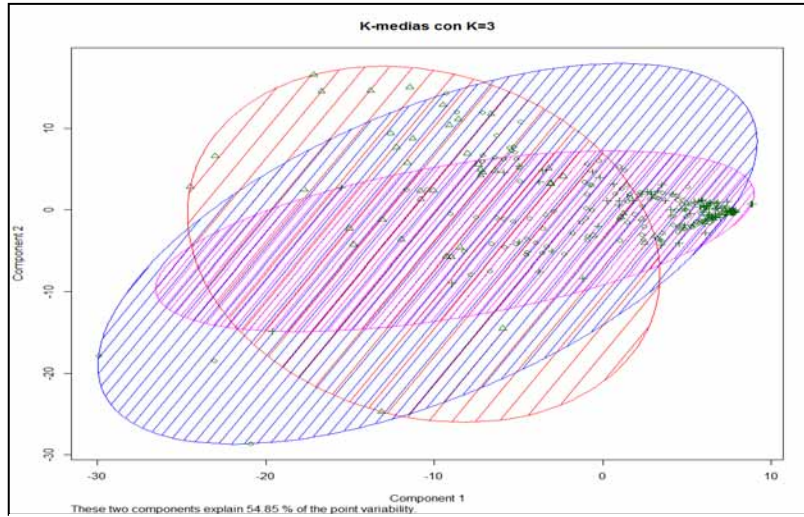


Figura C.35: K-Means con 3 clusters

- K-Means con 4 clusters

Tabla C.30: Iteraciones 1,2,3 y 4. K-Means con 4 clusters

1 iteración k-means con 4 clusters de tamaño: 110, 95, 65, 31				
Muestras	Grupos			
	1	2	3	4
DisparoReal	49	44	40	19
DisparoSalva	3	24	18	5
EfectoSonido	40	3	3	3
Explosivo	18	24	4	4
Tiempo de ejecución 0.80 segundos				
2 iteración k-means con 4 clusters de tamaño: 71, 43, 48, 139				
Muestras	Grupos			
	1	2	3	4
DisparoReal	16	36	34	66
DisparoSalva	3	0	6	41
EfectoSonido	39	1	3	6
Explosivo	13	6	5	26
Tiempo de ejecución 0.86 segundos				
3 iteración k-means con 4 clusters de tamaño: 101, 47, 116, 37				
Muestras	Grupos			
	1	2	3	4
DisparoReal	50	34	68	0
DisparoSalva	1	5	31	13
EfectoSonido	35	3	2	9
Explosivo	15	5	15	15
Tiempo de ejecución 0.83 segundos				
4 iteración k-means con 4 clusters de tamaño: 139, 48, 43, 71				
Muestras	Grupos			
	1	2	3	4
DisparoReal	66	34	36	16
DisparoSalva	41	6	0	3
EfectoSonido	6	3	1	39
Explosivo	26	5	6	13
Tiempo de ejecución 0.76 segundos				

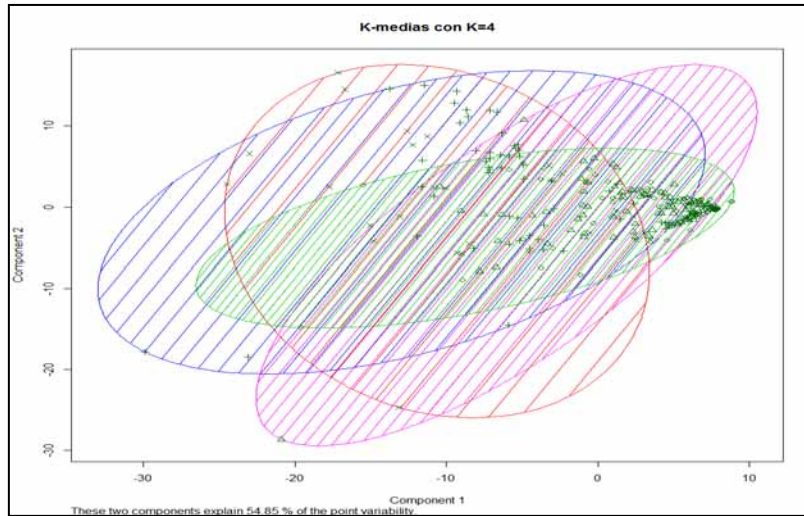


Figura C.36: Iteración 1. Kmeans con 4 clusters

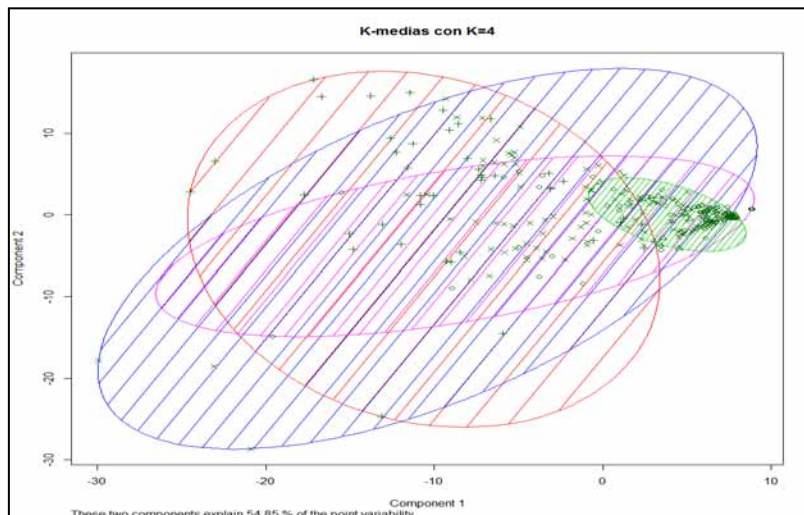


Figura C.37: Iteración 2. Kmeans con 4 clusters

Tabla C.31: Iteraciones 5,6,7 y 8. K-Means con 4 clusters

5 iteración k-means con 4 clusters de tamaño: 116, 101, 47, 37				
Muestras	Grupos			
	1	2	3	4
DisparoReal	68	50	34	0
DisparoSalva	31	1	5	13
EfectoSonido	2	35	3	9
Explosivo	15	15	5	15
Tiempo de ejecución 0.72 segundos				
6 iteración k-means con 4 clusters de tamaño: 53, 92, 119, 37				
Muestras	Grupos			
	1	2	3	4
DisparoReal	14	45	69	24
DisparoSalva	11	1	32	6
EfectoSonido	12	33	1	3
Explosivo	16	13	17	4
Tiempo de ejecución 0.80 segundos				
7 iteración k-means con 4 clusters de tamaño: 86, 125, 43, 47				
Muestras	Grupos			
	1	2	3	4
DisparoReal	50	68	0	34
DisparoSalva	1	35	9	5
EfectoSonido	23	3	20	3
Explosivo	12	19	14	5
Tiempo de ejecución 0.72 segundos				
8 iteración k-means con 4 clusters de tamaño: 125, 47, 86, 43				
Muestras	Grupos			
	1	2	3	4
DisparoReal	68	34	50	0
DisparoSalva	35	5	1	9
EfectoSonido	3	3	23	20
Explosivo	19	5	12	14
Tiempo de ejecución 0.83 segundos				

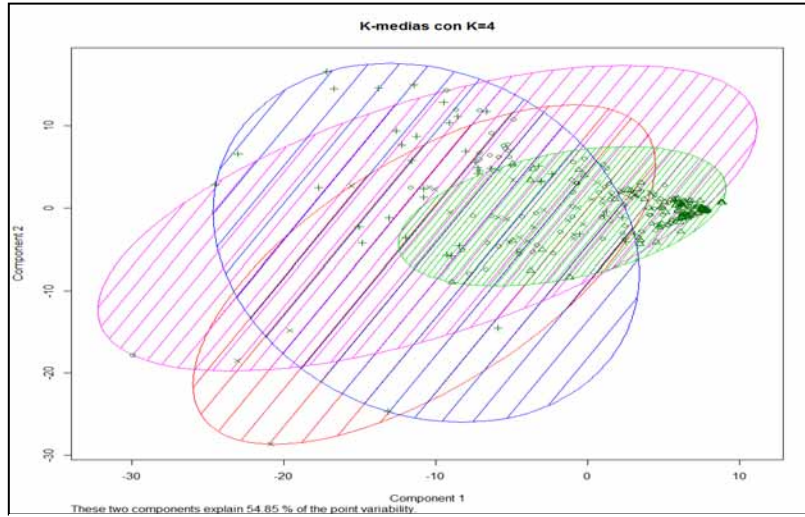


Figura C.38: Iteración 5. Kmeans con 4 clusters

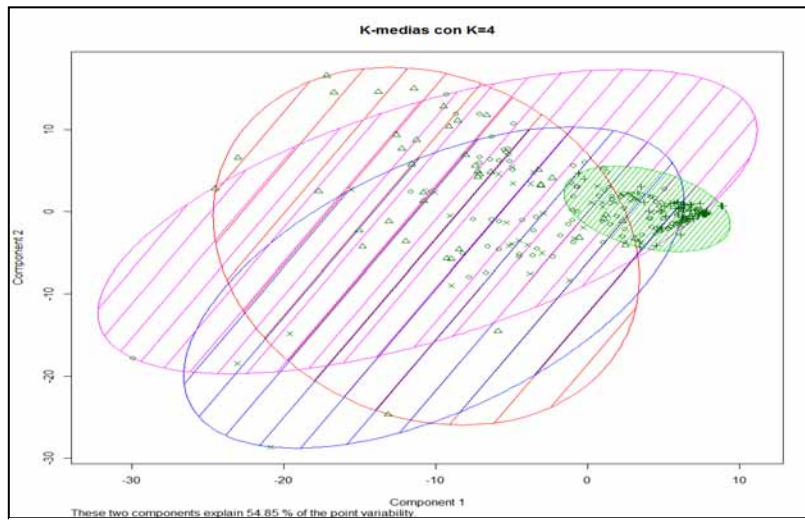


Figura C.39: Iteración 8. Kmeans con 4 clusters

Tabla C.32: Iteraciones 9 y 10. Kmeans con 4 clusters

9 iteración k-means con 4 clusters de tamaño: 71, 139, 48, 43				
Muestras	Grupos			
	1	2	3	4
DisparoReal	16	66	34	36
DisparoSalva	3	41	6	0
EfectoSonido	39	6	3	1
Explosivo	13	26	5	6
Tiempo de ejecución 0.77 segundos				
10 iteración k-means con 4 clusters de tamaño: 43, 139, 48, 71				
Muestras	Grupos			
	1	2	3	4
DisparoReal	36	66	34	16
DisparoSalva	0	41	6	3
EfectoSonido	1	6	3	39
Explosivos	6	26	5	13
Tiempo de ejecución 0.87 segundos				

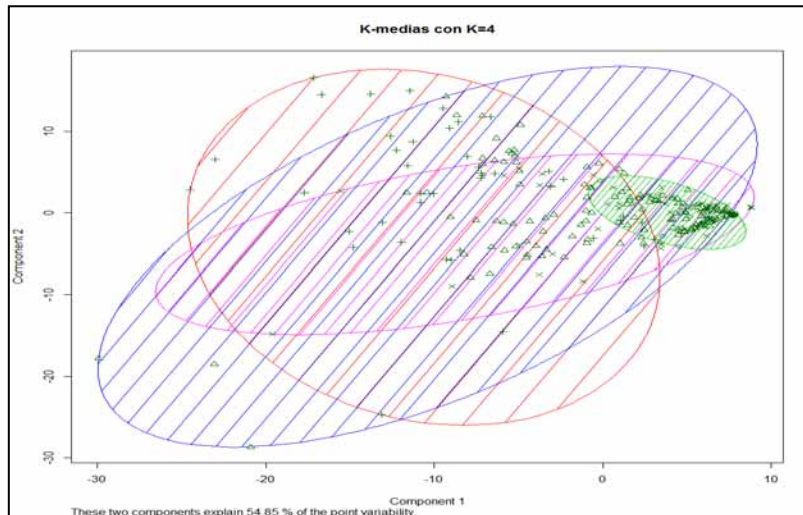


Figura C.40: Iteración 10. Kmeans con 4 clusters

- **K-Means con 5 clusters**

Tabla C.33: Iteración 1,2,3 y 4. K-Means con 4 clusters

1 iteración k-means con 5 clusters de tamaño: 47, 71, 109, 23, 51					
Muestras	Grupos				
	1	2	3	4	5
DisparoReal	33	44	60	1	14
DisparoSalva	6	1	32	0	11
EfectoSonido	3	17	1	18	10
Explosivo	5	9	16	4	16
Tiempo de ejecución 0.83 segundos					
2 iteración k-means con 5 clusters de tamaño: 31, 42, 85, 58, 85					
Muestras	Grupos				
	1	2	3	4	5
DisparoReal	20	0	43	40	49
DisparoSalva	4	9	21	15	1
EfectoSonido	3	20	3	0	23
Explosivo	4	13	18	3	12
Tiempo de ejecución 1.05 segundos					
3 iteración k-means con 5 clusters de tamaño: 58, 31, 85, 42, 85					
Muestras	Grupos				
	1	2	3	4	5
DisparoReal	40	20	49	0	43
DisparoSalva	15	4	1	9	21
EfectoSonido	0	3	23	20	3
Explosivo	3	4	12	13	18
Tiempo de ejecución 0.69 segundos					
4 iteración k-means con 5 clusters de tamaño: 32, 61, 94, 68, 46					
Muestras	Grupos				
	1	2	3	4	5
DisparoReal	20	17	46	40	29
DisparoSalva	5	3	18	24	0
EfectoSonido	3	30	5	1	10
Explosivo	4	11	25	3	7
Tiempo de ejecución 0.83 segundos					

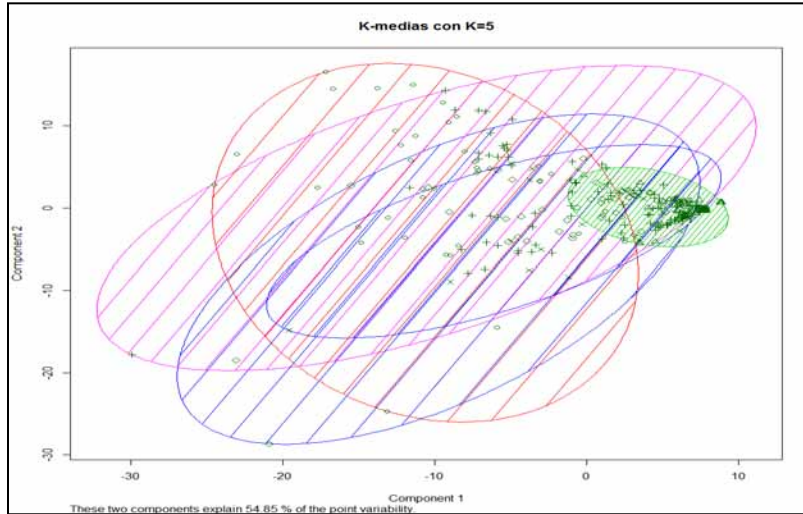


Figura C.41: Iteración 1. Kmeans con 5 clusters

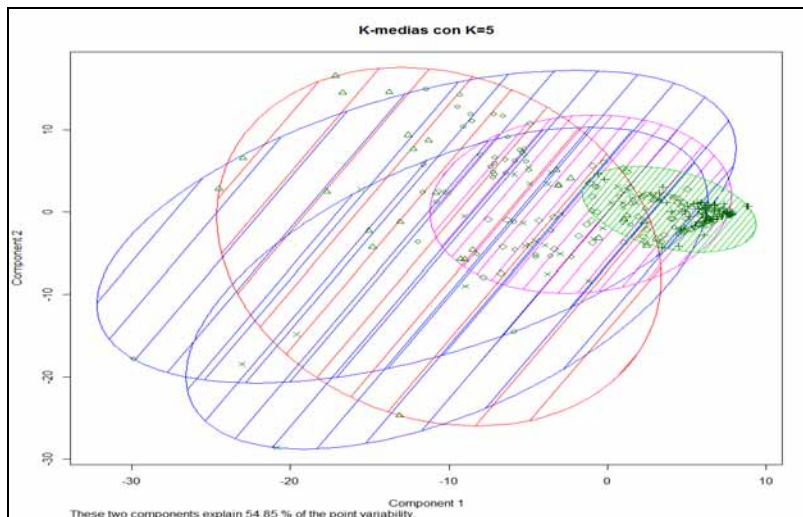


Figura C.42: Iteración 3. K-Means con 5 clusters

Tabla C . 34: Iteraciones 5,6,7 y 8. Kmeans con 5 clusters

5 iteración k-means con 5 clusters de tamaño: 62, 34, 47, 111, 47					
Muestras	Grupos				
	1	2	3	4	5
DisparoReal	16	1	38	63	34
DisparoSalva	1	12	1	31	5
EfectoSonido	35	8	1	3	3
Explosivo	10	13	7	15	5
Tiempo de ejecución 0.72 segundos					
6 iteración k-means con 5 clusters de tamaño: 54, 82, 30, 34, 101					
Muestras	Grupos				
	1	2	3	4	5
DisparoReal	40	43	20	0	49
DisparoSalva	13	20	4	12	1
EfectoSonido	0	3	2	9	35
Explosivo	1	16	4	13	16
Tiempo de ejecución 0.99 segundos					
7 iteración k-means con 5 clusters de tamaño: 42, 59, 114, 47, 39					
Muestras	Grupos				
	1	2	3	4	5
DisparoReal	36	16	66	34	0
DisparoSalva	0	1	31	5	13
EfectoSonido	1	33	2	3	10
Explosivo	5	9	15	5	16
Tiempo de ejecución 0.73 segundos					
8 iteración k-means con 5 clusters de tamaño: 54, 34, 30, 82, 101					
Muestras	Grupos				
	1	2	3	4	5
DisparoReal	40	0	20	43	49
DisparoSalva	13	12	4	20	1
EfectoSonido	0	9	2	3	35
Explosivo	1	13	4	16	16
Tiempo de ejecución 0.80 segundos					

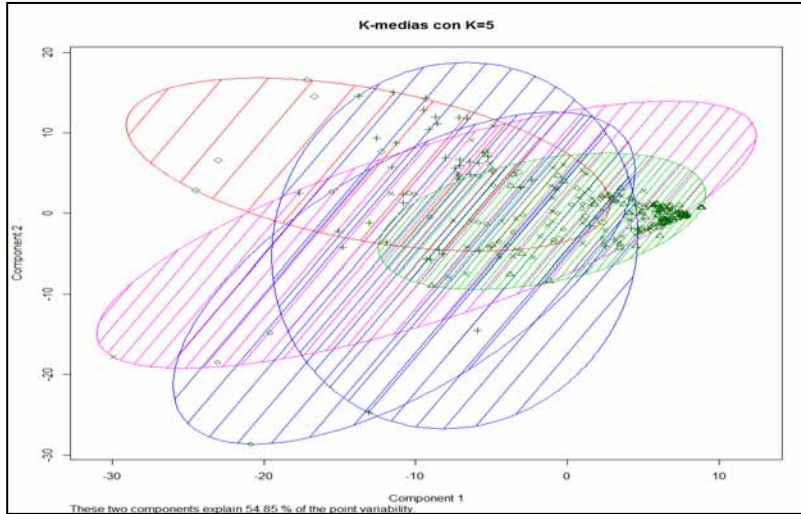


Figura C.43: Iteración 8. Kmeans con 5 clusters

Tabla C.35: Iteraciones 9 y 10. Kmeans con 5 clusters

9 iteración k-means con 5 clusters de tamaño: 36, 100, 52, 103, 10					
Muestras	Grupos				
	1	2	3	4	5
DisparoReal	0	49	42	55	6
DisparoSalva	12	1	6	31	0
EfectoSonido	9	35	3	2	0
Explosivo	15	15	1	15	4
Tiempo de ejecución 0.8 segundos					
10 iteración k-means con 5 clusters de tamaño: 59, 39, 47, 42, 114					
Muestras	Grupos				
	1	2	3	4	5
DisparoReal	16	0	34	36	66
DisparoSalva	1	13	5	0	31
EfectoSonido	33	10	3	1	2
Explosivo	9	16	5	5	15
Tiempo de ejecución 0.89 segundos					

- **Resultados al usar k-medias con k=6**

Tabla C.36: Iteraciones 1,2,3 y 4 . Kmeans con 6 clusters

1 Iteración k.means con 6 clusters de tamaño: 82, 35, 61, 53, 31, 39						
Muestras	Grupos					
	1	2	3	4	5	6
DisparoReal	43	0	16	40	20	33
DisparoSalva	21	12	1	12	4	0
EfectoSonido	3	8	34	0	3	1
Explosivo	15	15	10	1	4	5
Tiempo de ejecución 15.06 segundos						
2 Iteración k.means con 6 clusters de tamaño: 109, 45, 50, 17, 46, 34						
Muestras	Grupos					
	1	2	3	4	5	6
DisparoReal	60	16	14	0	33	29
DisparoSalva	32	1	10	2	5	0
EfectoSonido	1	22	10	12	3	1
Explosivo	16	6	16	3	5	4
Tiempo de ejecución 0.92 segundos						
3 Iteración k.means con 6 clusters de tamaño: 19, 57, 26, 82, 83, 34						
Muestras	Grupos					
	1	2	3	4	5	6
DisparoReal	15	38	7	46	46	0
DisparoSalva	0	18	12	1	15	4
EfectoSonido	0	0	4	23	3	19
Explosivo	4	1	3	12	19	11
Tiempo de ejecución 0.87 segundos						
4 Iteración k.means con 6 clusters de tamaño: 29, 36, 85, 61, 40, 50						
Muestras	Grupos					
	1	2	3	4	5	6
DisparoReal	19	0	43	16	34	40
DisparoSalva	4	12	24	1	0	9
EfectoSonido	2	9	3	34	1	0
Explosivo	4	15	15	10	5	1
Tiempo de ejecución 0.92 segundos						

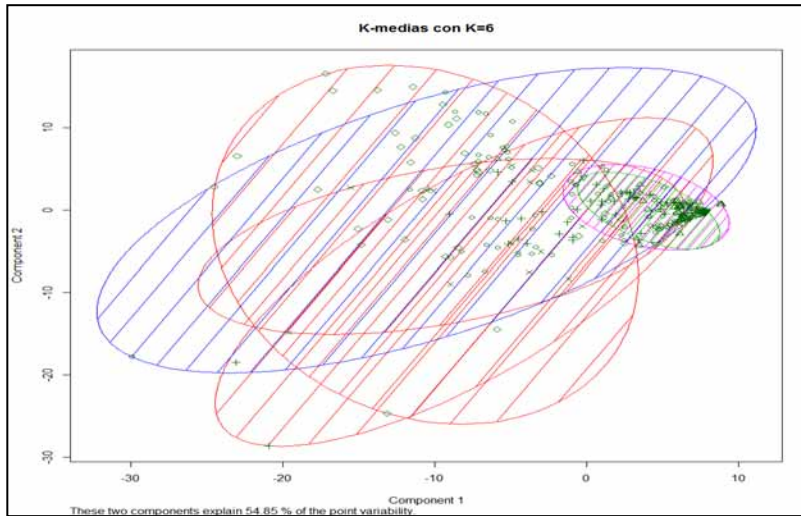


Figura C.44: Iteración 2. Kmeans con 6 clusters

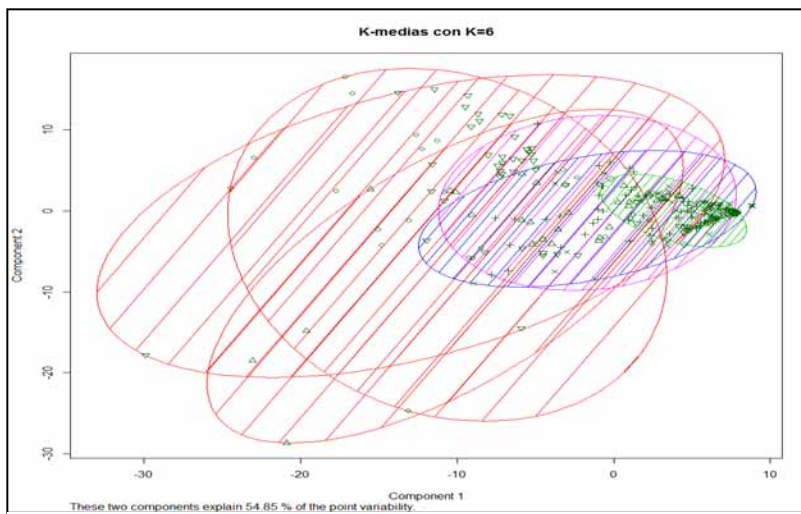


Figura C.45: Iteración 4. Kmeans con 6 clusters

Tabla C.37: Iteraciones 5,6,7 y 8. Kmeans con 6 clusters

5 Iteración k.means con 6 clusters de tamaño: 44, 29, 62, 86, 33, 47						
Muestras	Grupos					
	1	2	3	4	5	6
DisparoReal	33	18	21	20	20	25
DisparoSalva	0	2	17	6	18	1
EfectoSonido	0	17	0	24	8	0
Explosivo	3	7	20	3	17	0
Tiempo de ejecución 0.89 segundos						
6 Iteración k.means con 6 clusters de tamaño: 36, 59, 75, 26, 49, 35						
Muestras	Grupos					
	1	2	3	4	5	6
DisparoReal	36	19	16	41	0	40
DisparoSalva	0	4	1	27	12	6
EfectoSonido	1	2	35	3	8	0
Explosivo	7	4	10	15	13	1
Tiempo de ejecución 0.89 segundos						
7 Iteración k.means con 6 clusters de tamaño: 82, 32, 67, 54, 30, 36						
Muestras	Grupos					
	1	2	3	4	5	6
DisparoReal	43	0	43	40	20	6
DisparoSalva	20	12	0	13	4	1
EfectoSonido	3	7	15	0	2	22
Explosivo	16	13	9	1	4	7
Tiempo de ejecución 1.11 segundos						
8 Iteración k.means con 6 clusters de tamaño: 33, 50, 34, 87, 29, 68						
Muestras	Grupos					
	1	2	3	4	5	6
DisparoReal	0	40	32	44	19	17
DisparoSalva	12	9	0	24	4	1
EfectoSonido	8	0	1	3	2	35
Explosivo	13	1	1	16	4	15
Tiempo de ejecución 0.74 segundos						

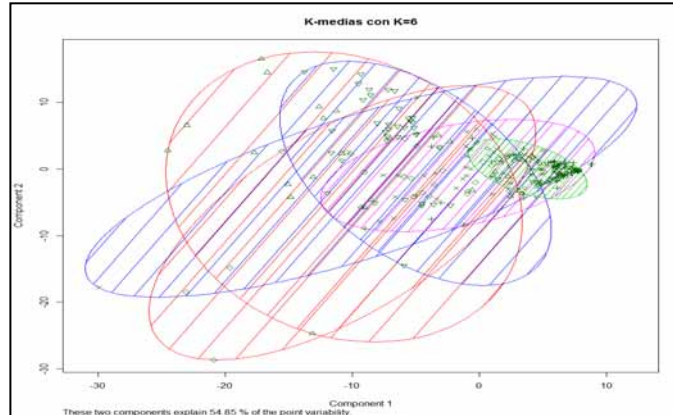


Figura C.46: Iteración 5. Kmeans con 6 clusters

Tabla C.38: Iteraciones 9 y 10. Kmeans con 6 clusters

9 Iteración k.means con 6 clusters de tamaño: 33, 46, 101, 46, 29, 46						
Muestras	Grupos					
	1	2	3	4	5	6
DisparoReal	0	18	49	26	19	40
DisparoSalva	11	18	1	11	4	5
EfectoSonido	9	0	35	3	2	0
Explosivo	13	10	16	6	4	1
Tiempo de ejecución 0.97 segundos						
10 Iteración k.means con 6 clusters de tamaño: 6, 35, 52, 100, 27, 81						
Muestras	Grupos					
	1	2	3	4	5	6
DisparoReal	3	0	39	49	18	43
DisparoSalva	0	12	12	1	5	20
EfectoSonido	0	8	0	35	3	3
Explosivo	3	15	1	15	1	15
Tiempo de ejecución 0.75 segundos						

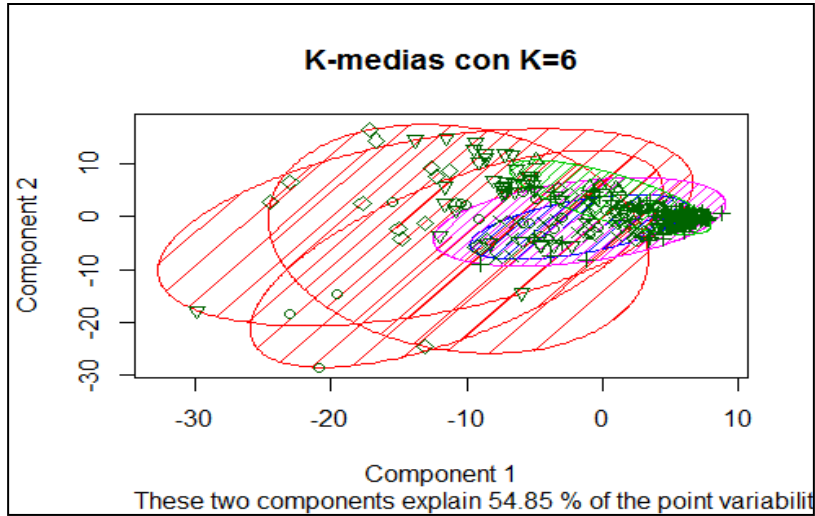


Figura C . 47: Iteración 9. Kmeans con 6 clusters

- **K-Means con 7 clusters**

Tabla C.39: Iteraciones 1 y 2. K-Means con 7 clusters

1 Iteración k.means con 7 clusters de tamaño: 56, 33, 50, 87, 14, 32, 29							
Muestras	Grupos						
	1	2	3	4	5	6	7
DisparoReal	17	32	40	44	0	0	19
DisparoSalva	1	0	9	24	0	12	4
EfectoSonido	25	1	0	3	11	7	2
Explosivo	13	0	1	16	3	13	4
Tiempo de ejecución 0.92 segundos							
2 Iteración k.means con 7 clusters de tamaño: 14, 30, 54, 56, 33, 82, 32							
Muestras	Grupos						
	1	2	3	4	5	6	7
DisparoReal	0	20	40	17	32	43	0
DisparoSalva	0	4	13	1	0	20	12
EfectoSonido	11	2	0	25	1	3	7
Explosivo	3	4	1	13	0	16	13
Tiempo de ejecución 1.03 segundos							

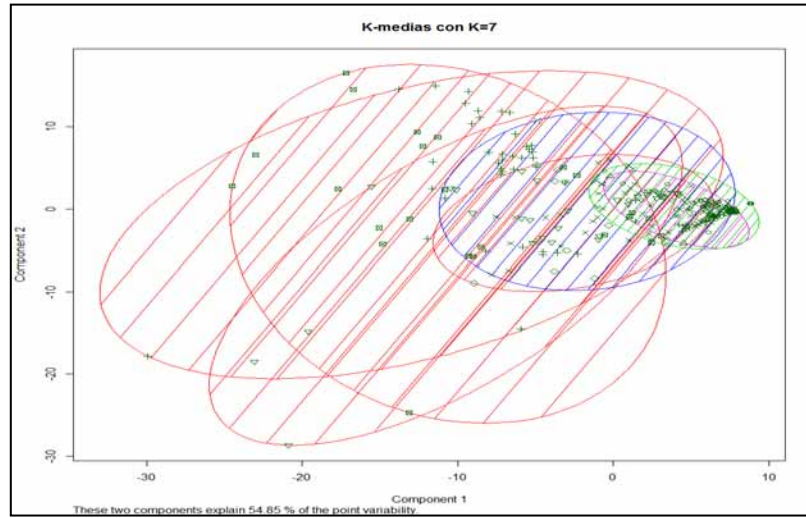


Figura C.48: Iteración 1. Kmeans con 7 clusters

Tabla C.40: Iteraciones 3,4 y 5. Kmeans con 7 clusters

3 Iteración k.means con 7 clusters de tamaño: 40, 30, 31, 66, 63, 29, 42							
Muestras	Grupos						
	1	2	3	4	5	6	7
DisparoReal	26	29	9	36	20	19	25
DisparoSalva	4	0	11	18	6	17	1
EfectoSonido	2	1	8	0	16	5	0
Explosivo	8	0	12	20	2	14	0
Tiempo de ejecución 0.86 segundos							
4 Iteración k.means con 7 clusters de tamaño: 33, 46, 46, 46, 33, 29, 68							
Muestras	Grupos						
	1	2	3	4	5	6	7
DisparoReal	0	18	26	40	6	19	43
DisparoSalva	11	18	11	5	1	4	0
EfectoSonido	9	0	3	0	19	2	16
Explosivo	13	10	6	1	7	4	9
Tiempo de ejecución 0.95 segundos							
5 Iteración k.means con 7 clusters de tamaño: 25, 43, 63, 27, 79, 24, 40							
Muestras	Grupos						
	1	2	3	4	5	6	7
DisparoReal	19	28	24	0	45	0	36
DisparoSalva	2	4	28	10	1	2	3
EfectoSonido	0	2	1	6	23	17	0
Explosivo	4	9	10	11	10	5	1
Tiempo de ejecución 0.88 segundos							

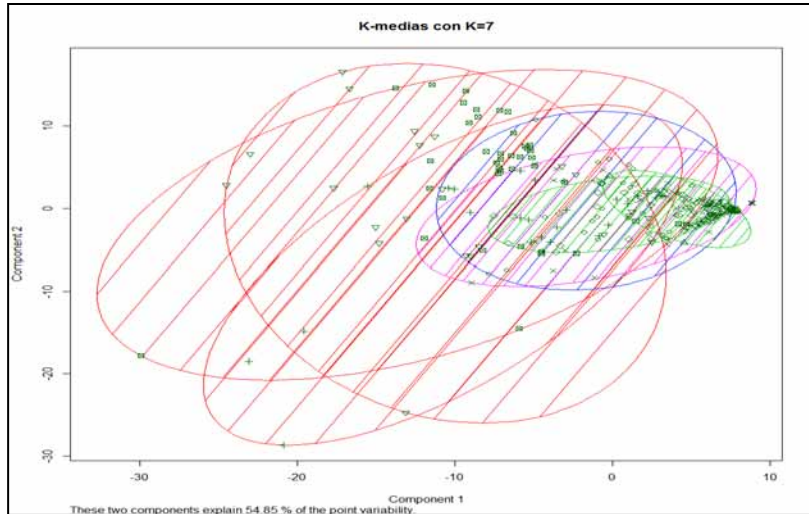


Figura C.49: Iteración 3. Kmeans con 7 clusters

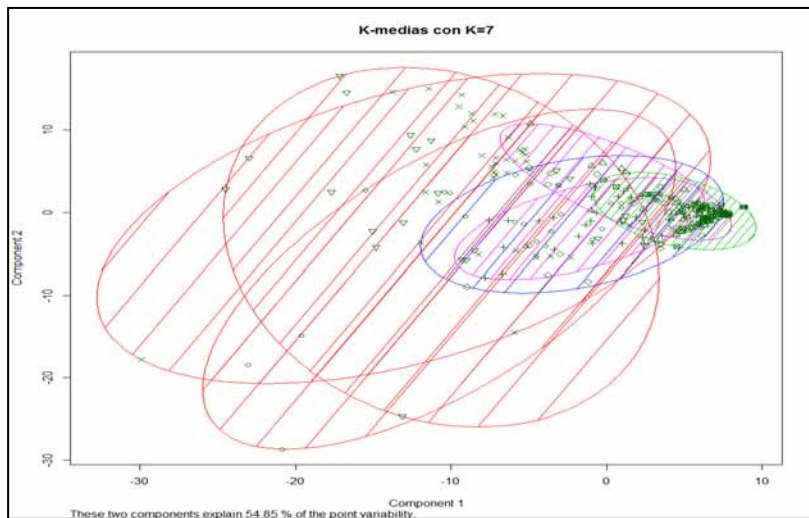


Figura C.50: Iteración 4. Kmeans con 7 clusters

Tabla C.41: Iteraciones 6,7,8 y 9. Kmeans con 7 clusters

6 Iteración k.means con 7 clusters de tamaño: 31, 73, 42, 29, 30, 30, 66							
Muestras	Grupos						
	1	2	3	4	5	6	7
DisparoReal	0	31	38	19	29	19	16
DisparoSalva	11	28	3	4	0	3	1
EfectoSonido	8	1	0	2	1	2	35
Explosivo	12	13	1	4	0	6	14
Tiempo de ejecución 0.86 segundos							
7 Iteración k.means con 7 clusters de tamaño: 82, 54, 30, 38, 26, 24, 47							
Muestras	Grupos						
	1	2	3	4	5	6	7
DisparoReal	43	40	20	32	0	0	17
DisparoSalva	20	13	4	0	10	2	1
EfectoSonido	3	0	2	1	4	17	22
Explosivo	16	1	4	5	12	5	7
Tiempo de ejecución 0.86 segundos							
8 Iteración k.means con 7 clusters de tamaño: 38, 47, 24, 54, 26, 30, 82							
Muestras	Grupos						
	1	2	3	4	5	6	7
DisparoReal	32	17	0	40	0	20	43
DisparoSalva	0	1	2	13	10	4	20
EfectoSonido	1	22	17	0	4	2	3
Explosivo	5	7	5	1	12	4	16
Tiempo de ejecución 0.86 segundos							
9 Iteración k.means con 7 clusters de tamaño: 37, 46, 42, 48, 84, 30, 14							
Muestras	Grupos						
	1	2	3	4	5	6	7
DisparoReal	0	16	36	40	41	19	0
DisparoSalva	12	1	0	7	26	4	0
EfectoSonido	9	23	1	0	2	3	11
Explosivo	16	6	5	1	15	4	3
Tiempo de ejecución 0.98 segundos							

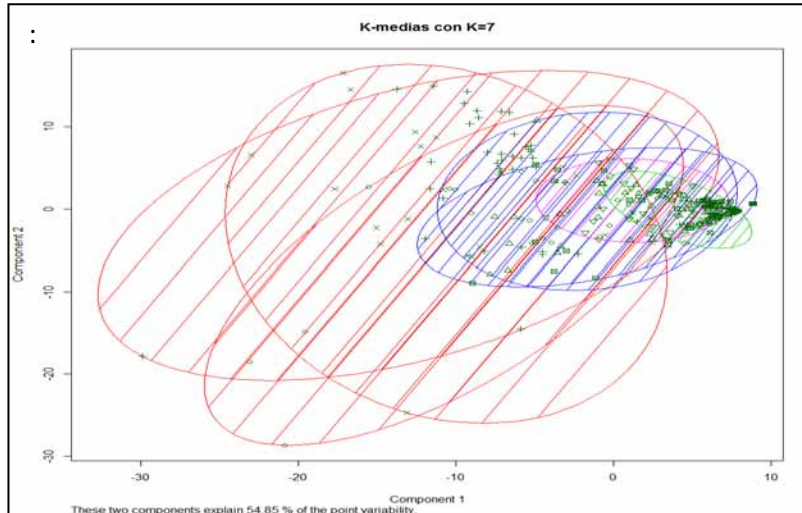


Figura C.51: Iteración 6. Kmeans con 7 clusters

Tabla C . 42: Iteración 10. Kmeans con 7 clusters

10 Iteración k.means con 7 clusters de tamaño: 32, 33, 63, 31, 42, 71, 29							
Muestras	Grupos						
	1	2	3	4	5	6	7
DisparoReal	20	29	16	0	38	30	19
DisparoSalva	3	0	1	11	3	28	4
EfectoSonido	2	1	35	8	0	1	2
Explosivo	7	3	11	12	1	12	4
Tiempo de ejecución 0.89 segundos							

- **K-Means con 8 clusters**

Tabla C.43: Iteraciones 1,2,3 y 4. Kmeans con 8 clusters

1 Iteración k.means con 8 clusters de tamaño: 30, 31, 23, 29, 18, 33, 75, 62								
Muestras	Grupos							
	1	2	3	4	5	6	7	8
DisparoReal	0	28	13	19	16	29	31	16
DisparoSalva	11	2	3	4	1	0	28	1
EfectoSonido	8	0	2	2	0	1	1	35
Explosivo	11	1	5	4	1	3	15	10
Tiempo de ejecución 0.84 segundos								
2 Iteración k.means con 8 clusters de tamaño: 37, 31, 42, 40, 63, 31 17,40								
Muestras	Grupos							
	1	2	3	4	5	6	7	8
DisparoReal	6	25	35	21	40	0	12	13
DisparoSalva	1	1	4	12	0	12	1	19
EfectoSonido	23	2	2	1	14	7	0	0
Explosivo	7	3	1	6	9	12	4	8
Tiempo de ejecución 1.05 segundos								
3 Iteración k.means con 8 clusters de tamaño: 73, 48, 47, 14, 30, 29, 30, 30								
Muestras	Grupos							
	1	2	3	4	5	6	7	8
DisparoReal	33	40	16	0	0	19	15	29
DisparoSalva	24	7	1	0	11	4	3	0
EfectoSonido	1	0	22	11	7	2	5	1
Explosivo	15	1	8	3	12	4	7	0
Tiempo de ejecución 0.97 segundos								
4 Iteración k.means con 8 clusters de tamaño: 25, 68, 28, 30, 42, 35, 49, 24								
Muestras	Grupos							
	1	2	3	4	5	6	7	8
DisparoReal	0	27	19	29	38	23	16	0
DisparoSalva	10	28	3	0	3	3	1	2
EfectoSonido	4	1	2	1	0	2	22	17
Explosivo	11	12	4	0	1	7	10	5
Tiempo de ejecución 0.87 segundos								

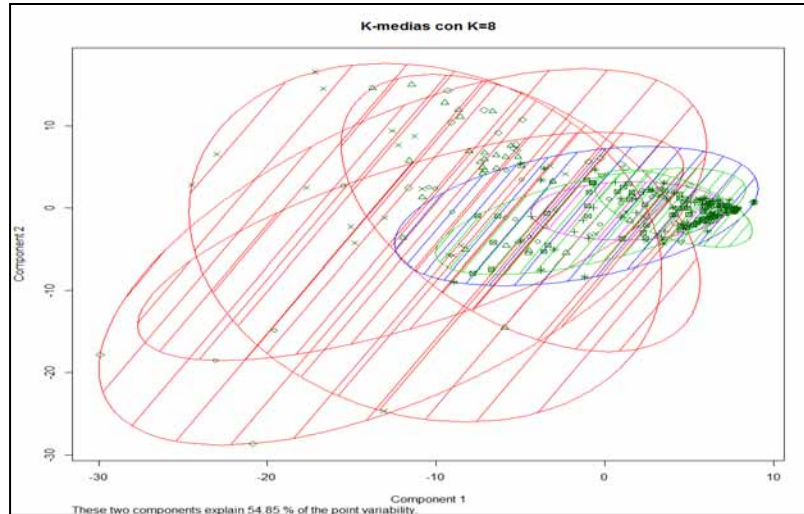


Figura C.52: Iteración 1. Kmeans con 8 clusters

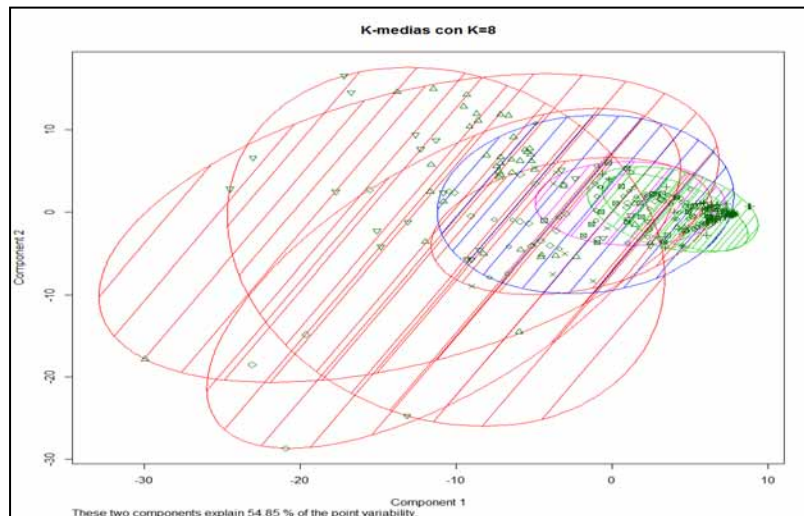


Figura C.53: Iteración 3. Kmeans con 8 clusters

Tabla C.44: Iteraciones 5,6,7 y 8. Kmeans con 8 clusters

5 Iteración k.means con 8 clusters de tamaño: 42, 30, 53, 30, 73, 29, 30, 14								
Muestras	Grupos							
	1	2	3	4	5	6	7	8
DisparoReal	38	0	16	19	31	19	29	0
DisparoSalva	3	11	1	3	28	4	0	0
EfectoSonido	0	7	25	2	1	2	1	11
Explosivo	1	12	11	6	13	4	0	3
Tiempo de ejecución 0.89 segundos								
6 Iteración k.means con 8 clusters de tamaño: 49, 49, 47, 6, 48, 25, 38, 39								
Muestras	Grupos							
	1	2	3	4	5	6	7	8
DisparoReal	19	37	16	3	28	17	32	0
DisparoSalva	20	9	1	0	9	4	0	7
EfectoSonido	0	0	23	0	3	3	1	19
Explosivo	10	3	7	3	8	1	5	13
Tiempo de ejecución 0.89 segundos								
7 Iteración k.means con 8 clusters de tamaño: 14, 53, 63, 30, 30, 29, 50,42								
Muestras	Grupos							
	1	2	3	4	5	6	7	8
DisparoReal	0	16	24	29	0	19	26	38
DisparoSalva	0	1	27	0	11	4	4	3
EfectoSonido	11	25	1	1	7	2	2	0
Explosivo	3	11	11	0	12	4	8	1
Tiempo de ejecución 1.55 segundos								
8 Iteración k.means con 8 clusters de tamaño: 74, 26, 30,23, 19, 58, 37,34								
Muestras	Grupos							
	1	2	3	4	5	6	7	8
DisparoReal	30	0	21	6	15	16	35	29
DisparoSalva	29	6	2	11	0	1	1	0
EfectoSonido	1	9	2	4	0	32	0	1
Explosivo	14	11	5	2	4	9	1	4
Tiempo de ejecución 1.03 segundos								

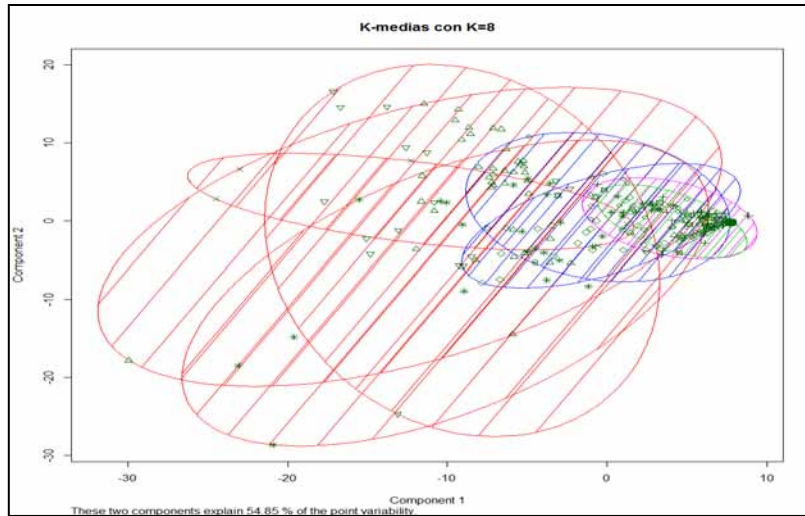


Figura C.54: Iteración 6. Kmeans con 8 clusters

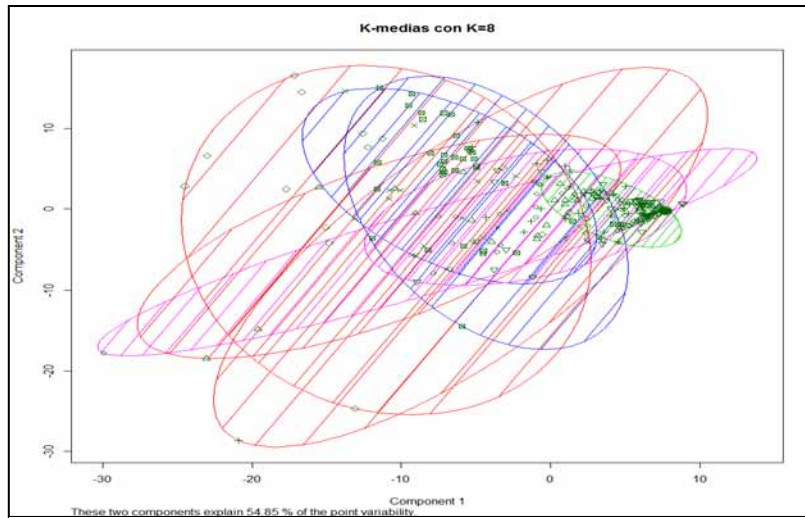


Figura C.55: Iteración 8. Kmeans con 8 clusters

Tabla C.45: Iteraciones 9 y 10. Kmeans con 8 clusters

9 Iteración k.means con 8 clusters de tamaño: 24, 25, 46, 6, 44, 26, 85, 45								
Muestras	Grupos							
	1	2	3	4	5	6	7	8
DisparoReal	0	0	38	3	36	17	42	16
DisparoSalva	2	10	7	0	0	5	25	1
EfectoSonido	17	3	0	0	1	3	3	22
Explosivo	5	12	1	3	7	1	15	6
Tiempo de ejecución 0.94 segundos								
10 Iteración k.means con 8 clusters de tamaño: 24, 21, 40, 23, 11, 32, 79, 71								
Muestras	Grupos							
	1	2	3	4	5	6	7	8
DisparoReal	0	12	37	0	8	20	45	30
DisparoSalva	2	5	2	9	0	3	1	28
EfectoSonido	17	3	0	3	0	2	23	1
Explosivo	5	1	1	11	3	7	10	12
Tiempo de ejecución 0.98 segundos								

- **K-Means con 9 clusters**

Tabla C.46: Iteración 1,2,3 y 4. Kmeans con 9 clusters

1 Iteración k.means con 9 clusters de tamaño: 30, 19, 62, 22, 38, 39, 39, 27, 25									
Muestras	Grupos								
	1	2	3	4	5	6	7	8	9
DisparoReal	29	0	16	1	34	12	21	20	19
DisparoSalva	0	8	1	5	3	18	12	1	2
EfectoSonido	1	6	32	7	0	0	1	2	0
Explosivo	0	5	13	9	1	9	5	4	4
Tiempo de ejecución 0.81 segundos									
2 Iteración k.means con 9 clusters de tamaño: 29, 30, 30, 14, 53, 40, 37, 39, 29									
Muestras	Grupos								
	1	2	3	4	5	6	7	8	9
DisparoReal	19	0	29	0	16	12	34	21	21
DisparoSalva	4	11	0	0	1	18	2	12	2
EfectoSonido	2	7	1	11	25	0	0	1	2
Explosivo	4	12	0	3	11	10	1	5	4
Tiempo de ejecución 1.22 segundos									
3 Iteración k.means con 9 clusters de tamaño: 41, 47, 25, 71, 15, 30, 21, 25, 26									
Muestras	Grupos								
	1	2	3	4	5	6	7	8	9
DisparoReal	37	16	19	28	0	29	0	2	21
DisparoSalva	3	1	2	30	0	0	9	5	0
EfectoSonido	0	22	0	1	12	1	6	6	1
Explosivo	1	8	4	12	3	0	6	12	4
Tiempo de ejecución 1.94 segundos									
4 Iteración k.means con 9 clusters de tamaño: 14, 29, 29, 66, 30, 14, 25, 53, 41									
Muestras	Grupos								
	1	2	3	4	5	6	7	8	9
DisparoReal	0	19	0	26	29	12	13	16	37
DisparoSalva	0	4	11	28	0	0	3	1	3
EfectoSonido	11	2	7	1	1	0	2	25	0
Explosivo	3	4	11	11	0	2	7	11	1
Tiempo de ejecución 1.06 segundos									

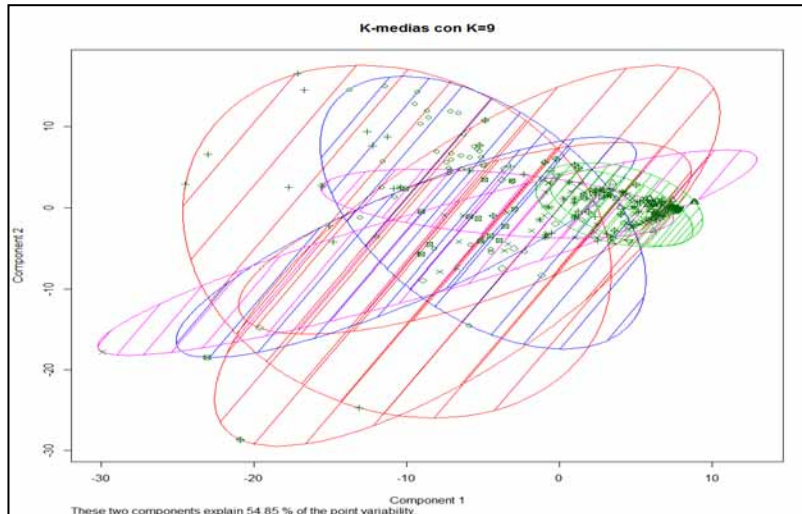


Figura C.56: Iteración 3. Kmeans con 9 clusters

Tabla C.47: Iteraciones 5 y 6. Kmeans con 9 clusters

5 Iteración k.means con 9 clusters de tamaño: 14, 29, 29, 66, 30, 14, 25, 53, 41									
Muestras	Grupos								
	1	2	3	4	5	6	7	8	9
DisparoReal	0	19	0	26	29	12	13	16	37
DisparoSalva	0	4	11	28	0	0	3	1	3
EfectoSonido	11	2	7	1	1	0	2	25	0
Explosivo	3	4	11	11	0	2	7	11	1
Tiempo de ejecución 1.06 segundos									
6 Iteración k.means con 9 clusters de tamaño: 14, 29, 29, 66, 30, 14, 25, 53, 41									
Muestras	Grupos								
	1	2	3	4	5	6	7	8	9
DisparoReal	0	19	0	26	29	12	13	16	37
DisparoSalva	0	4	11	28	0	0	3	1	3
EfectoSonido	11	2	7	1	1	0	2	25	0
Explosivo	3	4	11	11	0	2	7	11	1
Tiempo de ejecución 1.06 segundos									

Tabla C.48: Iteraciones 7,8,9 y 10. Kmeans con 9 clusters

7 Iteración k.means con 9 clusters de tamaño: 37, 45, 30, 33, 41, 36, 13, 32, 34									
Muestras	Grupos								
	1	2	3	4	5	6	7	8	9
DisparoReal	34	16	19	17	28	9	0	0	29
DisparoSalva	2	1	4	11	4	18	0	10	0
EfectoSonido	0	22	3	1	2	0	10	10	1
Explosivo	1	6	4	4	7	9	3	12	4
Tiempo de ejecución 0.79 segundos									
8 Iteración k.means con 9 clusters de tamaño: 25, 24, 30, 38, 28, 48, 25, 39, 44									
Muestras	Grupos								
	1	2	3	4	5	6	7	8	9
DisparoReal	16	0	29	35	19	16	0	21	16
DisparoSalva	2	2	0	2	3	1	10	12	18
EfectoSonido	2	17	1	0	2	22	4	1	0
Explosivo	5	5	0	1	4	9	11	5	10
Tiempo de ejecución 1.89 segundos									
9 Iteración k.means con 9 clusters de tamaño: 45, 37, 23, 13, 42, 19, 53, 44, 25									
Muestras	Grupos								
	1	2	3	4	5	6	7	8	9
DisparoReal	16	32	1	0	37	0	35	12	19
DisparoSalva	1	0	4	0	4	8	9	22	2
EfectoSonido	22	1	7	10	0	6	3	0	0
Explosivo	6	4	11	3	1	5	6	10	4
Tiempo de ejecución 0.74 segundos									
10 Iteración k.means con 9 clusters de tamaño: 30, 13, 50, 33, 41, 36, 37, 30, 31									
Muestras	Grupos								
	1	2	3	4	5	6	7	8	9
DisparoReal	29	0	16	17	28	9	34	19	0
DisparoSalva	0	0	1	11	4	18	2	4	10
EfectoSonido	1	10	23	1	2	0	0	3	9
Explosivo	0	3	10	4	7	9	1	4	12
Tiempo de ejecución 0.89 segundos									

- **K-Means con 10 clusters**

Tabla C.49: Iteraciones 1,2,3 y 4. Kmeans con 10 clusters

1 Iteración k.means con 10 clusters de tamaño: 18, 36, 25, 23, 21, 14, 79, 25, 33, 27										
Muestras	Grupos									
	1	2	3	4	5	6	7	8	9	10
DisparoReal	8	21	0	11	16	13	45	0	29	9
DisparoSalva	3	11	2	12	1	0	1	10	3	7
EfectoSonido	2	1	17	0	0	0	23	6	0	0
Explosivo	5	3	6	0	4	1	10	9	1	11
Tiempo de ejecución 1.08 segundos										
2 Iteración k.means con 10 clusters de tamaño: 36, 79, 33, 14, 21, 24, 24, 20, 25, 25										
Muestras	Grupos									
	1	2	3	4	5	6	7	8	9	10
DisparoReal	33	45	8	12	0	0	14	6	19	15
DisparoSalva	2	1	19	0	9	2	9	4	2	2
EfectoSonido	0	23	0	0	6	17	1	2	0	0
Explosivo	1	10	6	2	6	5	0	8	4	8
Tiempo de ejecución 1.11 segundos										
3 Iteración k.means con 10 clusters de tamaño: 49, 13, 28, 29, 30, 30, 23, 36, 38, 25										
Muestras	Grupos									
	1	2	3	4	5	6	7	8	9	10
DisparoReal	16	0	13	19	0	29	14	34	12	15
DisparoSalva	1	0	11	4	9	0	2	1	19	3
EfectoSonido	23	10	1	2	10	1	2	0	0	0
Explosivo	9	3	3	4	11	0	5	1	7	7
Tiempo de ejecución 0.87 segundos										
4 Iteración k.means con 10 clusters de tamaño: 28, 14, 36, 30, 39, 28, 65, 6, 25, 30										
Muestras	Grupos									
	1	2	3	4	5	6	7	8	9	10
DisparoReal	0	12	33	29	12	13	16	3	17	17
DisparoSalva	10	0	2	0	18	5	1	0	4	10
EfectoSonido	8	0	0	1	0	2	34	0	3	1
Explosivo	10	2	1	0	9	8	14	3	1	2
Tiempo de ejecución 1.03 segundos										

Tabla C.50: Iteraciones 5,6,7 y 8. Kmeans con 10 clusters

5 Iteración k.means con 10 clusters de tamaño: 39, 19, 39, 38, 13, 49, 25, 27, 30, 22										
Muestras	Grupos									
	1	2	3	4	5	6	7	8	9	10
DisparoReal	21	0	12	34	0	16	19	20	29	1
DisparoSalva	12	8	18	3	0	1	2	1	0	5
EfectoSonido	1	6	0	0	10	22	0	2	1	7
Explosivo	5	5	9	1	3	10	4	4	0	9
Tiempo de ejecución 1.04 segundos										
6 Iteración k.means con 10 clusters de tamaño: 42, 40, 40, 29, 17, 16, 30, 25, 19, 43										
Muestras	Grupos									
	1	2	3	4	5	6	7	8	9	10
DisparoReal	35	21	13	24	12	0	29	0	4	14
DisparoSalva	4	12	19	1	1	1	0	11	0	1
EfectoSonido	2	1	0	2	0	12	1	4	5	22
Explosivo	1	6	8	2	4	3	0	10	10	6
Tiempo de ejecución 0.98 segundos										
7 Iteración k.means con 10 clusters de tamaño: 28 37 13 20 19 47 22 48 45 22										
Muestras	Grupos									
	1	2	3	4	5	6	7	8	9	10
DisparoReal	24	32	0	15	0	13	20	31	16	1
DisparoSalva	3	0	0	1	8	24	1	8	1	4
EfectoSonido	0	1	10	0	6	0	0	3	22	7
Explosivo	1	4	3	4	5	10	1	6	6	10
Tiempo de ejecución 0.98 segundos										
8 Iteración k.means con 10 clusters de tamaño: 15, 29, 20, 29, 32, 30, 37, 16, 32, 61										
Muestras	Grupos									
	1	2	3	4	5	6	7	8	9	10
DisparoReal	13	15	6	1	19	29	12	2	6	39
DisparoSalva	0	10	10	6	0	0	19	0	1	0
EfectoSonido	0	1	4	9	0	0	0	13	19	14
Explosivo	2	3	0	13	1	6	6	4	6	8
Tiempo de ejecución 1.09 segundos										

Tabla C.51: Iteraciones 9 y 10. Kmeans con 10 clusters

9 Iteración k.means con 10 clusters de tamaño: 25 29 22 24 26 14 39 25 18 79										
Muestras	Grupos									
	1	2	3	4	5	6	7	8	9	10
DisparoReal	0	16	16	21	13	12	12	0	17	45
DisparoSalva	2	11	2	2	3	0	18	10	1	1
EfectoSonido	17	1	0	0	2	0	0	6	0	23
Explosivo	6	1	4	1	8	2	9	9	0	10
Tiempo de ejecución 1.92 segundos										
10 Iteración k.means con 10 clusters de tamaño: 8, 19, 20, 17, 50, 29, 30, 25, 22, 60										
Muestras	Grupos									
	1	2	3	4	5	6	7	8	9	10
DisparoReal	20	3	29	0	0	27	16	7	17	33
DisparoSalva	3	0	0	0	6	24	1	7	5	4
EfectoSonido	2	0	1	10	9	1	22	1	3	0
Explosivo	5	3	4	3	9	14	6	4	1	1
Tiempo de ejecución 1.07 segundos										

Apéndice D

Resultados obtenidos por el método k-Medoids

D.1 Resultados obtenidos al implementar K-Medoids sobre las 280 observaciones por bandas de frecuencia promediadas en escala logarítmica

- **K-Medoids con 2 clusters**

Al iterar 10 veces y generando 2 *clusters* se obtuvieron los mismos resultados, la *Tabla D.1* muestra el resultado, y la *Tabla D.2* muestra los tiempos de ejecución por cada iteración

Tabla D.1: Kmedoids con 2 clusters

K-Medoids con 2 clusters de tamaño: 109,171		
Muestras	Grupos	
	1	2
DisparoReal	53	84
DisparoSalva	14	30
EfectoSonido	33	16
Explosivo	9	41

Tabla D.2: Tiempos de ejecución

iteración	Tiempo de ejecución
1	0.24 segundos
2	0.22 segundos
3	0.19 segundos
4	0.25 segundos
5	0.18 segundos
6	0.25 segundos
7	0.25 segundos
8	0.25 segundos
9	0.17 segundos
10	0.20 segundos

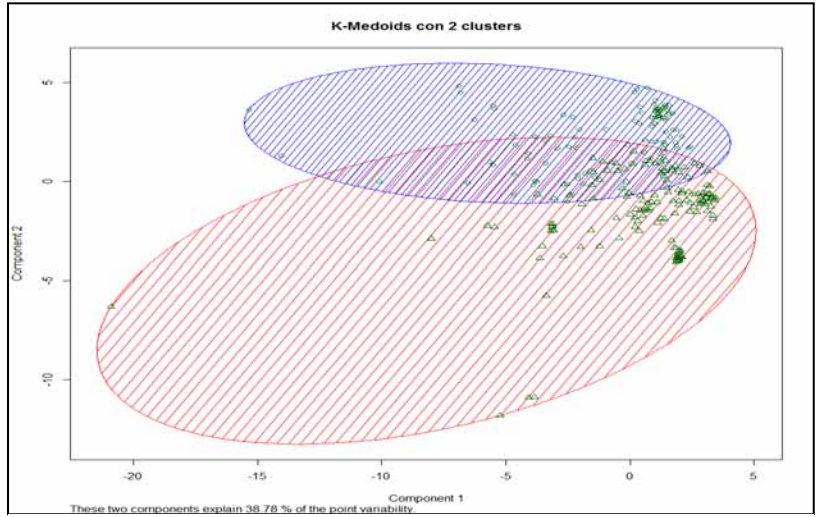


Figura D.1: Kmedoids con 2 clusters

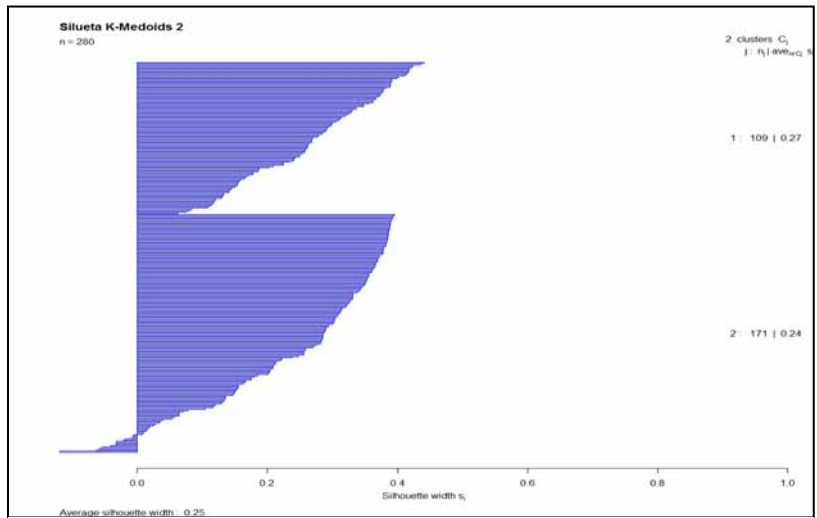


Figura D.2: "Silhouette" sobre Kmedoids con 2 clusters

- **K-Medoids con 3 clusters**

La *Tabla D.3*, expone el resultado al particionar en 3 *cluster* y es el mismo resultado obtenido en cada una de las iteraciones. La *Tabla D.4* muestra los tiempos de ejecución por iteración

Tabla D.3: Kmedoids con 3 clusters

k-medoids con 3 clusters de tamaño: 98,41,141			
Muestras	Grupos		
	1	2	3
DisparoReal	53	35	49
DisparoSalva	13	0	31
EfectoSonido	28	1	20
Explosivos	4	5	41

Tabla D.4: Tiempos de ejecución. Kmedoids con 3 clusters

iteración	Tiempo de ejecución
1	0.25 segundos
2	0.28 segundos
3	0.25 segundos
4	0.30 segundos
5	0.28 segundos
6	0.31 segundos
7	0.25 segundos
8	0.18 segundos
9	0.29 segundos
10	0.26 segundos

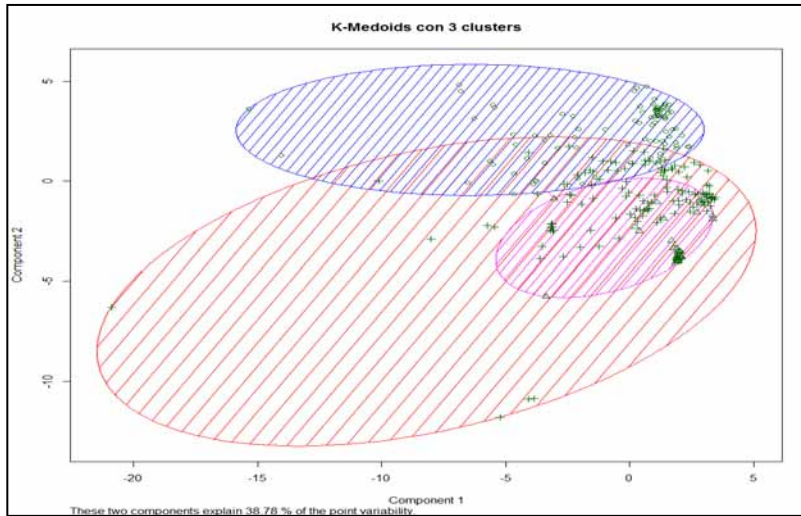


Figura D. 3: Kmedoids con 3 clusters

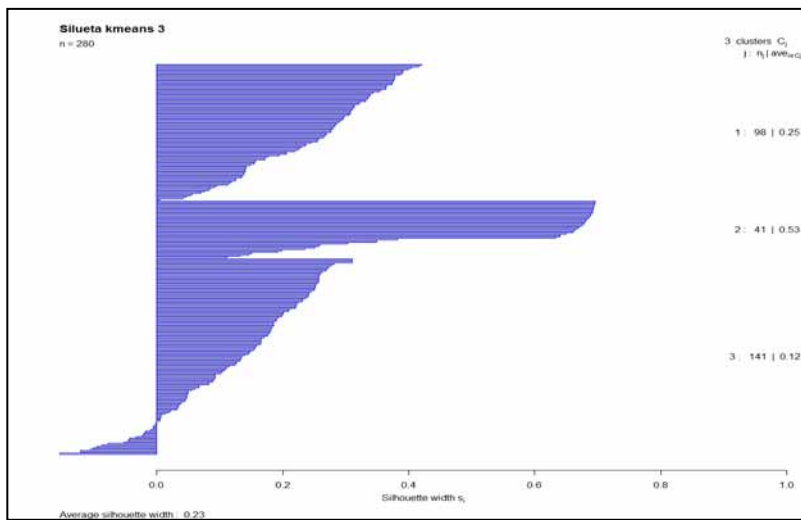


Figura D. 4: "Silhouette" sobre Kmedoids en 3 clusters

- **K-Medoids con 4 clusters**

La *Tabla D.5*, expone el resultado al particionar en 4 *cluster* y es el mismo resultado obtenido en cada una de las iteraciones. La *Tabla D.6* muestra los tiempos de ejecución por iteración

Tabla D.5: Kmedoids con 4 clusters

K-medoids on 4 clusters de tamaño:79,62,94,42				
Muestras	Grupos			
	1	2	3	4
DisparoReal	50	29	25	30
DisparoSalva	9	17	18	0
EfectoSonido	17	0	27	5
Explosivo	3	16	24	7

Tabla D.6: Tiempos de ejecución. Kmedoids con 4 clusters

iteración	Tiempo de ejecución
1	0.30 segundos
2	0.31 segundos
3	0.34 segundos
4	0.29 segundos
5	0.31 segundos
6	0.30 segundos
7	0.31 segundos
8	0.29 segundos
9	0.33 segundos
10	0.33 segundos

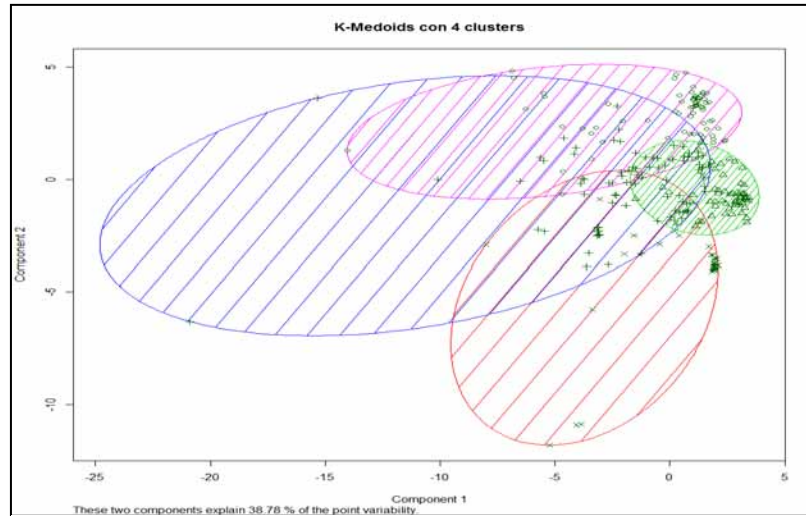


Figura D. 5: Kmedoids con 4 clusters

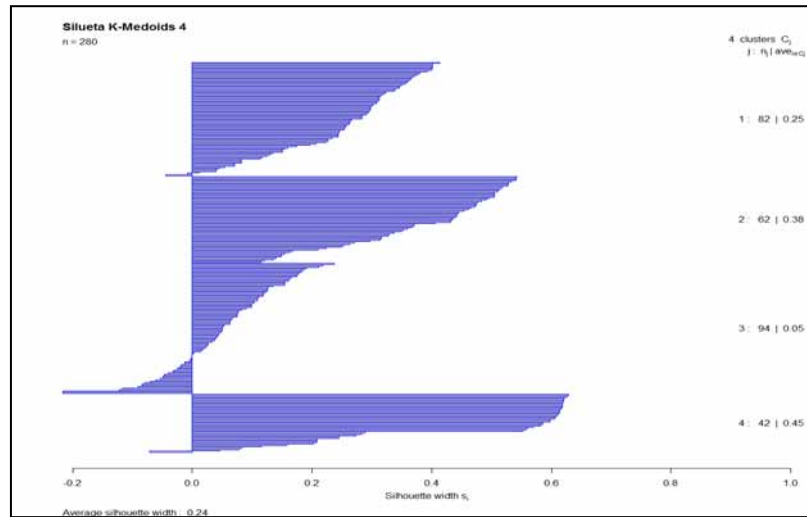


Figura D. 6: "Silhouette" sobre Kmedoids en 4 clusters

- **K-Medoids con 5 clusters**

La *Tabla D.7*, expone el resultado al particionar en 5 *cluster* y es el mismo resultado obtenido en cada una de las iteraciones. La *Tabla D.8* muestra los tiempos de ejecución por iteración.

Tabla D.7: k-medoides con k=5

k-medoids con 5 clusters de tamaño: 81,62,73,33,31					
Muestras	Grupos				
	1	2	3	4	5
DisparoReal	53	29	25	2	28
DisparoSalva	9	17	18	0	0
EfectoSonido	17	0	10	22	0
Explosivo	2	16	20	9	3

Tabla D.8: Tiempos de ejecución. Kmedoids con 5 clusters

iteración	Tiempo de ejecución
1	0.34 segundos
2	0.36 segundos
3	0.33 segundos
4	0.34 segundos
5	0.32 segundos
6	0.35 segundos
7	0.36 segundos
8	0.36 segundos
9	0.32 segundos
10	0.36 segundos

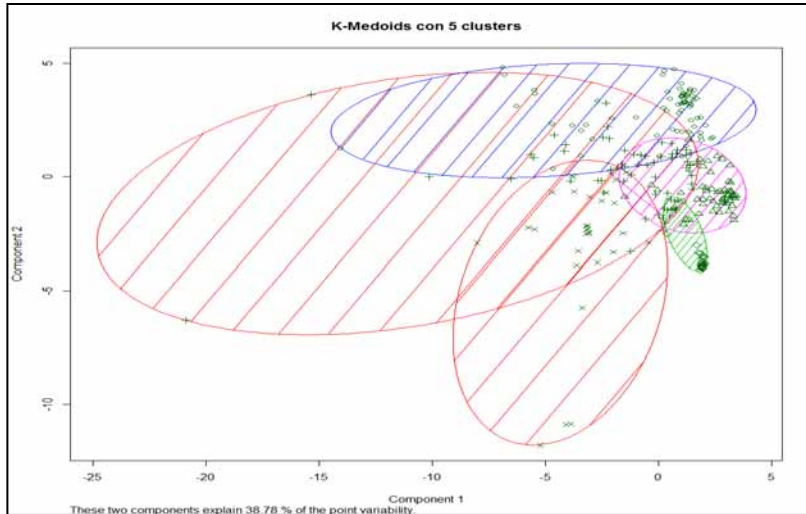


Figura D. 7: medios con 5 clusters

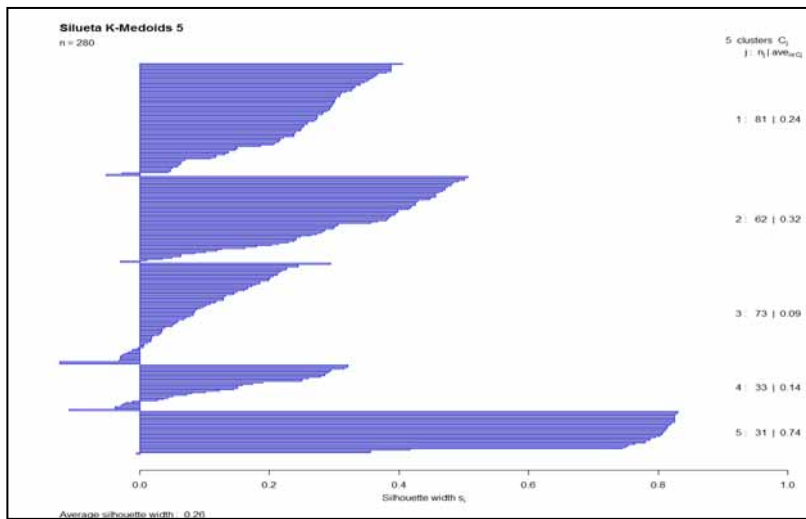


Figura D. 8: "Silhouette" sobre Kmedoids con 5 clusters

- **K-Medoids con 6 clusters**

La *Tabla D.9*, expone el resultado al particionar en 6 *cluster* y es el mismo resultado obtenido en cada una de las iteraciones. La *Tabla D.10* muestra los tiempos de ejecución por iteración.

Tabla D.9: Kmedoids con 6 clusters

k-medoids con 6 clusters de tamaño: 55,62,27,72,33,31						
Muestras	Grupos					
	1	2	3	4	5	6
DisparoReal	28	29	25	25	2	28
DisparoSalva	8	17	2	17	0	0
EfectoSonido	17	0	0	10	22	0
Explosivo	2	16	0	20	9	3

Tabla D.10: Tiempos de ejecución. Kmedoids con 6 clusters

iteración	Tiempo de ejecución
1	0.36 segundos
2	0.36 segundos
3	0.34 segundos
4	0.34 segundos
5	0.32 segundos
6	0.34 segundos
7	0.34 segundos
8	0.94 segundos
9	0.47 segundos
10	0.38 segundos

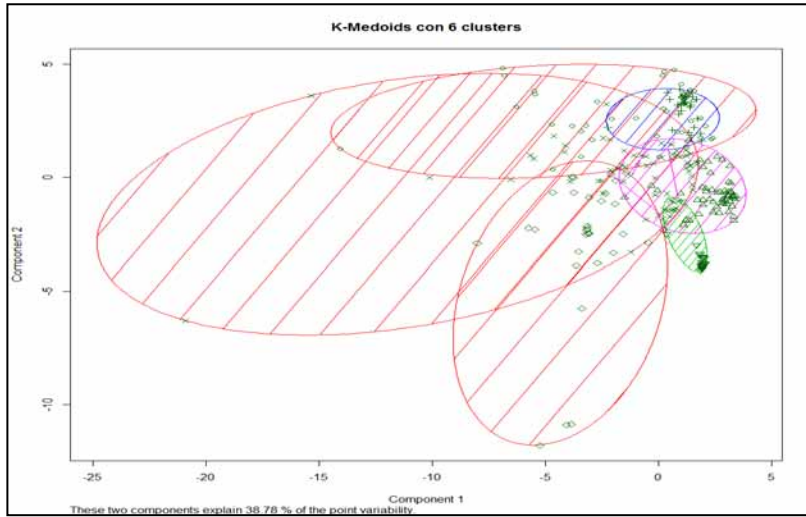


Figura D. 9: Kmedoids con 6 clusters

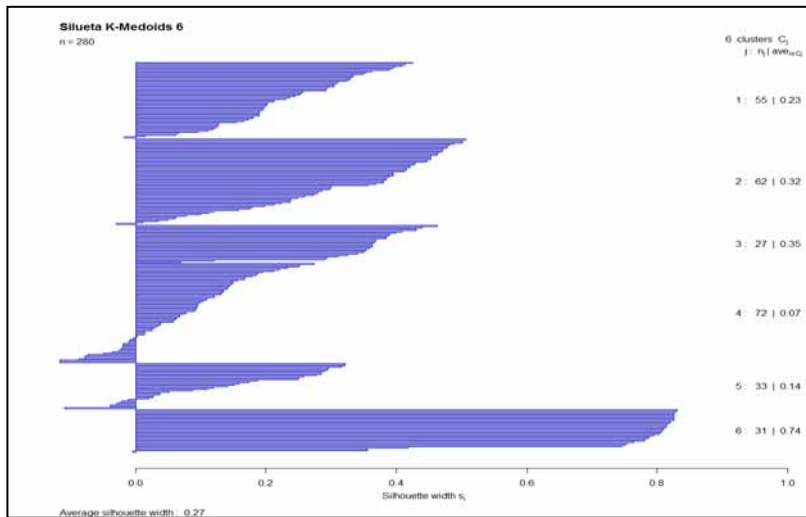


Figura D.10: "Silhouette". Kmedoids con 6 clusters

- **K-Medoids con 7 clusters**

La *Tabla D.11*, expone el resultado al particionar en 7 *cluster* y es el mismo resultado obtenido en cada una de las iteraciones. La *Tabla D.12* muestra los tiempos de ejecución por iteración.

Tabla D.11: Kmedoids con 7 clusters

k-medoids con 7 clusters de tamaño: 49,62,27,62,32,31,17							
Muestras	Grupos						
	1	2	3	4	5	6	7
DisparoReal	27	28	25	12	2	28	15
DisparoSalva	6	17	2	17	0	0	2
EfectoSonido	14	0	0	15	20	0	0
Explosivo	2	17	0	18	10	3	0

Tabla D.12: Tiempos de ejecución. Kmedoids con 7 clusters

iteración	Tiempo de ejecución
1	0.36 segundos
2	0.41 segundos
3	0.36 segundos
4	0.37 segundos
5	0.36 segundos
6	0.36 segundos
7	0.35 segundos
8	0.36 segundos
9	0.35 segundos
10	0.34 segundos

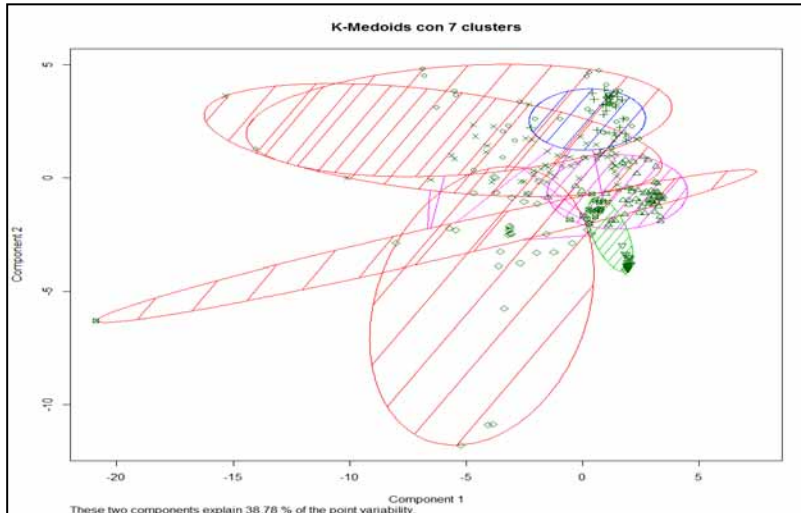


Figura D.11: Kmedoids con 7 clusters

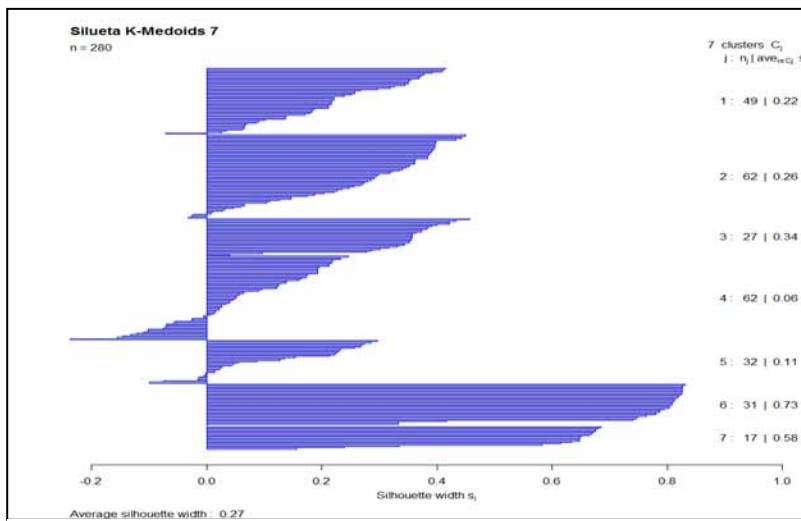


Figura D.12: "Silhouette" sobre Kmedoids con 7 clusters

- **K-Medoids con 8 clusters**

La *Tabla D.13*, expone el resultado al particionar en 8 *cluster* y es el mismo resultado obtenido en cada una de las iteraciones. La *Tabla D.14* muestra los tiempos de ejecución por iteración.

Tabla D.13: Kmedoids con 8 clusters

k-medoids con 8 clusters de tamaño: 49, 32,27,62, 31,32,30,17								
Muestras	Grupos							
	1	2	3	4	5	6	7	8
DisparoReal	27	19	25	12	9	2	28	15
DisparoSalva	6	0	2	17	17	0	0	2
EfectoSonido	14	0	0	15	0	20	0	0
Explosivo	2	13	0	18	5	10	2	0

Tabla D.14: Tiempo de ejecución. Kmedoids con 8 clusters

iteración	Tiempo de ejecución
1	0.39 segundos
2	0.31 segundos
3	0.36 segundos
4	0.36 segundos
5	0.45 segundos
6	0.34 segundos
7	0.37 segundos
8	0.34 segundos
9	0.38 segundos
10	0.38 segundos

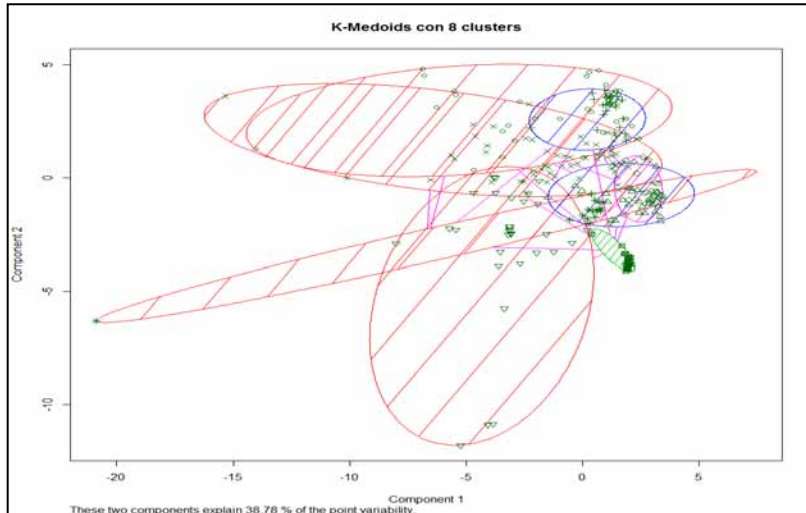


Figura D.13: Kmedoids con 8 clusters

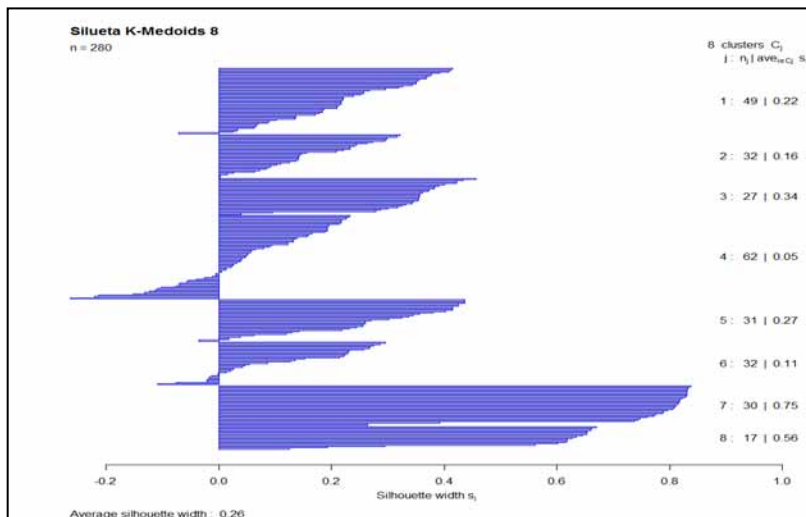


Figura D.14: "Silhouette" sobre Kmedoids con 8 clusters

- **K-Medoids con 9 clusters**

La *Tabla D.15*, expone el resultado al particionar en 9 *cluster* y es el mismo resultado obtenido en cada una de las iteraciones. La *Tabla D.16* muestra los tiempos de ejecución por iteración.

Tabla D.15: Kmedoids con 9 clusters

k-medoids con 9 clusters de tamaño: 49,32,27,43,32,28,30,16,23									
Muestras	Grupos								
	1	2	3	4	5	6	7	8	9
DisparoReal	27	19	25	13	8	2	28	15	0
DisparoSalva	8	0	2	12	17	0	0	1	4
EfectoSonido	12	0	0	3	0	17	0	0	17
Explosivo	2	13	0	15	7	9	2	0	2

Tabla D.16: Tiempos de ejecución. Kmedoids con 9 clusters

iteración	Tiempo de ejecución
1	0.38 segundos
2	0.37 segundos
3	0.36 segundos
4	0.49 segundos
5	0.36 segundos
6	0.34 segundos
7	0.35 segundos
8	0.50 segundos
9	0.34 segundos
10	0.53 segundos

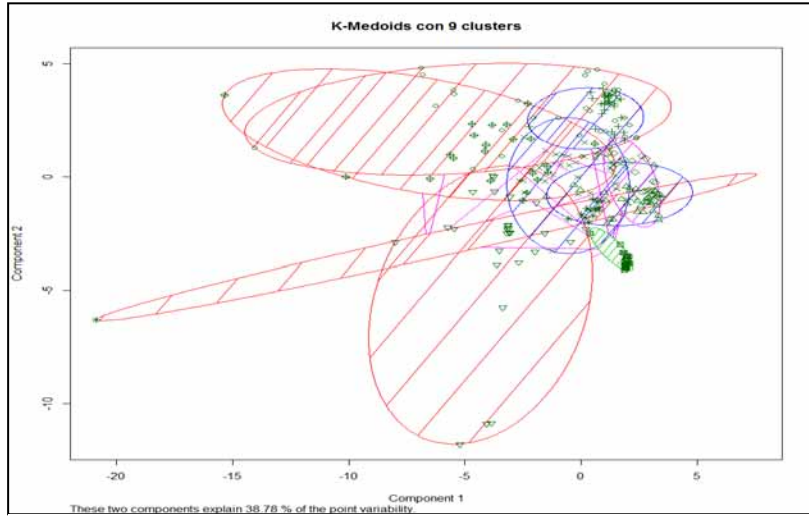


Figura D.15: Kmedoids con 9 clusters

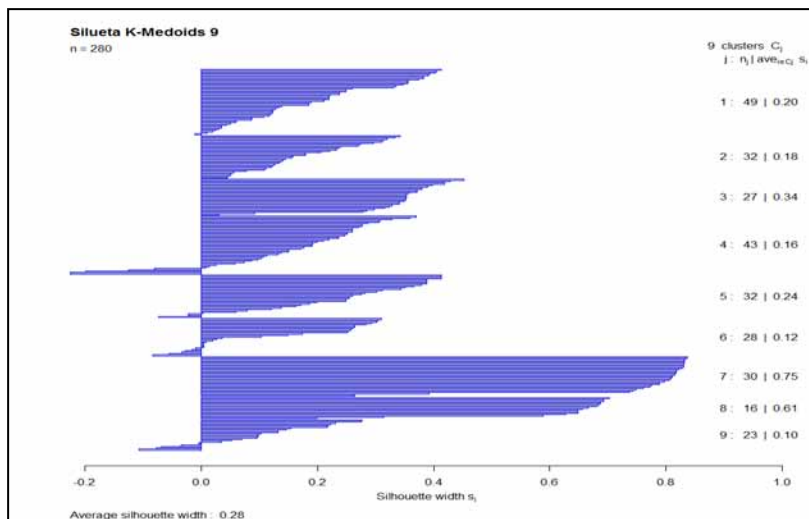


Figura D.16: "Silhouette" sobre Kmedoids con 9 clusters

- **K-Medoids con 10 clusters**

La *Tabla D.17*, expone el resultado al particionar en 10 *cluster* y es el mismo resultado obtenido en cada una de las iteraciones. La *Tabla D.18* muestra los tiempos de ejecución por iteración.

Tabla D.17: Kmedoids con 10 clusters

k-medoids con 10 clusters de tamaño: 49, 29,26,28, 22,30,31,26,16,22										
Muestras	Grupos									
	1	2	3	4	5	6	7	8	9	10
DisparoReal	27	17	25	3	13	2	28	7	15	0
DisparoSalva	8	0	1	9	4	0	0	17	1	4
EfectoSonido	12	0	0	2	2	16	0	0	0	17
Explosivo	2	12	0	14	3	9	2	7	0	1

Tabla D.18: Tiempos de ejecución. Kmedoids con 10 clusters

iteración	Tiempo de ejecución
1	0.40 segundos
2	0.42 segundos
3	0.56 segundos
4	0.43 segundos
5	0.39 segundos
6	0.42 segundos
7	0.39 segundos
8	0.40 segundos
9	0.40 segundos
10	0.42 segundos

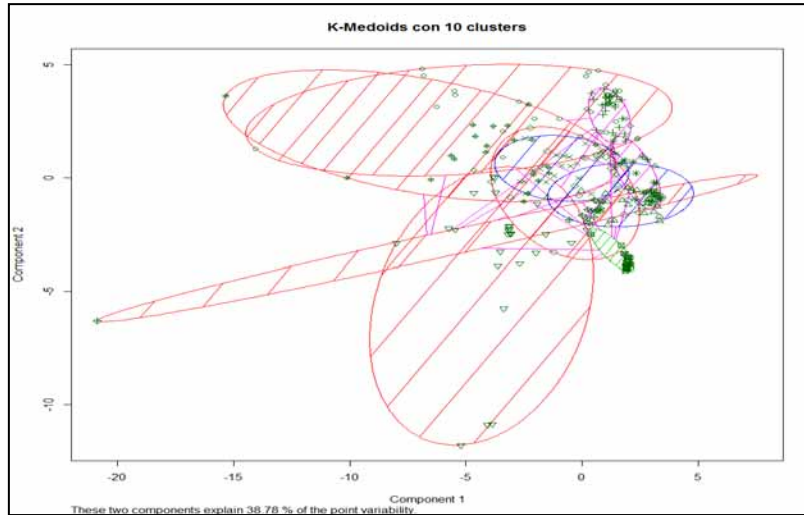


Figura D.17: Kmedoids con 10 clusters

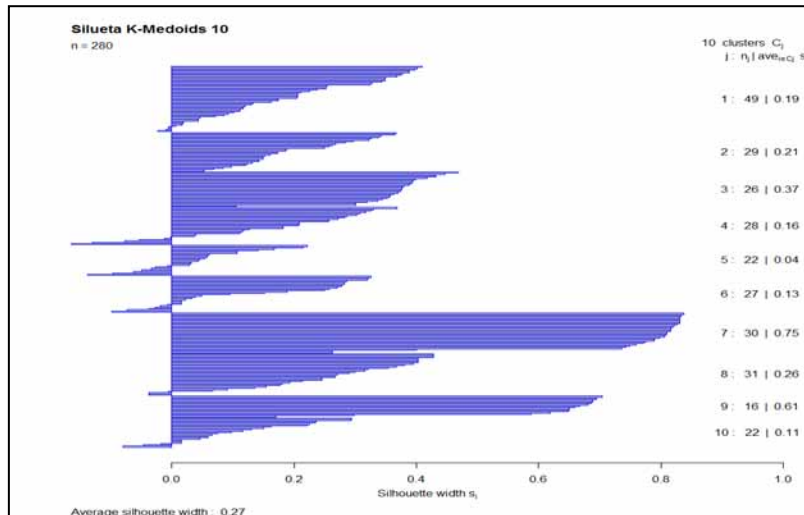


Figura D.18; "Silhouette" sobre Kmedoids con 10 clusters

D.2 Resultados obtenidos al implementar K-Medoids sobre 301 observaciones promediadas por bandas de frecuencia de 50 en 50 Hz

- **K-Medoids con 2 clusters**

La *Tabla D.19*, expone el resultado al particionar en 2 *cluster* y es el mismo resultado obtenido en cada una de las iteraciones. La *Tabla D.20* muestra los tiempos de ejecución por iteración.

Tabla D.19: Kmedoids con 2 clusters

k-medoids con 2 clusters de tamaño: 186,115		
Muestras	Grupos	
	1	2
DisparoReal	105	47
DisparoSalva	44	6
EfectoSonido	7	42
Explosivo	30	20

Tabla D.20: Tiempos de ejecución. Kmedoids con 2 clusters

iteración	Tiempo de ejecución
1	0.58 segundos
2	0.32 segundos
3	0.34 segundos
4	0.34 segundos
5	0.34 segundos
6	0.39 segundos
7	0.35 segundos
8	0.36 segundos
9	0.37 segundos
10	0.39 segundos

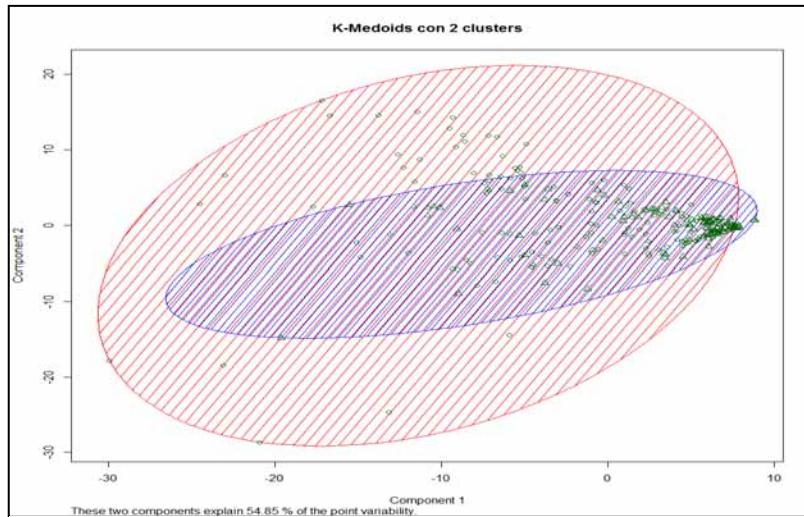


Figura D.19: Kmedoids con 2 clusters

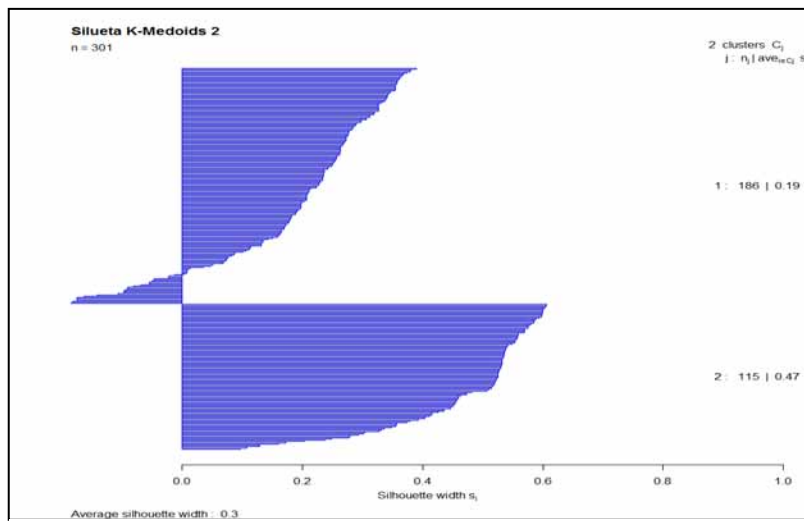


Figura D.20: índice "Silhouette" sobre Kmedoids con 2 clusters

- **K-Medoids con 3 clusters**

La *Tabla D.21*, expone el resultado al particionar en 3 *cluster* y es el mismo resultado obtenido en cada una de las iteraciones. La *Tabla D.22* muestra los tiempos de ejecución por iteración

Tabla D.21: K-Medoids con 3 clusters

k-medoids con 3 clusters de tamaño: 156,45,100			
Muestras	Grupos		
	1	2	3
DisparoReal	102	3	47
DisparoSalva	35	14	1
EfectoSonido	2	11	36
Explosivo	17	17	16

Tabla D.22: Tiempos de ejecución. K-Medoids con 3 clusters

iteración	Tiempo de ejecución
1	0.45 segundos
2	0.34 segundos
3	0.36 segundos
4	0.36 segundos
5	0.38 segundos
6	0.37 segundos
7	0.36 segundos
8	0.38 segundos
9	0.77 segundos
10	0.34 segundos

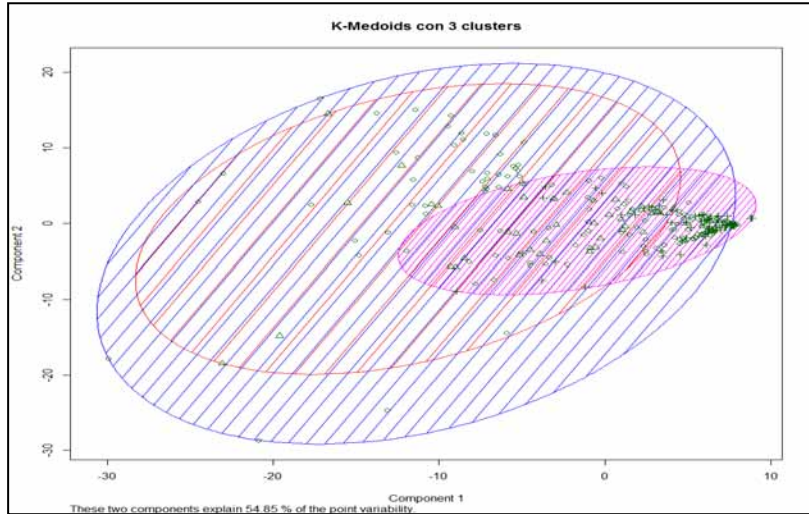


Figura D.21: Kmedoids con 3 clusters

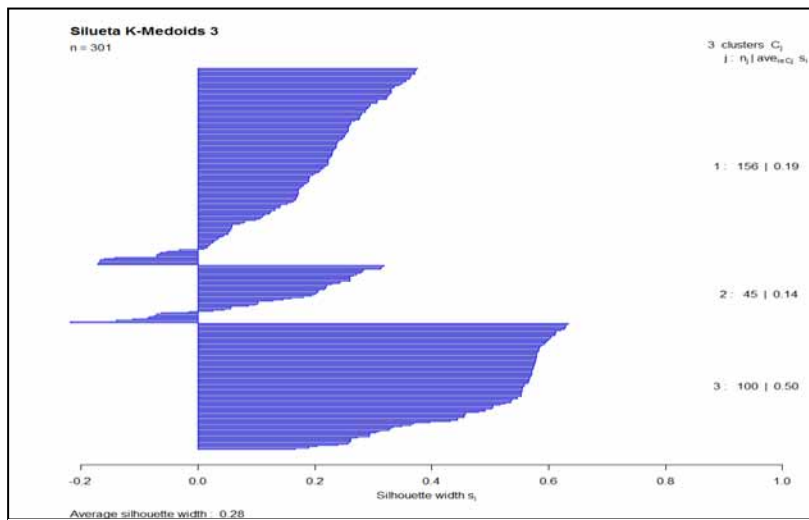


Figura D.22: "Silhouette" sobre Kmedoids con 3 clusters

- **K-Medoids con 4 clusters**

La *Tabla D.23*, expone el resultado al particionar en 4 *cluster* y es el mismo resultado obtenido en cada una de las iteraciones. La *Tabla D.24* muestra los tiempos de ejecución por iteración

Tabla D.23: Kmedoids con 4 clusters

1 iteración k-medoids con 4 clusters de tamaño: 140,48,49,64				
Muestras	Grupos			
	1	2	3	4
DisparoReal	92	41	2	17
DisparoSalva	32	0	16	2
EfectoSonido	2	1	11	35
Explosivo	14	6	20	10

Tabla D.24: Tiempo de ejecución. Kmedoids con 4 clusters

iteración	Tiempo de ejecución
1	0.36 segundos
2	0.26 segundos
3	0.34 segundos
4	0.39 segundos
5	0.38 segundos
6	0.36 segundos
7	0.34 segundos
8	0.38 segundos
9	0.37 segundos
10	0.36 segundos

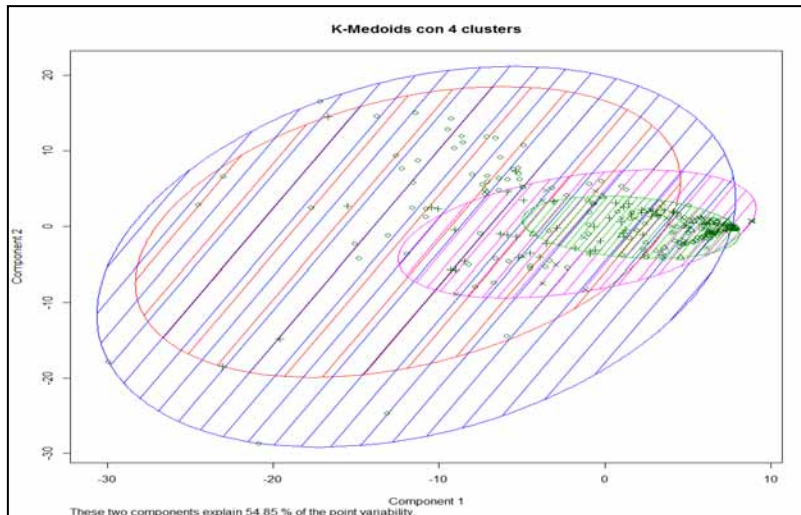


Figura D.23: Kmedoids con 4 clusters

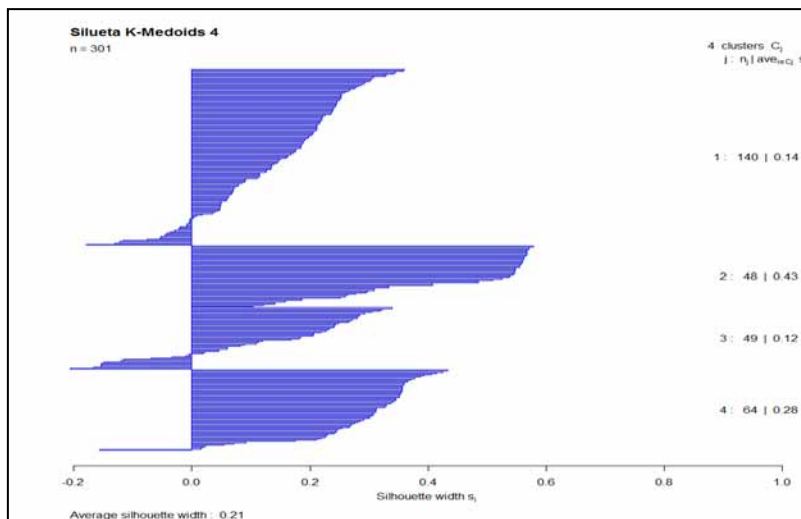


Figura D.24: "Silhouette" sobre kMedoids con 4 clusters

- **K-Medoids con 5 clusters**

La *Tabla D.25*, expone el resultado al particionar en 5 cluster y es el mismo resultado obtenido en cada una de las iteraciones. La *Tabla D.26* muestra los tiempos de ejecución por iteración

Tabla D.25: Kmedoids con 5 clusters

k-medoids con 5 clusters de tamaño: 49,45,102,63,42					
Muestras	Grupos				
	1	2	3	4	5
DisparoReal	40	38	57	17	0
DisparoSalva	8	0	27	1	14
EfectoSonido	1	1	1	35	11
Explosivo	0	6	17	10	17

Tabla D.26: Tiempos de ejecución. Kmedoids con 5 clusters

iteración	Tiempo de ejecución
1	0.41 segundos
2	0.40 segundos
3	0.48 segundos
4	0.38 segundos
5	0.39 segundos
6	0.38 segundos
7	0.35 segundos
8	0.36 segundos
9	0.39 segundos
10	0.48 segundos

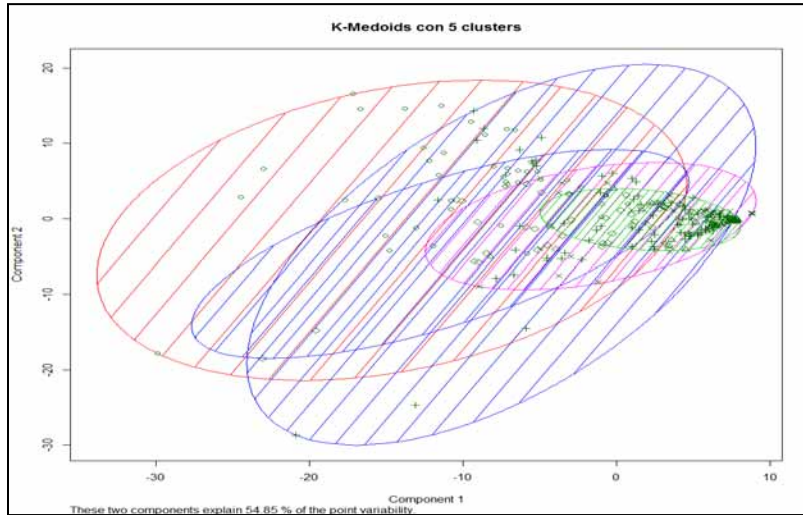


Figura D.25: Kmedoids con 5 clusters

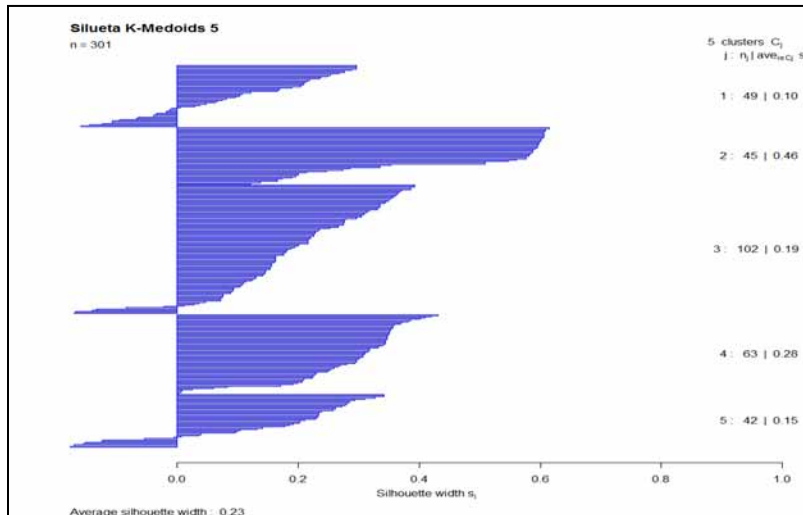


Figura D.26: "Silhouette" sobre Kmedoids con 5 clusters

- **K-Medoids con 6 clusters**

La *Tabla D.27*, expone el resultado al particionar en 6 cluster y es el mismo resultado obtenido en cada una de las iteraciones. La *Tabla D.28* muestra los tiempos de ejecución por iteración

Tabla D.27: Kmedoids con 6 clusters

k-medoids con 6 clusters de tamaño: 37,41,85,33,63,42						
Muestras	Grupos					
	1	2	3	4	5	6
DisparoReal	32	35	46	22	17	0
DisparoSalva	5	0	26	4	1	14
EfectoSonido	0	1	0	2	35	11
Explosivo	0	5	13	5	10	17

Tabla D.28: Tiempos de ejecución por cada iteración. Kmedoids con 6 clusters

iteración	Tiempo de ejecución
1	0.69 segundos
2	0.45 segundos
3	0.42 segundos
4	0.40 segundos
5	0.41 segundos
6	0.39 segundos
7	0.43 segundos
8	0.47 segundos
9	0.41 segundos
10	0.40 segundos

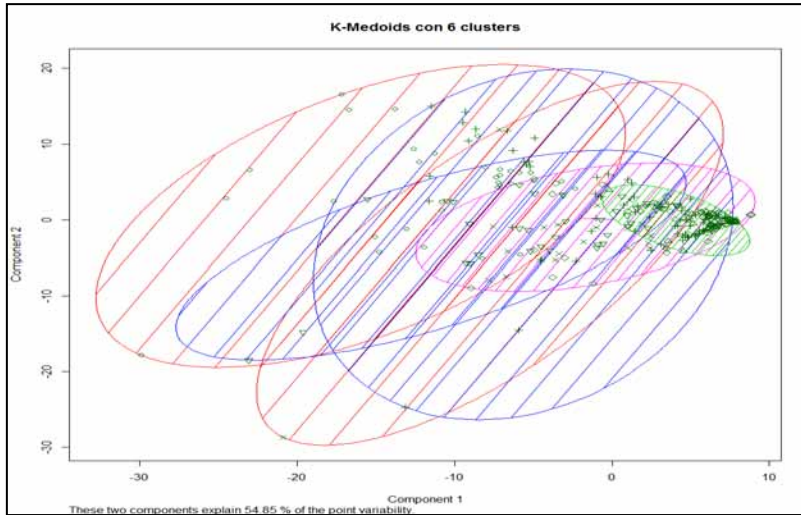


Figura D.27: Kmedoids con 6 clusters

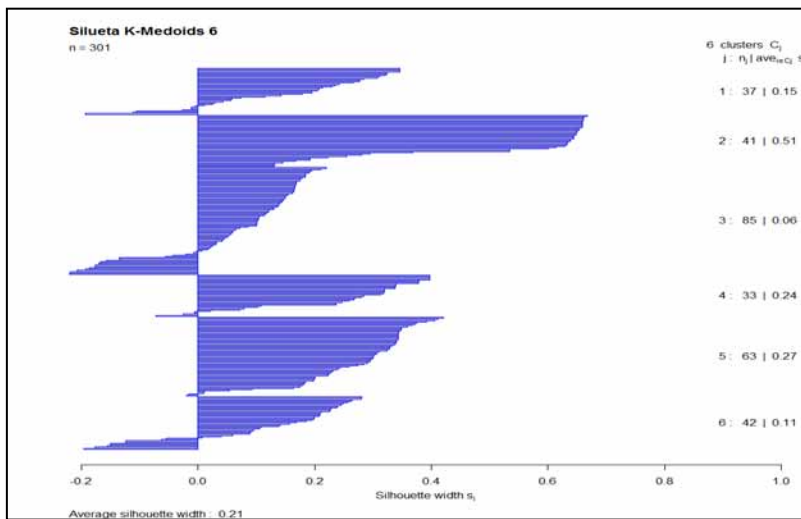


Figura D.28: "Silhouette" sobre Kmedoids con 6 clusters

- **K-Medoids con 7 clusters**

La *Tabla D.29*, expone el resultado al particionar en 7 *cluster* y es el mismo resultado obtenido en cada una de las iteraciones. La *Tabla D.30* muestra los tiempos de ejecución por iteración

Tabla D.29: Kmedoids con 7 clusters

k-medoids con 7 clusters de tamaño: 39,37,72,39,11,39,64							
Muestras	Grupos						
	1	2	3	4	5	6	7
DisparoReal	32	32	38	26	6	1	17
DisparoSalva	7	0	22	6	0	14	1
EfectoSonido	0	1	0	2	0	11	35
Explosivo	0	4	12	5	5	13	11

Tabla D.30: Tiempos de ejecución por cada iteración. Kmedoids con 7 clusters

iteración	Tiempo de ejecución
1	0.40 segundos
2	0.42 segundos
3	0.42 segundos
4	0.44 segundos
5	0.44 segundos
6	0.42 segundos
7	0.56 segundos
8	0.42 segundos
9	0.63 segundos
10	0.41 segundos

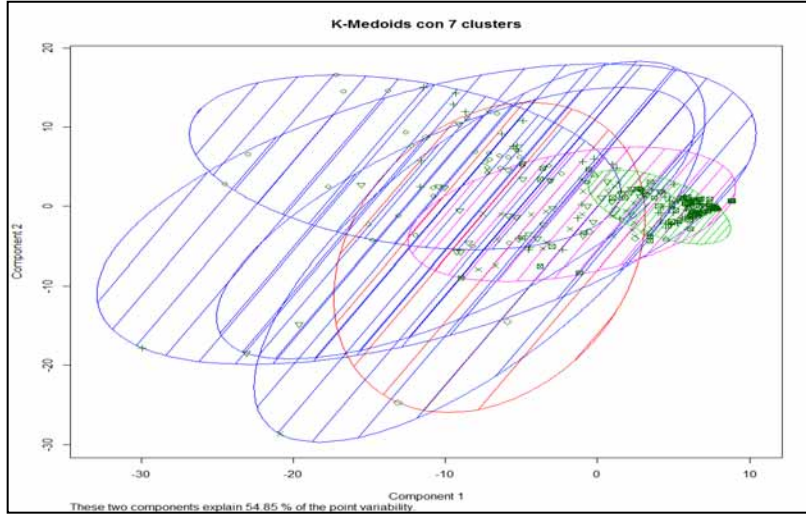


Figura D.29: medios con 7 clusters

- **K-Medoids con 8 clusters**

La *Tabla D.31*, expone el resultado al particionar en 8 cluster y es el mismo resultado obtenido en cada una de las iteraciones. La *Tabla D.32* muestra los tiempos de ejecución por iteración

Tabla D.31: Kmedoids con 8 clusters

k-medoids con 8 clusters de tamaño: 19,37,40,72,37,11,35,50								
Muestras	Grupos							
	1	2	3	4	5	6	7	8
DisparoReal	2	30	34	38	24	6	1	17
DisparoSalva	1	7	0	22	6	0	13	1
EfectoSonido	14	0	1	0	2	0	8	24
Explosivo	2	0	5	12	5	5	13	8

Tabla D.32: Tiempos de ejecución por cada iteración. Kmedoids con 8 clusters

iteración	Tiempo de ejecución
1	0.47 segundos
2	0.53 segundos
3	0.43 segundos
4	0.44 segundos
5	0.47 segundos
6	0.46 segundos
7	0.47 segundos
8	0.45 segundos
9	0.49 segundos
10	0.46 segundos

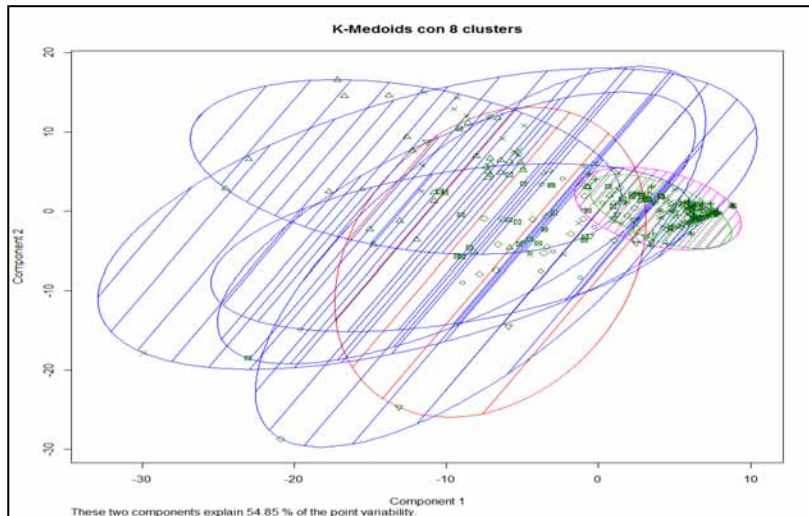


Figura D.30: Kmedoids con 8 clusters

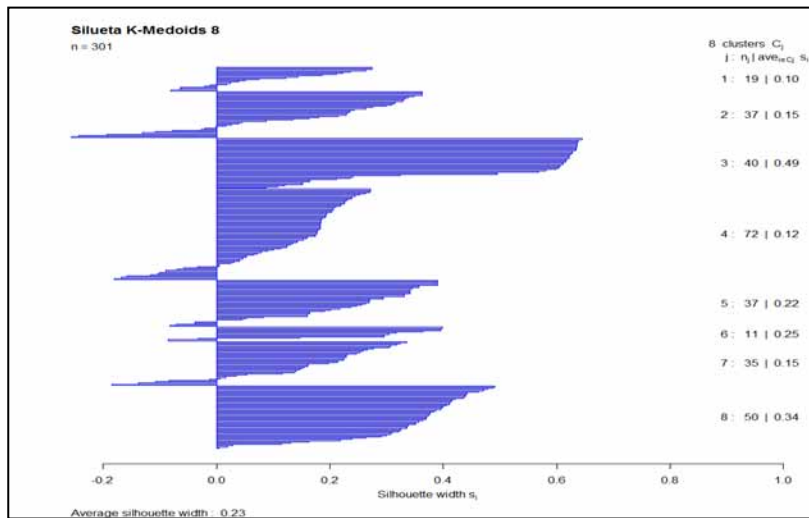


Figura D.31: "Silhouette" sobre Kmedoids con 8 clusters

- **K-Medoids con 9 clusters**

La *Tabla D.33*, expone el resultado al particionar en 9 cluster y es el mismo resultado obtenido en cada una de las iteraciones. La *Tabla D.34* muestra los tiempos de ejecución por iteración

Tabla D.33: Kmedoids con 9 cluster

k-medoids con 9 clusters de tamaño: 19,39,37,30,46,10,36,50,34									
Muestras	Grupos								
	1	2	3	4	5	6	7	8	9
DisparoReal	2	32	24	11	29	6	2	17	29
DisparoSalva	1	7	2	18	8	0	13	1	0
EfectoSonido	14	0	0	0	2	0	8	24	1
Explosivo	2	0	11	1	7	4	13	8	4

Tabla D.34: Tiempos de ejecución por cada iteración. Kmedoids con 9 clusters

iteración	Tiempo de ejecución
1	0.32 segundos
2	0.35 segundos
3	0.36 segundos
4	0.39 segundos
5	0.36 segundos
6	0.38 segundos
7	0.35 segundos
8	0.39 segundos
9	0.37 segundos
10	0.37 segundos

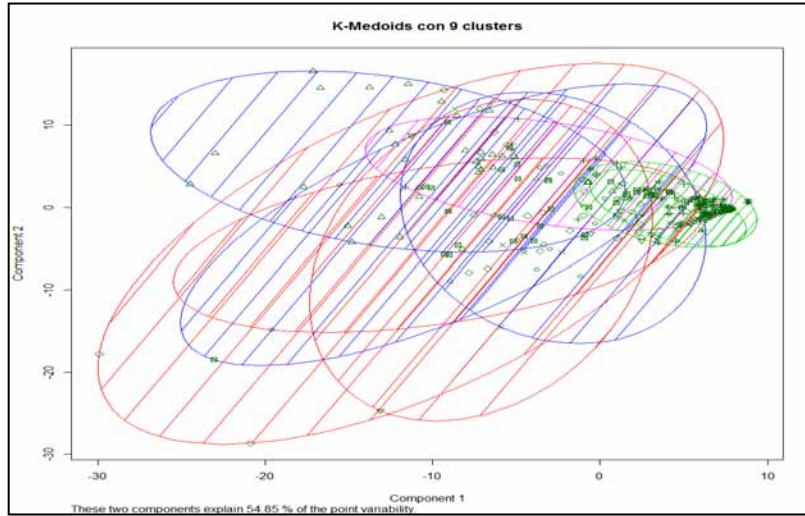


Figura D.32: Kmedoids con 9 clusters

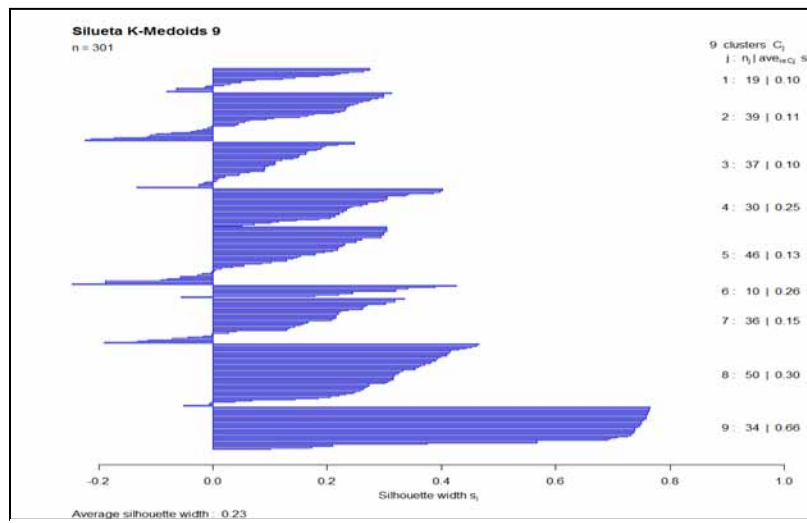


Figura D.33: "Silouette" sobre KMedoids con 9 clusters

- **K-Medoids con 10 clusters**

La *Tabla D.35*, expone el resultado al particionar en 10 *cluster* y es el mismo resultado obtenido en cada una de las iteraciones. La *Tabla D.36* muestra los tiempos de ejecución por iteración.

Tabla D.35: Kmedoids con 10 clusters

k-medoids con 10 clusters de tamaño: 21,33,31,10,36,28,41,50,34,17										
Muestras	Grupos									
	1	2	3	4	5	6	7	8	9	10
DisparoReal	20	26	18	6	2	9	25	17	29	0
DisparoSalva	1	7	2	0	13	18	7	1	0	1
EfectoSonido	0	0	0	0	8	0	2	24	14	14
Explosivo	0	0	11	4	13	1	7	8	4	2

Tabla D.36: Tiempos de ejecución por cada iteración. Kmedoids con 10 clusters

iteración	Tiempo de ejecución
1	0.40 segundos
2	0.44 segundos
3	0.40 segundos
4	0.40 segundos
5	0.42 segundos
6	0.42 segundos
7	0.38 segundos
8	0.39 segundos
9	0.56 segundos
10	0.56 segundos

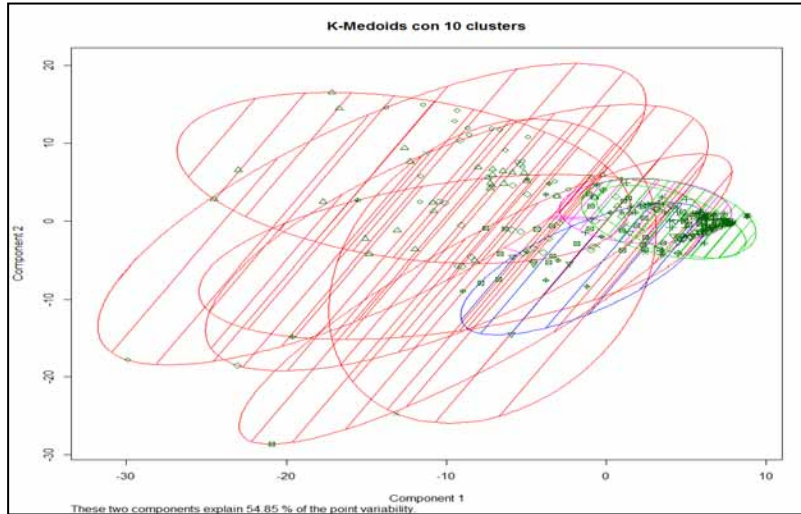


Figura D.34: Kmedoids con 10 clusters

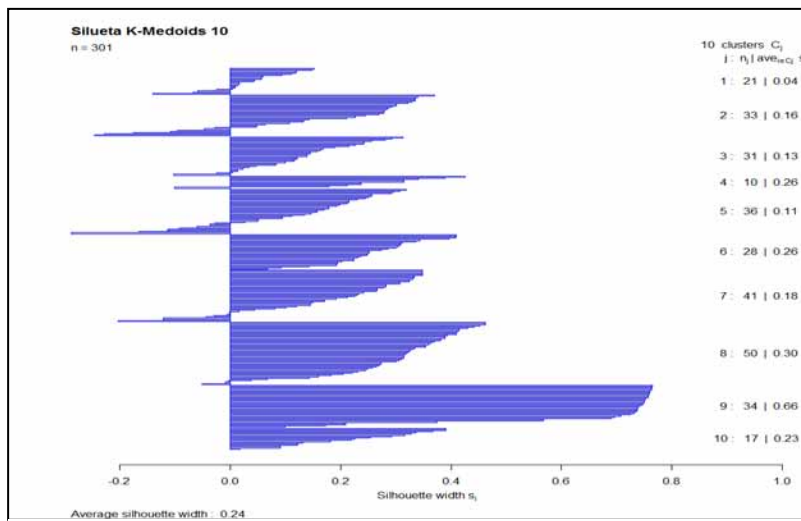


Figura D.35: "Silhouette" sobre Kmedoids con 10 clusters

Apéndice E

Resultados obtenidos al implementar el método Jerárquico

E.1 Resultados obtenidos al implementar Jerárquico sobre las 280 observaciones por bandas de frecuencia promediadas en escala logarítmica

- **Jerárquico con 2 grupos**

La *Tabla E.1*, expone el resultado al aglomerar en 2 *cluster* y es el mismo resultado obtenido en cada una de las iteraciones. La *Tabla E.2* muestra los tiempos de ejecución por iteración.

Tabla E.1: Jerárquico con 2 clusters

Hierarchical con 2 clusters de tamaño: 86,194		
Muestras	Grupos	
	1	2
DisparoReal	46	91
DisparoSalva	9	35
EfectoSonido	28	21
Explosivo	3	47

Tabla E.2: Tiempos de ejecución por cada iteración. Jerárquico con 2 clusters

iteración	Tiempo de ejecución
1	0.41 segundos
2	0.06 segundos
3	0.09 segundos
4	0.10 segundos
5	0.10 segundos
6	0.09 segundos
7	0.11 segundos
8	0.11 segundos
9	0.09 segundos
10	0.09 segundos

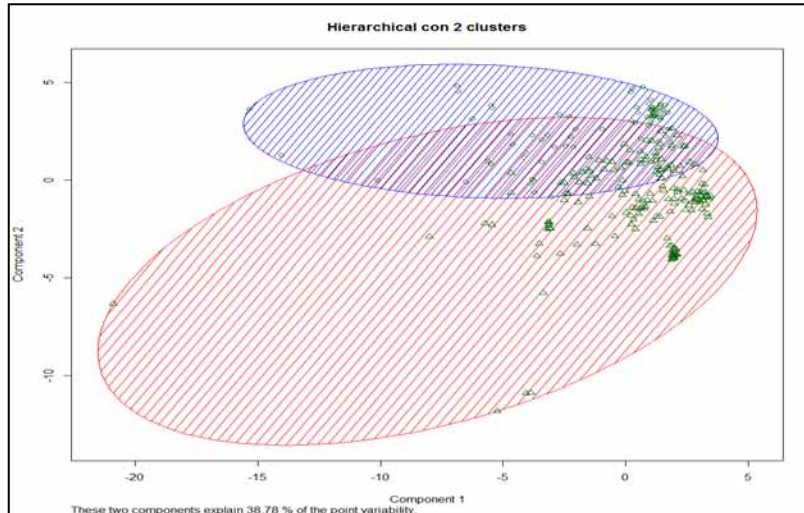


Figura E.1: Jerárquico con dos clusters

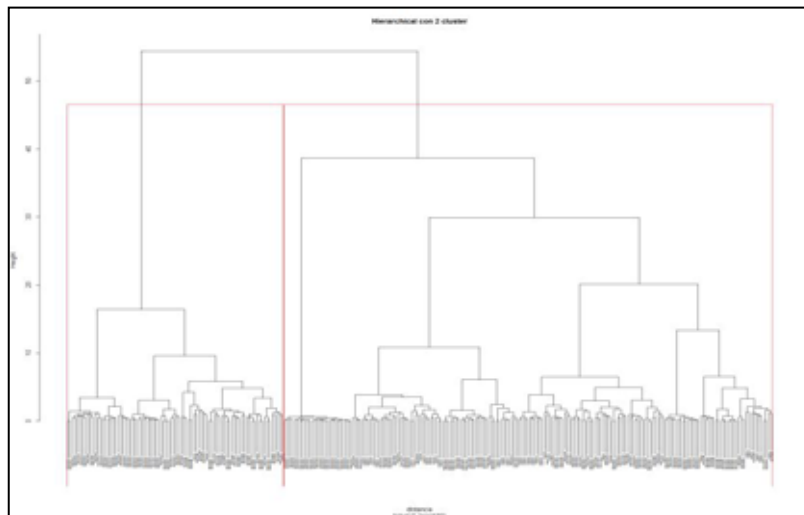


Figura E.2: Dendograma. Jerárquico con 2 clusters

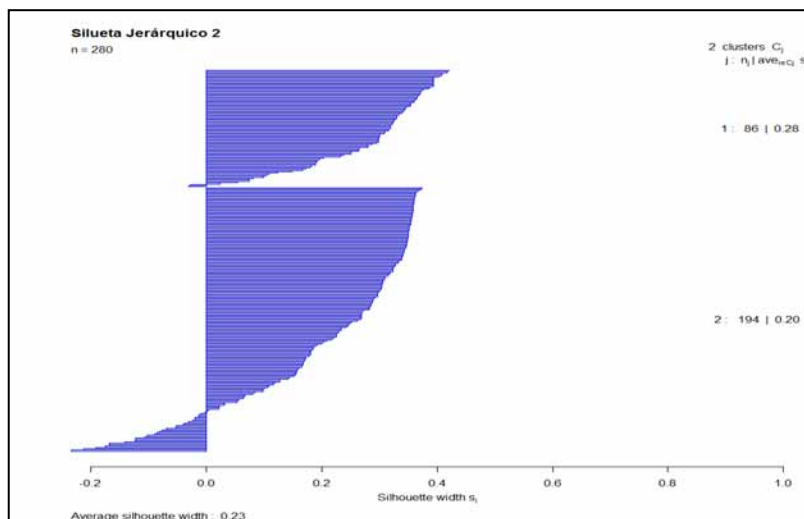


Figura E.3: "Silhouette" sobre jerárquico con 2 clusters

- **Jerárquico con 3 clusters**

La tabla E.3, expone el resultado al aglomerar en 3 *cluster* y es el mismo resultado obtenido en cada una de las iteraciones. La tabla E.4 muestra los tiempos de ejecución por iteración.

Tabla E.3: Jerárquico con 3 clusters

Hierarchical con 3 clusters de tamaño: 86,167,27			
Muestras	Grupos		
	1	2	3
DisparoReal	46	64	27
DisparoSalva	9	35	0
EfectoSonido	28	21	0
Explosivo	3	47	0

Tabla E.4: Tiempos de ejecución por cada iteración. Jerárquico con 3 clusters

iteración	Tiempo de ejecución
1	0.31 segundos
2	0.11 segundos
3	0.11 segundos
4	0.11 segundos
5	0.11 segundos
6	0.14 segundos
7	0.12 segundos
8	0.11 segundos
9	0.12 segundos
10	0.12 segundos

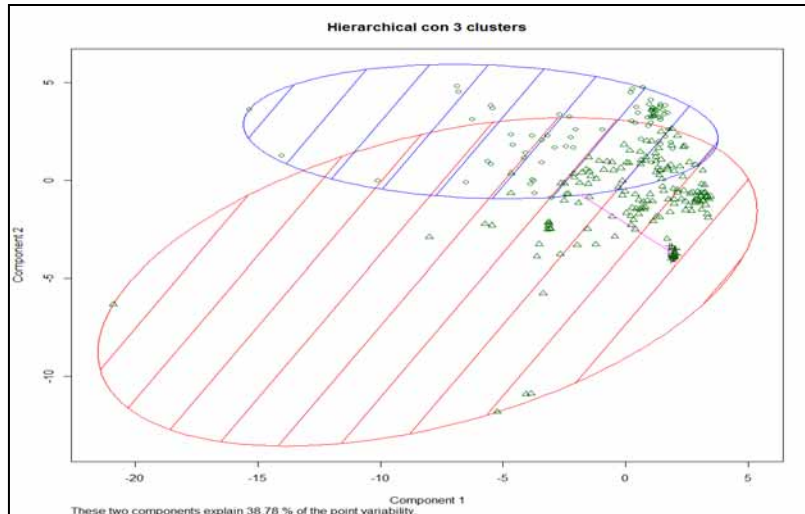


Figura E.4: Jerárquico con 3 clusters

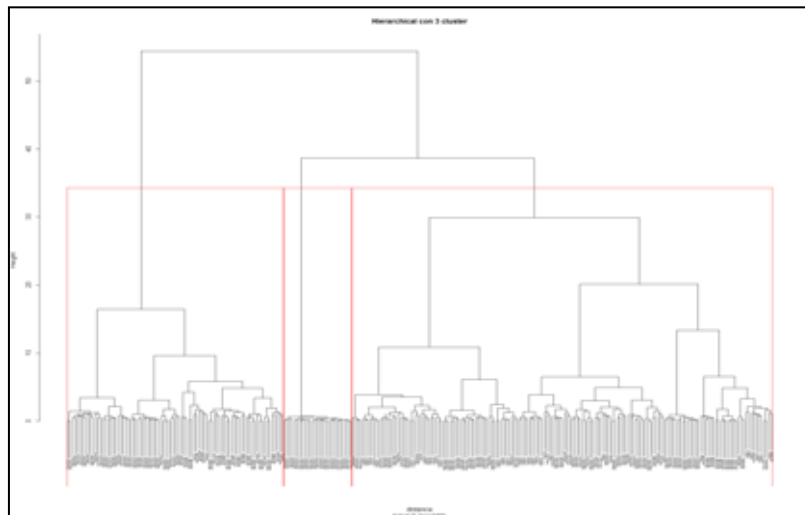


Figura E.5: Dendograma. Jerárquico con 3 clusters

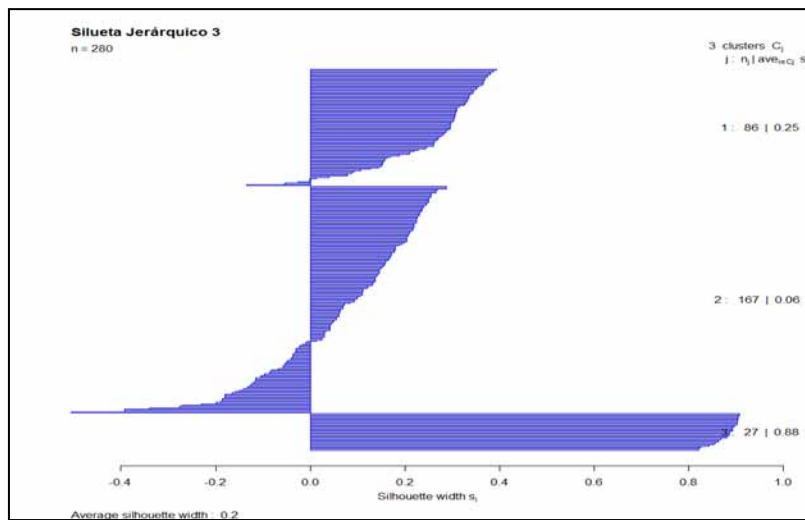


Figura E.6: “Silhouette” sobre Jerárquico con 3 clusters

- **Jerárquico con 4 clusters**

La *Tabla E.5*, expone el resultado al aglomerar en 4 *cluster* y es el mismo resultado obtenido en cada una de las iteraciones. La *Tabla E.6* muestra los tiempos de ejecución por iteración.

Tabla E.5: Jerárquico con 4 clusters

Hierarchical con 4 clusters de tamaño: 86,66,101,27				
Muestras	Grupos			
	1	2	3	4
DisparoReal	46	25	39	27
DisparoSalva	9	17	18	0
EfectoSonido	28	0	21	0
Explosivo	3	24	23	0

Tabla E.6: Tiempos de ejecución por cada iteración. Jerárquico con 4 clusters

iteración	Tiempo de ejecución
1	0.05 segundos
2	0.14 segundos
3	0.13 segundos
4	0.13segundos
5	0.13 segundos
6	0.11 segundos
7	0.14 segundos
8	0.11 segundos
9	0.12 segundos
10	0.12 segundos

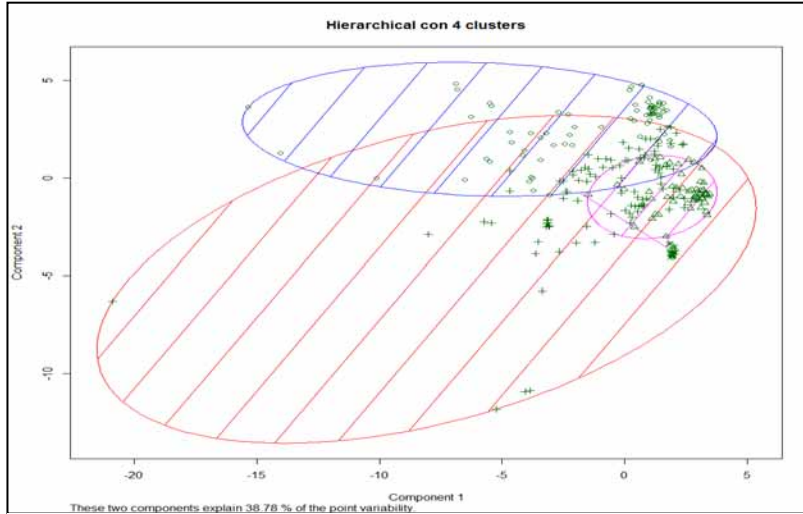


Figura E.7: Jerárquico con 4 clusters

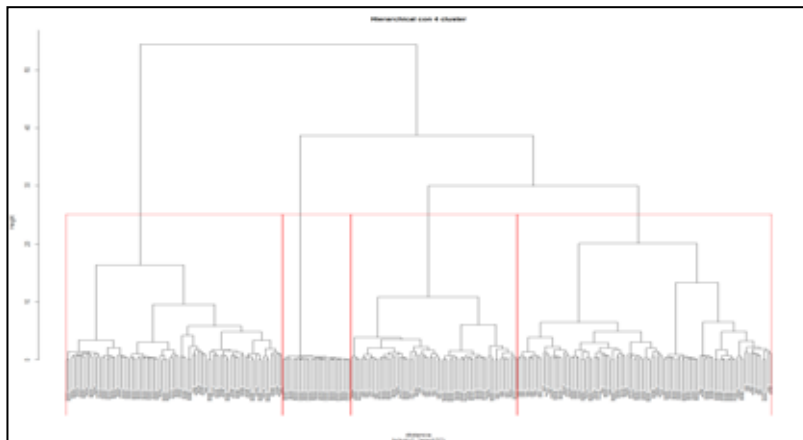


Figura E.8: Dendrograma. Jerárquico con 4 clusters

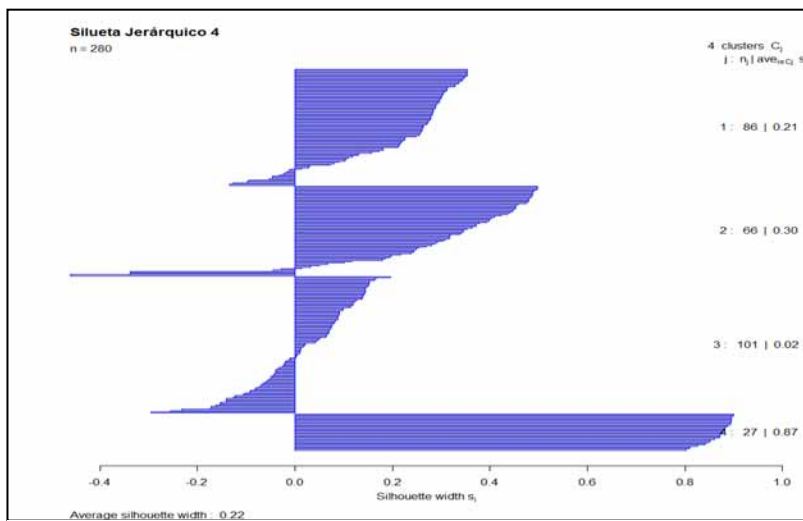


Figura E.9: "Silhouette". Jerárquico con 4 clusters

- **Jerárquico con 5 clusters**

La *Tabla E.7*, expone el resultado al aglomerar en 5 cluster y es el mismo resultado obtenido en cada una de las iteraciones. La *Tabla E.8* muestra los tiempos de ejecución por iteración.

Tabla E.7: Jerárquico con 5 clusters

Hierarchical con 5 clusters de tamaño: 86,66,58,43,27					
Muestras	Grupos				
	1	2	3	4	5
DisparoReal	46	25	23	16	27
DisparoSalva	9	17	18	0	0
EfectoSonido	28	0	2	19	0
Explosivo	3	24	15	8	0

Tabla E.8: Tiempos de ejecución por cada iteración. Jerárquico con 5 clusters

iteración	Tiempo de ejecución
1	0.05 segundos
2	0.07 segundos
3	0.11 segundos
4	0.11 segundos
5	0.13 segundos
6	0.12 segundos
7	0.11 segundos
8	0.13 segundos
9	0.11 segundos
10	0.14 segundos

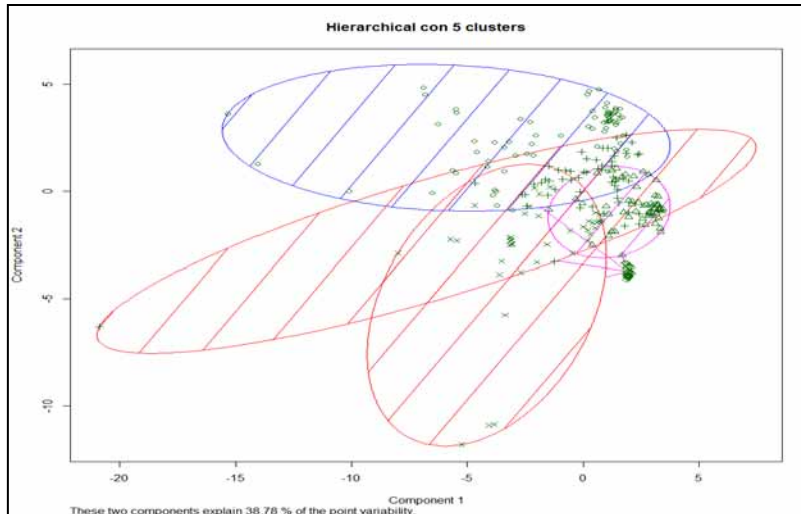


Figura E.10: Jerárquico con 5 clusters

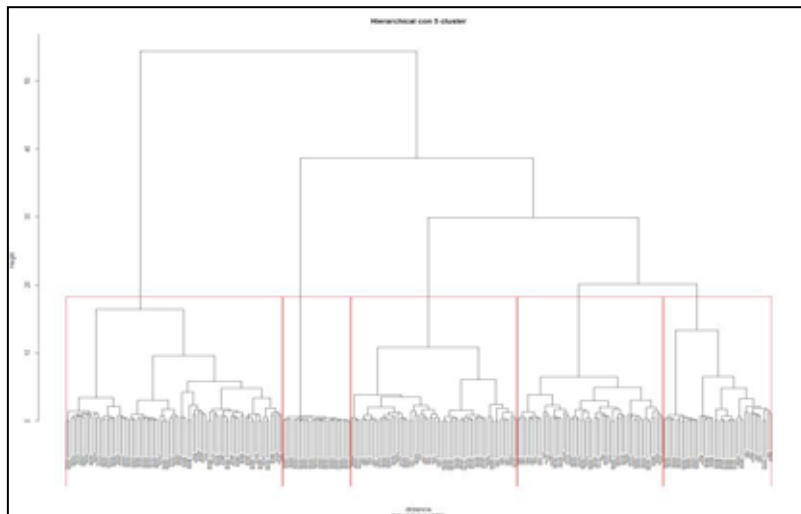


Figura E.11: Dendograma. Jerárquico con 5 clusters

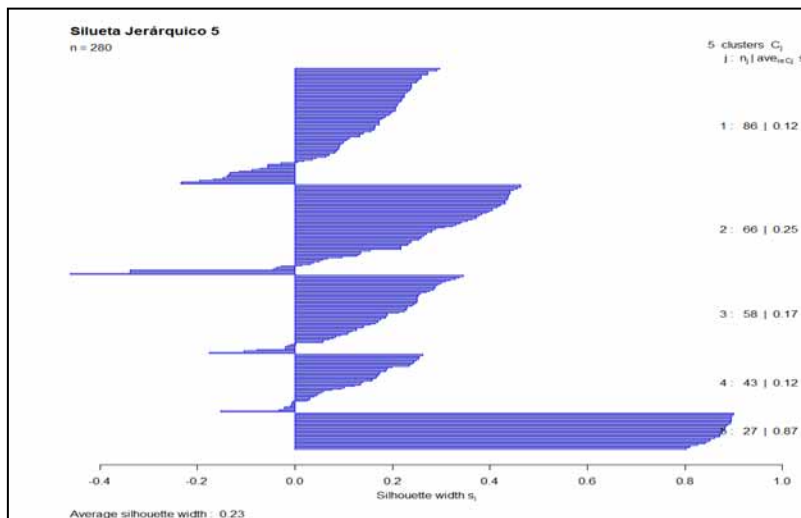


Figura E . 12: “Silhoouette” sobre Jerárquico con 5 clusters

- **Jerárquico con 6 clusters**

La *Tabla E.9*, expone el resultado al aglomerar en 6 cluster y es el mismo resultado obtenido en cada una de las iteraciones. La *Tabla E.10* muestra los tiempos de ejecución por iteración.

Tabla E.9: Jerárquico con 6 clusters

Hierarchical con 6 clusters de tamaño: 61,66,25,58,43,27						
Muestras	Grupos					
	1	2	3	4	5	6
DisparoReal	21	25	25	23	16	27
DisparoSalva	9	17	0	18	0	0
EfectoSonido	28	0	0	2	19	0
Explosivo	3	24	0	15	8	0

Tabla E.10: Tiempos de ejecución por cada iteración. Jerárquico con 6 clusters

iteración	Tiempo de ejecución
1	0.09 segundos
2	0.13 segundos
3	0.13 segundos
4	0.11 segundos
5	0.06 segundos
6	0.12 segundos
7	0.11 segundos
8	0.08 segundos
9	0.12 segundos
10	0.06 segundos

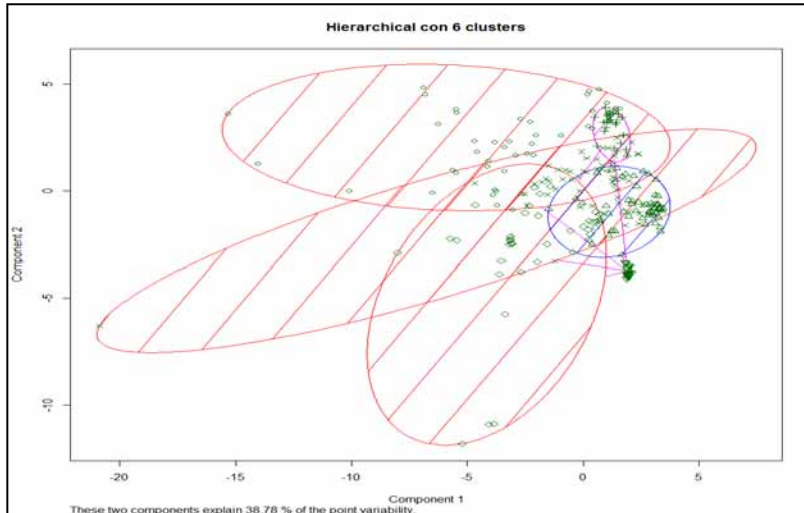


Figura E.13: Jerárquico con 6 clusters

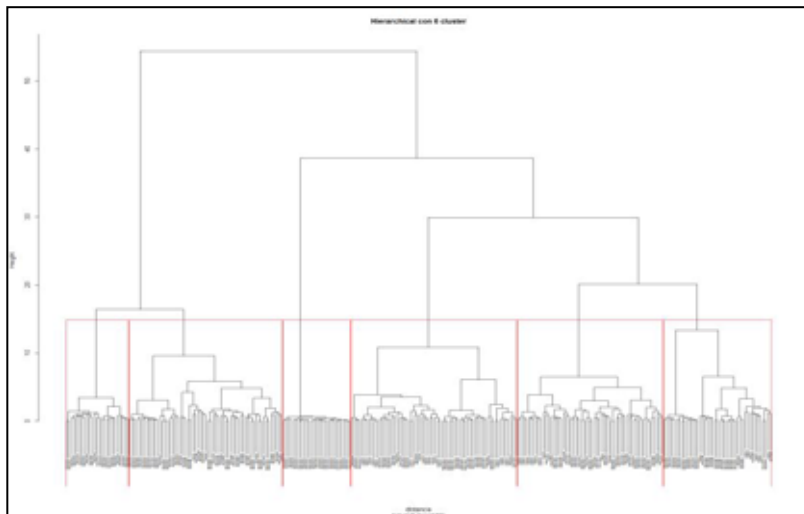


Figura E.14: Dendograma. Jerárquico con 6 clusters

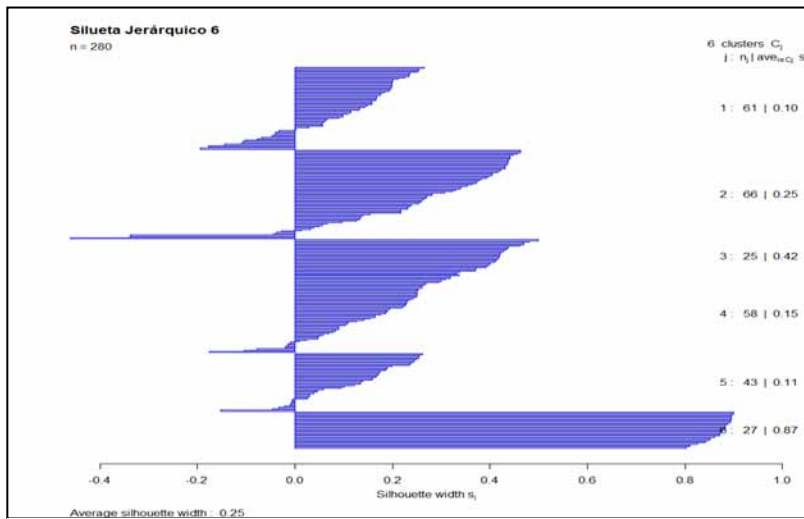


Figura E.15: “Silhouette” sobre Jerárquico con 6 clusters

- **Jerárquico con 7 clusters**

La *Tabla E.11*, expone el resultado al aglomerar en 7 *cluster* y es el mismo resultado obtenido en cada una de las iteraciones. La *Tabla E.12* muestra los tiempos de ejecución por iteración.

Tabla E.11: Jerárquico con 7 clusters

Hierarchical con 7 clusters de tamaño: 61,66,25,58,29,14,27							
Muestras	Grupos						
	1	2	3	4	5	6	7
DisparoReal	21	25	25	23	2	14	27
DisparoSalva	9	17	0	18	0	0	0
EfectoSonido	28	0	0	2	19	0	0
Explosivo	3	24	0	15	8	0	0

Tabla E.12: Tiempos de ejecución por cada iteración. Jerárquico con 7 clusters

iteración	Tiempo de ejecución
1	0.05 segundos
2	0.14 segundos
3	0.12 segundos
4	0.12segundos
5	0.12 segundos
6	0.12 segundos
7	0.06 segundos
8	0.12 segundos
9	0.14 segundos
10	0.14 segundos

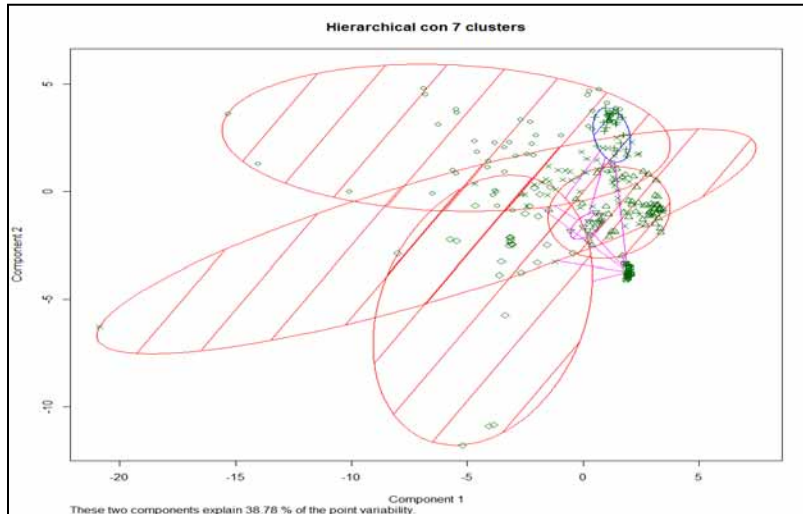


Figura E.16: Jerárquico con 7 clusters

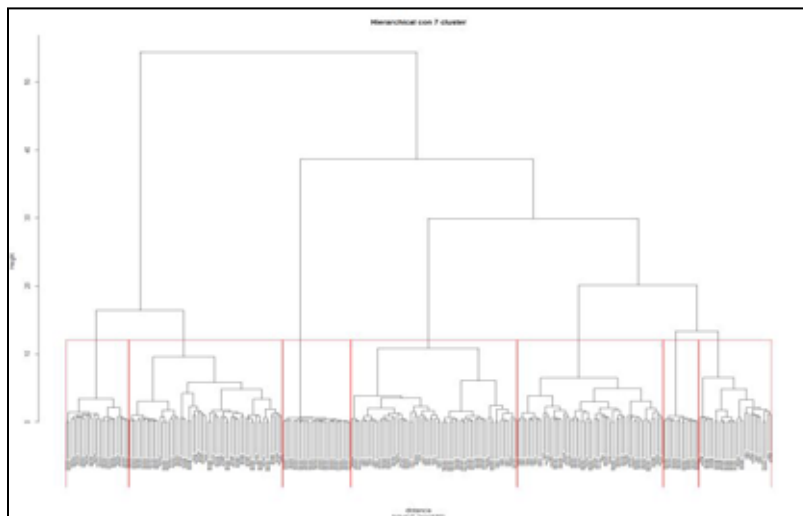


Figura E.17: Dendrograma. Jerárquico con 7 clusters

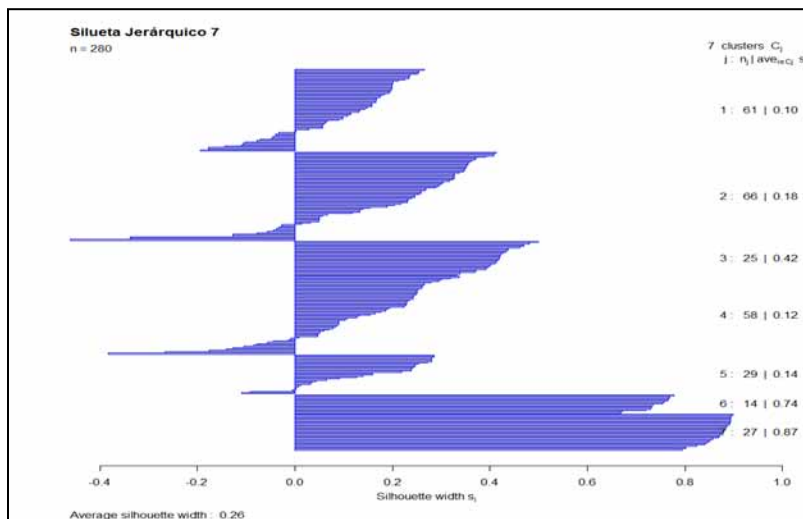


Figura E.18: "Silhouette" sobre Jerárquico con 7 clusters

- **Jerárquico con 8 clusters**

La *Tabla E.13*, expone el resultado al aglomerar en 8 *cluster* y es el mismo resultado obtenido en cada una de las iteraciones. La *Tabla E.14* muestra los tiempos de ejecución por iteración.

Tabla E.13: Jerárquico con 8 clusters

Hierarchical con 8 clusters de tamaño: 61,36,25,58,29,30,14,27								
Muestras	Grupos							
	1	2	3	4	5	6	7	8
DisparoReal	21	18	25	23	2	7	14	27
DisparoSalva	9	0	0	18	0	17	0	0
EfectoSonido	28	0	0	2	19	0	0	0
Explosivo	3	18	0	15	8	0	0	0

Tabla E.14: Tiempos de ejecución por cada iteración. Jerárquico con 8 clusters

iteración	Tiempo de ejecución
1	0.05 segundos
2	0.07 segundos
3	0.07 segundos
4	0.14segundos
5	0.13 segundos
6	0.13 segundos
7	0.12 segundos
8	0.08 segundos
9	0.13 segundos
10	0.13 segundos

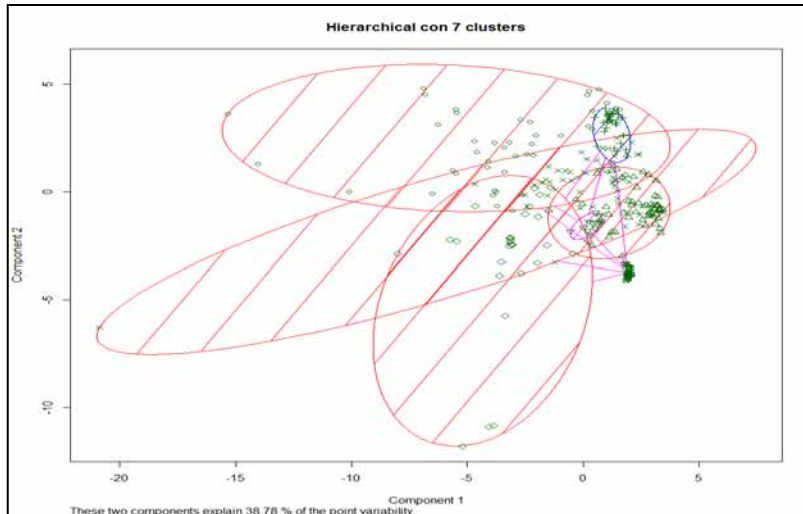


Figura E.19: Jerárquico con 7 clusters

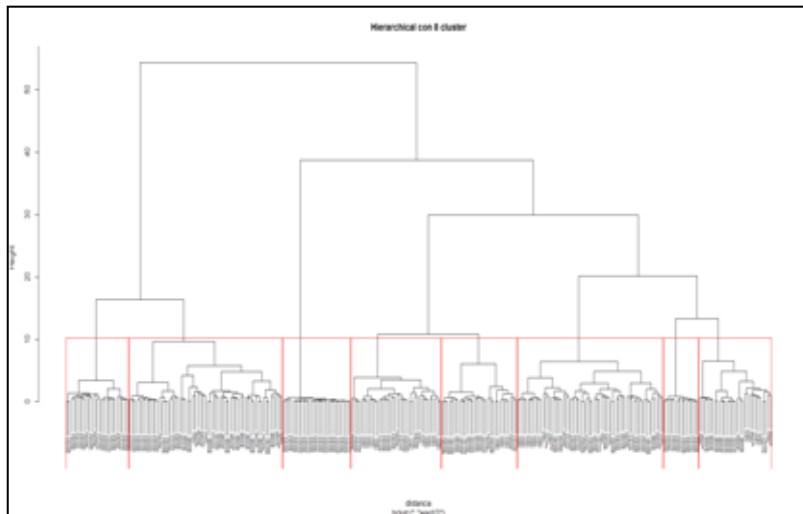


Figura E.20: Dendrograma. Jerárquico con 8 clusters

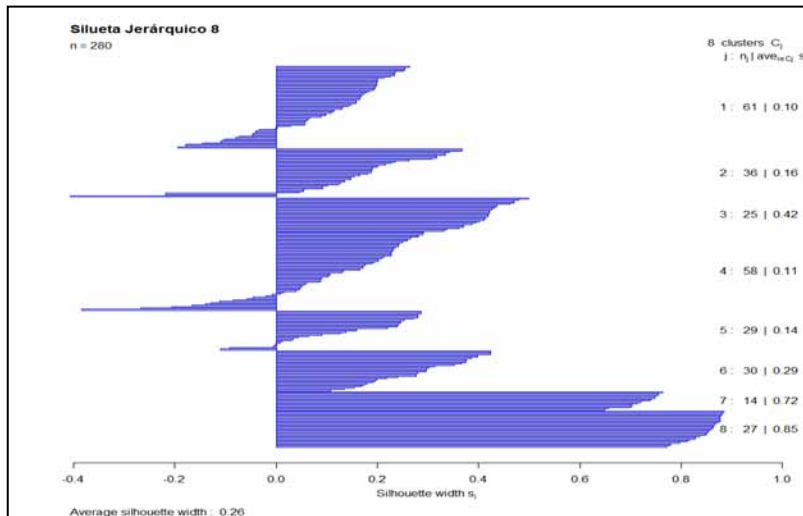


Figura E.21: "Silhouette" sobre Jerárquico con 8 clusters

- **Jerárquico con 9 clusters**

La *Tabla E.15*, expone el resultado al aglomerar en 9 **cluster** y es el mismo resultado obtenido en cada una de las iteraciones. La *Tabla E.16* muestra los tiempos de ejecución por iteración.

Tabla E.15: Jerárquico con 9 clusters

Hierarchical con 9 clusters de tamaño: 20,36,25,58,29,41,30,14,27									
Muestras	Grupos								
	1	2	3	4	5	6	7	8	9
DisparoReal	20	18	25	23	2	1	7	14	27
DisparoSalva	0	0	0	18	0	9	17	0	0
EfectoSonido	0	0	0	2	19	28	0	0	0
Explosivo	0	18	0	15	8	3	6	0	0

Tabla E.16: Tiempos de ejecución por cada iteración. Jerárquico con 9 clusters

iteración	Tiempo de ejecución
1	0.05 segundos
2	0.11 segundos
3	0.14 segundos
4	0.14segundos
5	0.13 segundos
6	0.15 segundos
7	0.13 segundos
8	0.14 segundos
9	0.12 segundos
10	0.12 segundos

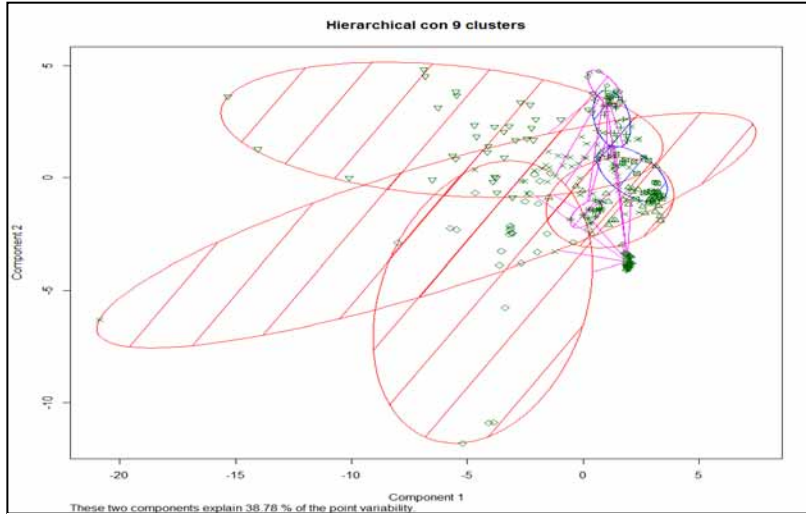


Figura E.22: Jerárquico con 9 clusters

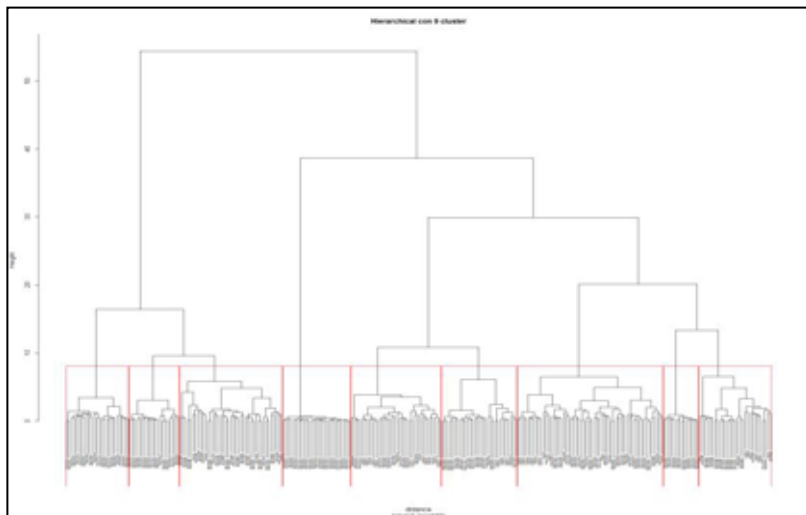


Figura E.23: Dendograma. Jerárquico con 9 clusters

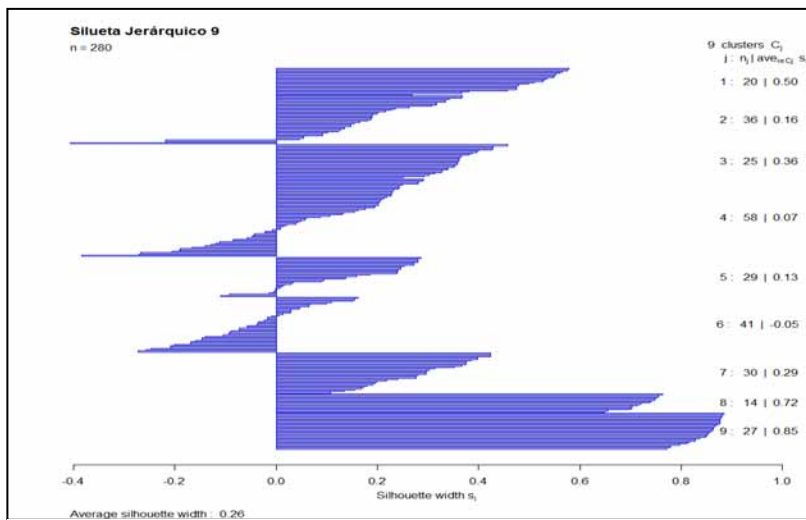


Figura E.24: “Silhouette” sobre Jerárquico con 9 clusters

- **Jerárquico con 10 clusters**

La *Tabla E.17*, expone el resultado al aglomerar en 10 *cluster* y es el mismo resultado obtenido en cada una de las iteraciones. La *Tabla E.18* muestra los tiempos de ejecución por iteración.

Tabla E.17: Jerárquico con 10 clusters

Hierarchical con 10 clusters de tamaño: 20,36,25,58,23,41,30,14,27,6										
Muestras	Grupos									
	1	2	3	4	5	6	7	8	9	10
DisparoReal	20	18	25	23	2	1	7	14	27	0
DisparoSalva	0	0	0	18	0	9	17	0	0	0
EfectoSonido	0	0	0	2	14	28	0	0	0	5
Explosivo	0	18	0	15	7	3	6	0	0	1

Tabla E.18: Tiempos de ejecución por cada iteración. Jerárquico con 10 clusters

iteración	Tiempo de ejecución
1	0.09 segundos
2	0.14 segundos
3	0.14 segundos
4	0.11 segundos
5	0.08 segundos
6	0.15 segundos
7	0.15 segundos
8	0.13 segundos
9	0.14 segundos
10	0.14 segundos

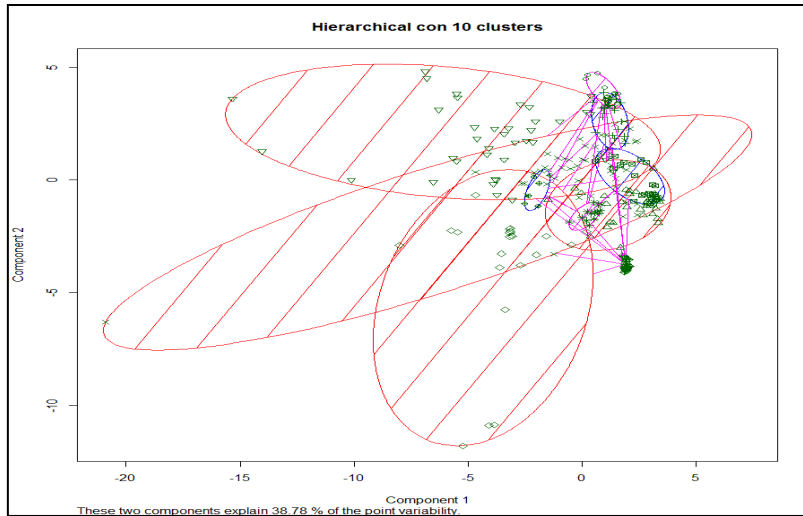


Figura E.25: Jerárquico con 10 clusters

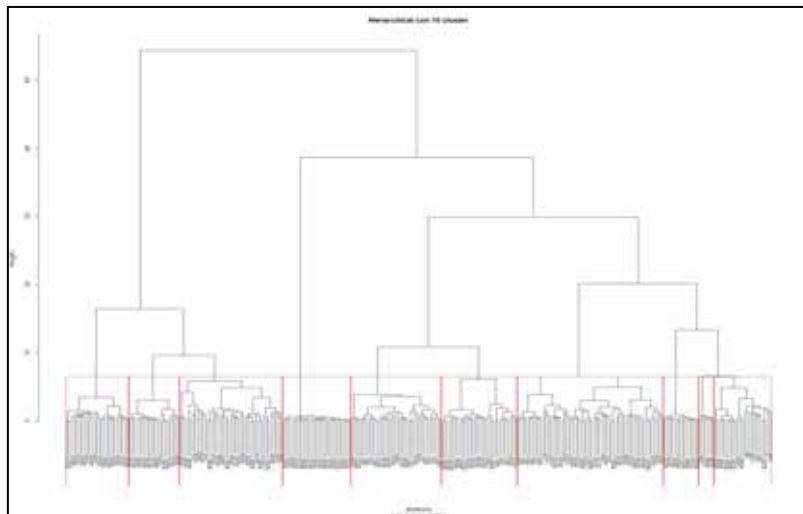


Figura E.26: Dendograma. Jerárquico con 10 clusters

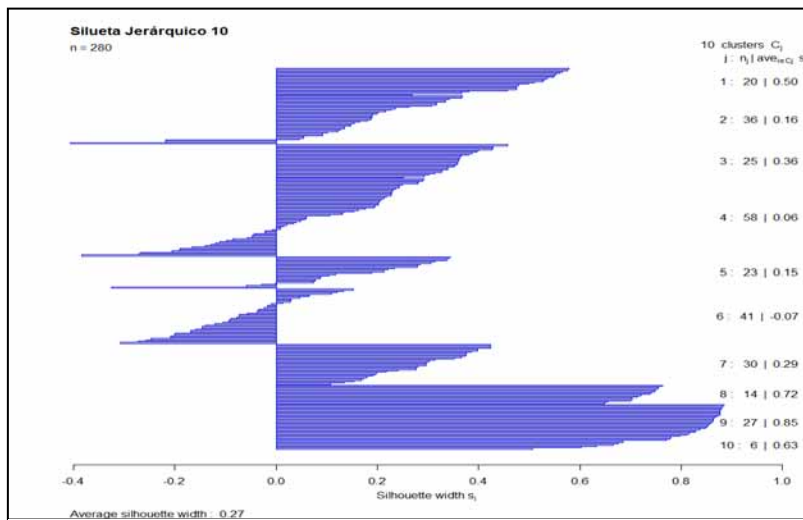


Figura E.27: “Silhouette” sobre Jerárquico con 10 clusters

E.2 Resultados obtenidos al implementar Jerárquico sobre 301 observaciones promediadas por bandas de frecuencia de 50 en 50 Hz

- **Jerárquico con 2 grupos**

La *Tabla E.19*, expone el resultado al aglomerar en 2 *cluster* y es el mismo resultado obtenido en cada una de las iteraciones. La *Tabla E.20* muestra los tiempos de ejecución por iteración.

Tabla E.19: Jerárquico con 2 grupos

Hierarchical con 2 clusters de tamaño: 213,86		
Muestras	Grupos	
	1	2
DisparoReal	105	47
DisparoSalva	49	1
EfectoSonido	24	25
Explosivo	35	15

Tabla E.20: Tiempos de ejecución por cada iteración. Jerárquico con 2 clusters

iteración	Tiempo de ejecución
1	0.17 segundos
2	0.14 segundos
3	0.10 segundos
4	0.22 segundos
5	0.10 segundos
6	0.11 segundos
7	0.11 segundos
8	0.11 segundos
9	0.11 segundos
10	0.11 segundos

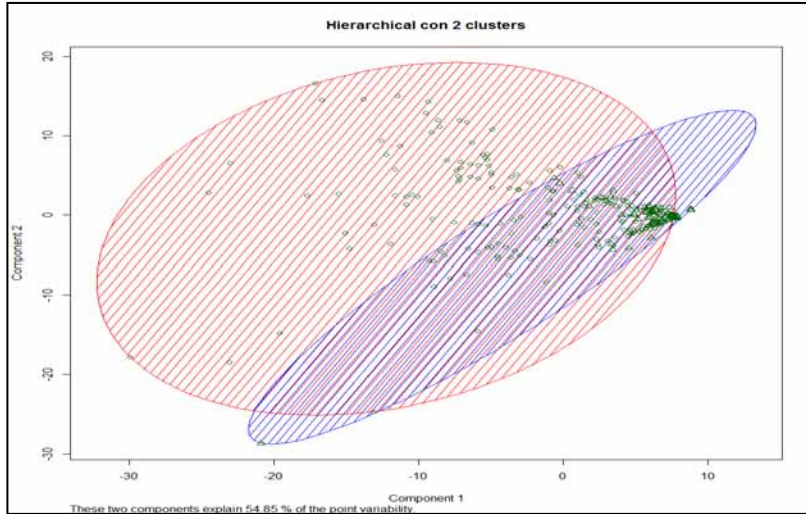


Figura E.28: Jerárquico con dos clusters

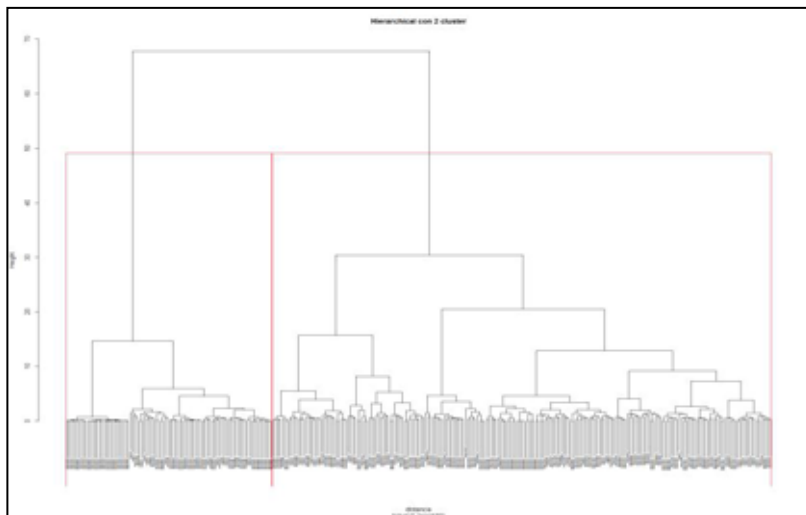


Figura E.29: Dendograma. Jerárquico con 2 clusters

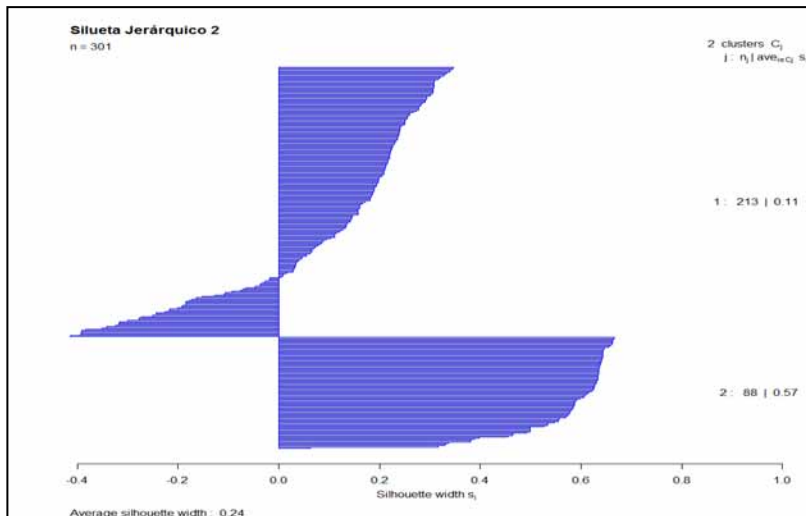


Figura E.30: “Silhouette” sobre jerárquico con 2 clusters

- **Jerárquico con 3 clusters**

La *Tabla E.21*, expone el resultado al aglomerar en 3 **cluster** y es el mismo resultado obtenido en cada una de las iteraciones. La *Tabla E.22* muestra los tiempos de ejecución por iteración

Tabla E.21: Jerárquico con 3 clusters

Hierarchical con 3 clusters de tamaño: 148,65,88			
Muestras	Grupos		
	1	2	3
DisparoReal	94	11	47
DisparoSalva	33	18	1
EfectoSonido	3	21	25
Explosivo	18	17	15

Tabla E.22: Tiempos de ejecución por cada iteración. Jerárquico con 3 clusters

iteración	Tiempo de ejecución
1	0.19 segundos
2	0.15 segundos
3	0.17 segundos
4	0.11 segundos
5	0.11 segundos
6	0.11 segundos
7	0.11 segundos
8	0.12 segundos
9	0.13 segundos
10	0.17 segundos

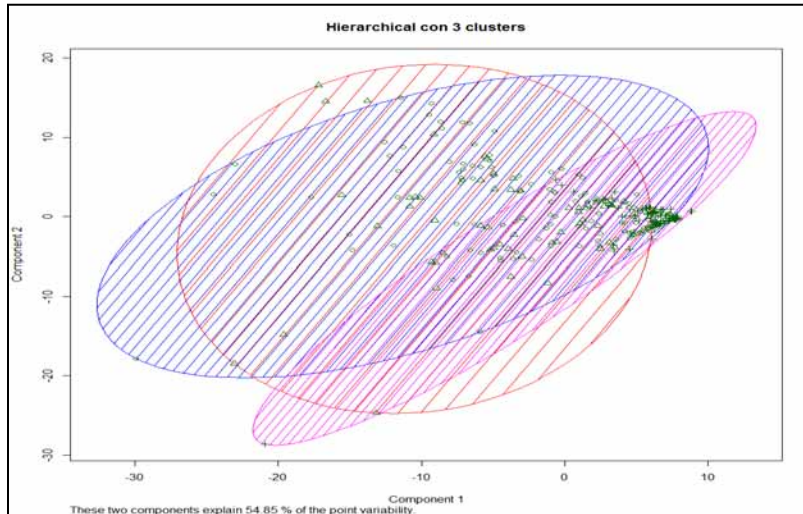


Figura E.31: Jerárquico con 3 clusters

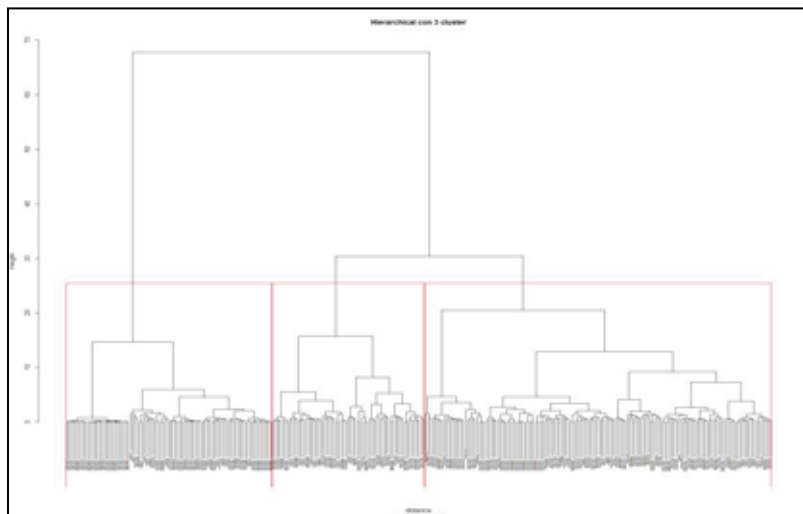


Figura E.32: Dendrograma. Jerárquico con 3 clusters

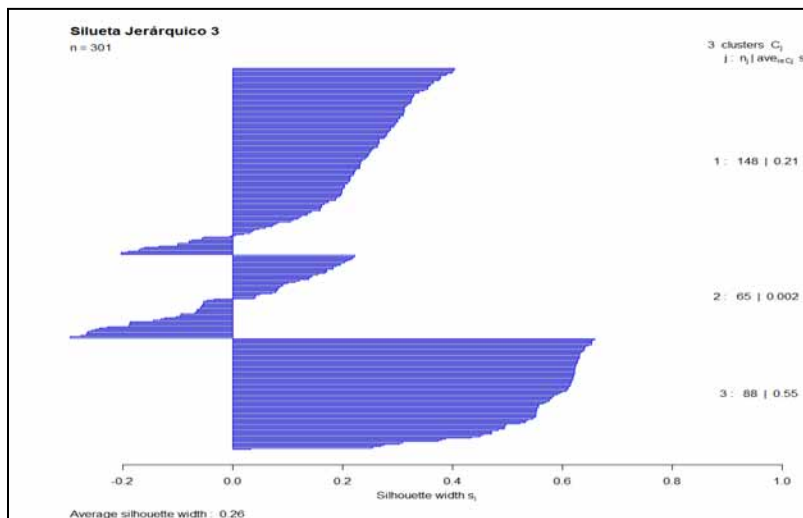


Figura E.33: "Silouette" sobre Jerárquico con 3 clusters

- **Jerárquico con 4 clusters**

La *Tabla E.23*, expone el resultado al aglomerar en 4 *cluster* y es el mismo resultado obtenido en cada una de las iteraciones. La *Tabla E.24* muestra los tiempos de ejecución por iteración

Tabla E.23: Jerárquico con 4 clusters

Hierarchical con 4 clusters de tamaño: 26,122,65,88				
Muestras	Grupos			
	1	2	3	4
DisparoReal	26	68	11	47
DisparoSalva	0	33	16	1
EfectoSonido	0	3	21	25
Explosivo	0	18	17	15

Tabla E.24: Tiempos de ejecución por cada iteración. Jerárquico con 4 clusters

iteración	Tiempo de ejecución
1	0.19 segundos
2	0.22 segundos
3	0.17 segundos
4	0.12segundos
5	0.13 segundos
6	0.11 segundos
7	0.11 segundos
8	0.11 segundos
9	0.11 segundos
10	0.17 segundos

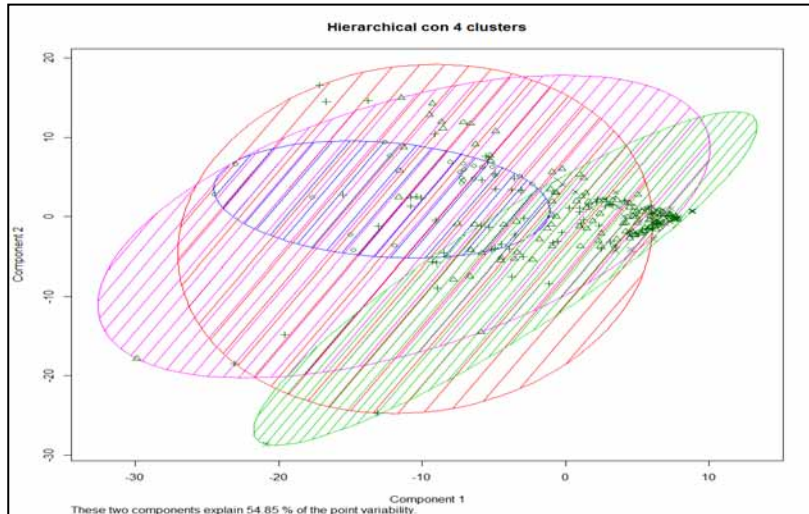


Figura E.34: Jerárquico con 4 clusters

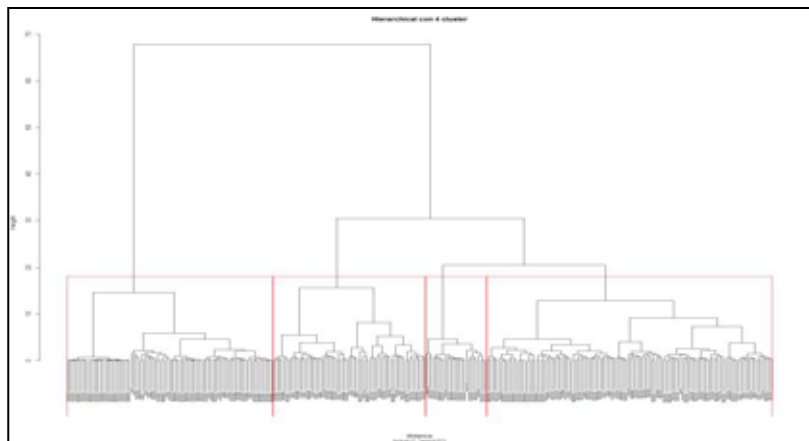


Figura E.35: Dendograma. Jerárquico con 4 clusters

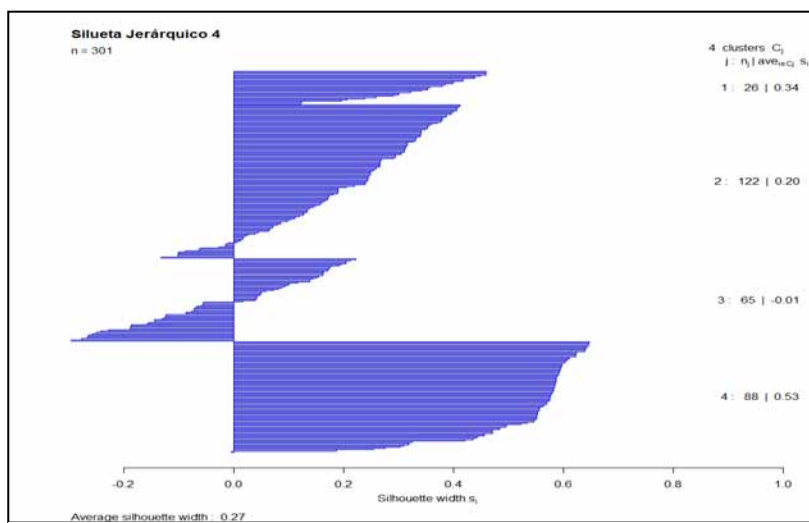


Figura E.36: "Silhouette". Jerárquico con 4 clusters

- **Jerárquico con 5 clusters**

La *Tabla E.25*, expone el resultado al aglomerar en 5 cluster y es el mismo resultado obtenido en cada una de las iteraciones. La *Tabla E.26* muestra los tiempos de ejecución por iteración

Tabla E.25: jerárquico con 5 clusters

Hierarchical con 5 clusters de tamaño: 26,122,33,88,32					
Muestras	Grupos				
	1	2	3	4	5
DisparoReal	26	68	11	47	0
DisparoSalva	0	33	11	1	5
EfectoSonido	0	3	6	25	15
Explosivo	0	18	17	15	12

Tabla E.26: Tiempos de ejecución por cada iteración. Jerárquico con 5 clusters

iteración	Tiempo de ejecución
1	0.23 segundos
2	0.14 segundos
3	0.33 segundos
4	0.11 segundos
5	0.12 segundos
6	0.11 segundos
7	0.11 segundos
8	0.17 segundos
9	0.13 segundos
10	0.11 segundos

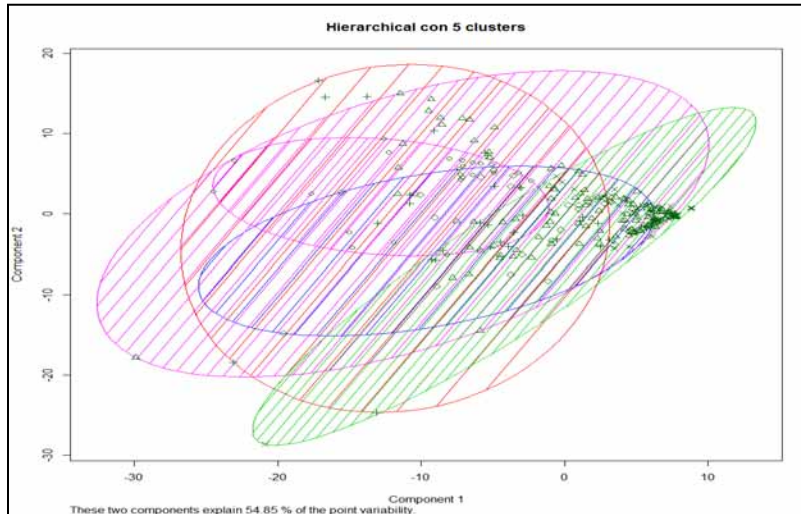


Figura E.37: Jerárquico con 5 clusters

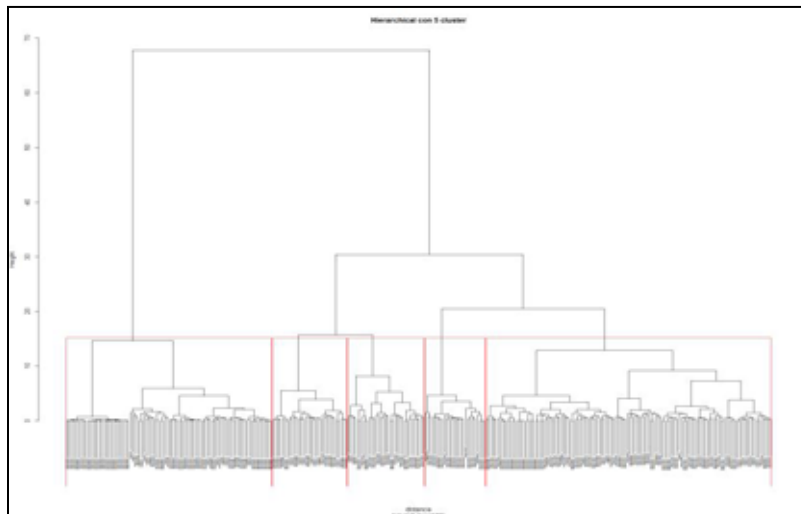


Figura E.38: Dendograma. Jerárquico con 5 clusters

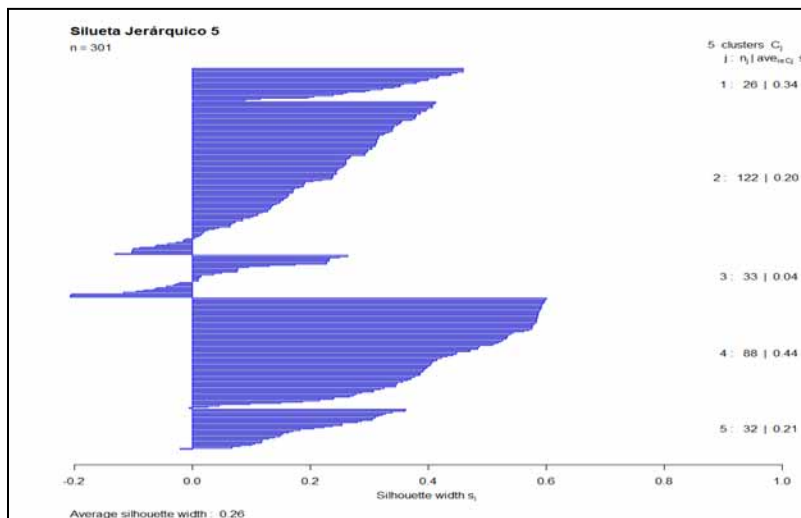


Figura E.39: “Silhouette” sobre Jerárquico con 5 clusters

- **Jerárquico con 6 clusters**

La *Tabla E.27*, expone el resultado al aglomerar en 6 *cluster* y es el mismo resultado obtenido en cada una de las iteraciones. La *Tabla E.28* muestra los tiempos de ejecución por iteración

Tabla E.27: Jerárquico con 6 clusters

Hierarchical con 6 clusters de tamaño: 26,122,33,61,27,32						
Muestras	Grupos					
	1	2	3	4	5	6
DisparoReal	26	68	11	20	27	0
DisparoSalva	0	33	11	1	0	5
EfectoSonido	0	3	6	25	0	15
Explosivo	0	18	5	15	0	12

Tabla E.28: Tiempos de ejecución por cada iteración. Jerárquico con 6 clusters

iteración	Tiempo de ejecución
1	0.20 segundos
2	0.21 segundos
3	0.18 segundos
4	0.09segundos
5	0.13 segundos
6	0.11 segundos
7	0.09 segundos
8	0.11 segundos
9	0.11 segundos
10	0.15 segundos

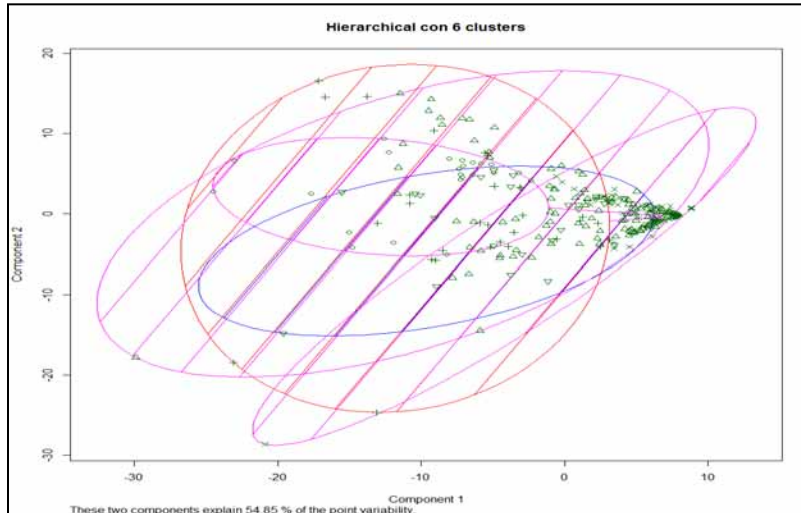


Figura E.40: Jerárquico con 6 clusters

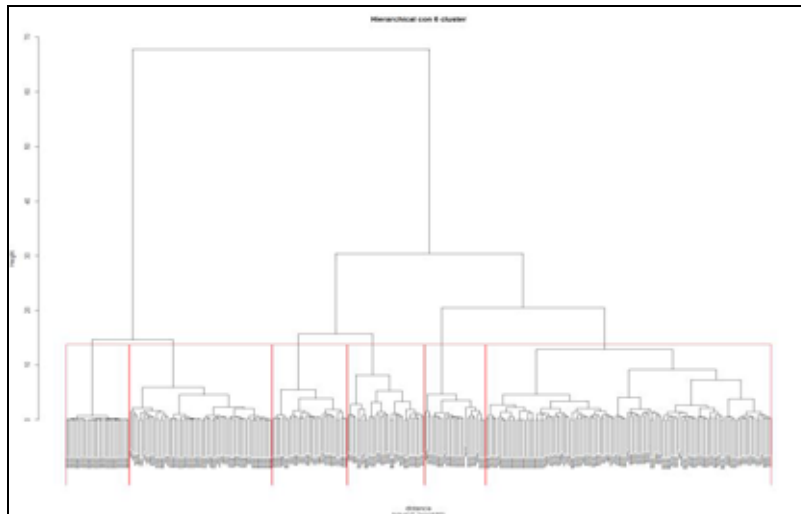


Figura E.41: Dendrograma . Jerárquico con 6 clusters

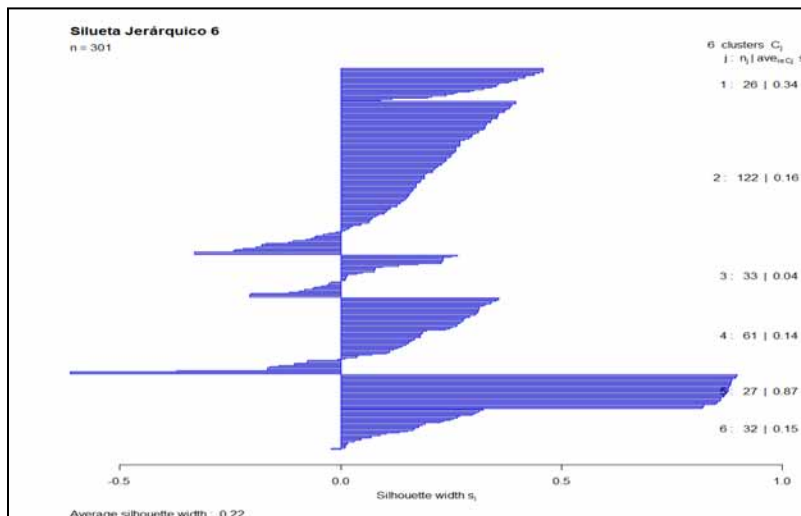


Figura E.42: "Silhouette" sobre Jerárquico con 6 clusters

- **Jerárquico con 7 clusters**

La *Tabla E.29*, expone el resultado al aglomerar en 7 *cluster* y es el mismo resultado obtenido en cada una de las iteraciones. La *Tabla E.30* muestra los tiempos de ejecución por iteración

Tabla E.29: Jerárquico con 7 clusters

Hierarchical con 7 clusters de tamaño: 26,67,33,55,61,27,32							
Muestras	Grupos						
	1	2	3	4	5	6	7
DisparoReal	26	38	11	30	20	27	0
DisparoSalva	0	19	11	14	1	0	5
EfectoSonido	0	0	6	3	25	0	15
Explosivo	0	10	5	8	15	0	12

Tabla E.30: Tiempos de ejecución por cada iteración. Jerárquico con 7 clusters

iteración	Tiempo de ejecución
1	0.25 segundos
2	0.17 segundos
3	0.21 segundos
4	0.11 segundos
5	0.13 segundos
6	0.09 segundos
7	0.11 segundos
8	0.12 segundos
9	0.14 segundos
10	0.16 segundos

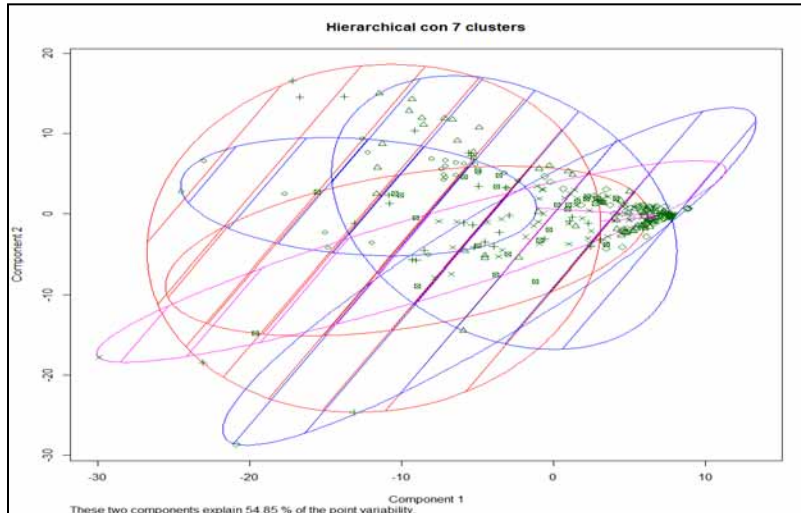


Figura E.43: Jerárquico con 7 clusters

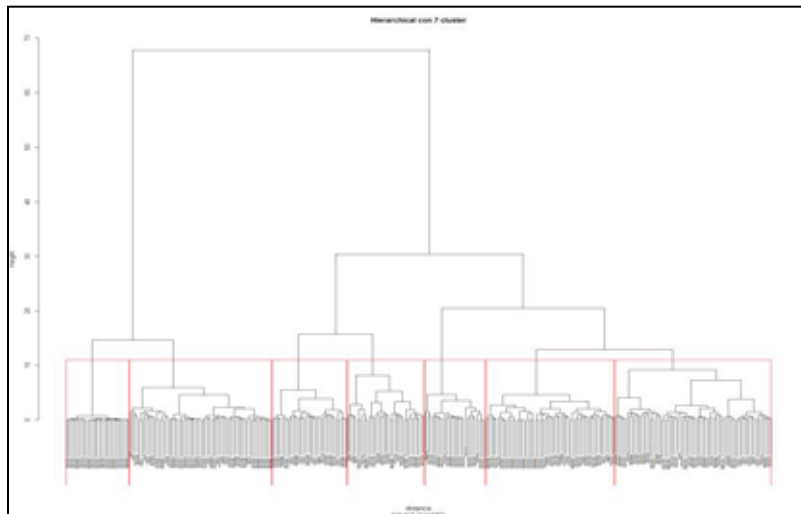


Figura E.44: Dendrograma. Jerárquico con 7 clusters

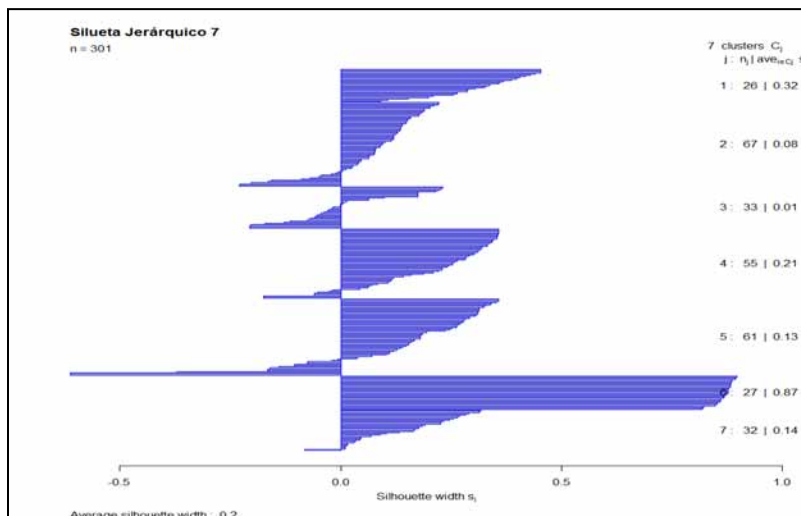


Figura E.45: "Silhouette" sobre Jerárquico con 7 clusters

- **Jerárquico con 8 clusters**

La *Tabla E.31*, expone el resultado al aglomerar en 8 *cluster* y es el mismo resultado obtenido en cada una de las iteraciones. La *Tabla E.32* muestra los tiempos de ejecución por iteración

Tabla E.31; Jerárquico con 8 clusters

Hierarchical con 8 clusters de tamaño: 26, 47, 20, 33, 55, 61, 27, 32								
Muestra	Grupos							
	1	2	3	4	5	6	7	8
DisparoReal	26	18	20	11	30	20	27	0
DisparoSalva	0	19	0	11	14	1	0	5
EfectoSonido	0	0	0	6	3	25	0	15
Explosivo	0	10	0	5	4	15	0	12

Tabla E.32: Tiempos de ejecución por cada iteración. Jerárquico con 8 clusters

iteración	Tiempo de ejecución
1	0.24 segundos
2	0.16 segundos
3	0.23 segundos
4	0.10segundos
5	0.11 segundos
6	0.10 segundos
7	0.11 segundos
8	0.13 segundos
9	0.14 segundos
10	0.18 segundos

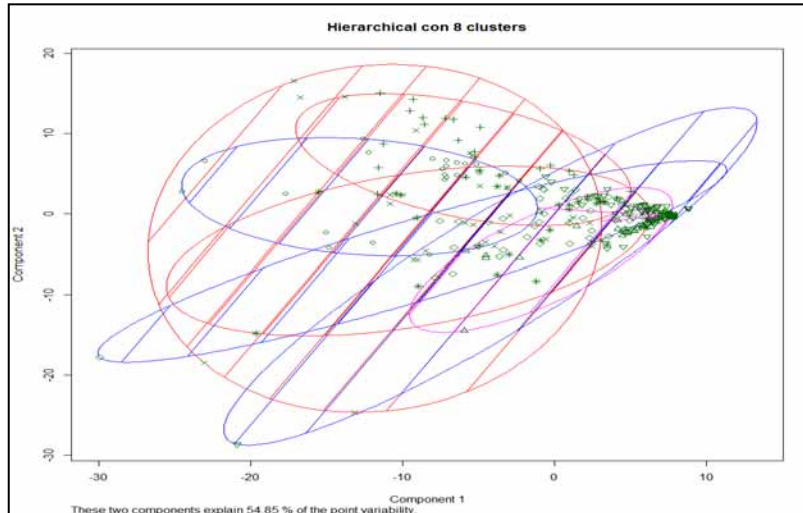


Figura E.46: Jerárquico con 8 clusters

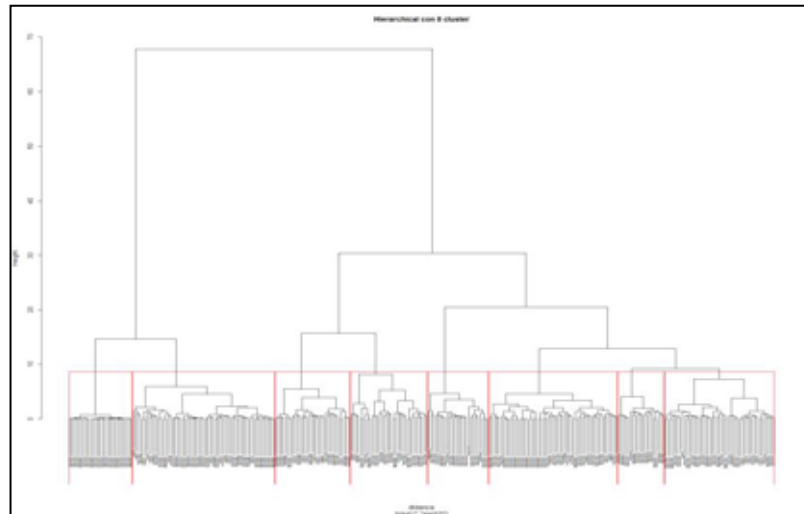


Figura E.47: Dendrograma. Jerárquico con 8 clusters

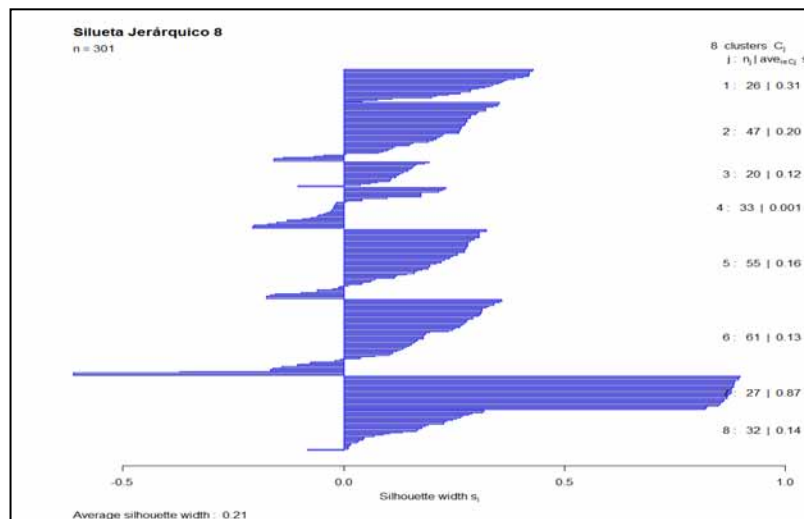


Figura E.48: “Silhouette” sobre Jerárquico con 8 clusters

- **Jerárquico con 9 clusters**

La *Tabla E.33*, expone el resultado al aglomerar en 9 *cluster* y es el mismo resultado obtenido en cada una de las iteraciones. La *Tabla E.34* muestra los tiempos de ejecución por iteración

Tabla E.33: Jerárquico con 9 clusters

Hierarchical con 9 clusters de tamaño: 26, 47, 20, 24, 55, 61, 9, 27, 32									
Muestras	Grupos								
	1	2	3	4	5	6	7	8	9
DisparoReal	26	18	20	6	30	20	5	27	0
DisparoSalva	0	19	0	11	14	1	0	0	5
EfectoSonido	0	0	0	6	3	25	0	0	15
Explosivo	0	10	0	1	8	15	4	0	12

Tabla E.34: Tiempos de ejecución por cada iteración. Jerárquico con 9 clusters

iteración	Tiempo de ejecución
1	0.19 segundos
2	0.19 segundos
3	0.18 segundos
4	0.11 segundos
5	0.11 segundos
6	0.11 segundos
7	0.11 segundos
8	0.11 segundos
9	0.14 segundos
10	0.19 segundos

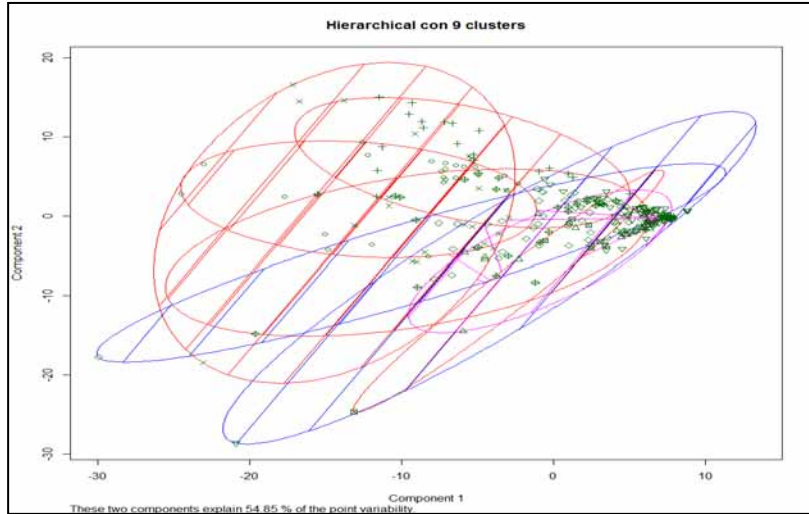


Figura E.49: Jerárquico con 9 clusters

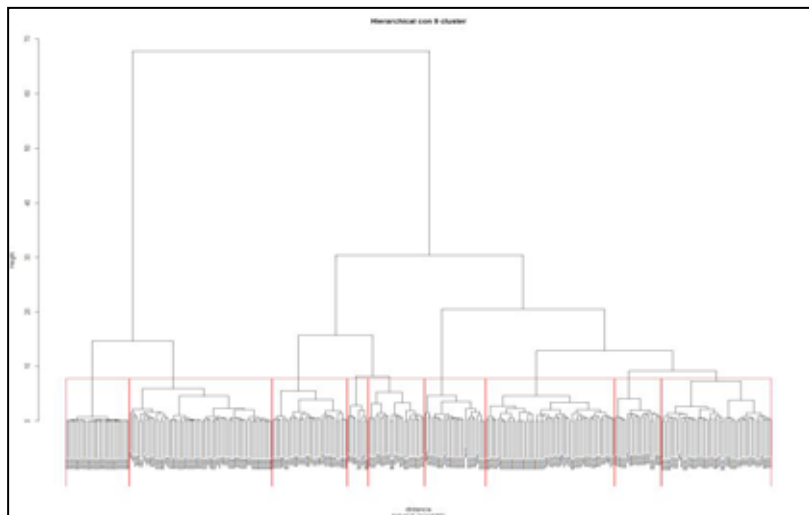


Figura E.50: Dendograma. Jerárquico con 9 clusters

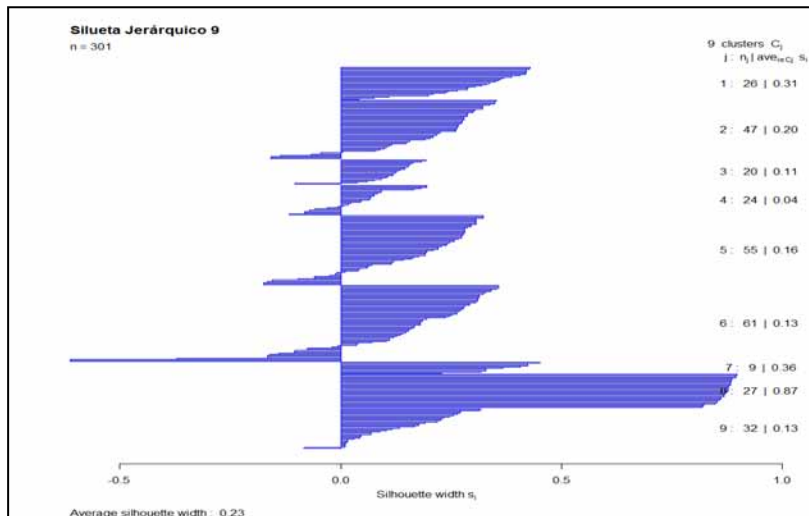


Figura E.51: "Silhouette" sobre Jerárquico con 9 clusters

- **Jerárquico con 10 clusters**

La *Tabla E.35*, expone el resultado al aglomerar en 10 **cluster** y es el mismo resultado obtenido en cada una de las iteraciones. La *Tabla E.36* muestra los tiempos de ejecución por iteración.

Tabla E.35: Jerárquico con 10 clusters

Hierarchical con 10 clusters de tamaño: 26, 20, 20, 24, 27, 55, 61, 9, 27, 32										
Muestras	Grupos									
	1	2	3	4	5	6	7	8	9	10
DisparoReal	26	11	20	6	7	30	20	5	27	0
DisparoSalva	0	0	0	11	19	14	1	0	0	5
EfectoSonido	0	0	0	6	0	3	25	0	0	15
Explosivo	0	9	0	1	1	8	15	4	0	12

Tabla E.36: Tiempos de ejecución por cada iteración. Jerárquico con 10 clusters

iteración	Tiempo de ejecución
1	0.24 segundos
2	0.17 segundos
3	0.22 segundos
4	0.22segundos
5	0.23 segundos
6	0.20 segundos
7	0.24 segundos
8	0.29 segundos
9	0.20 segundos
10	0.25 segundos

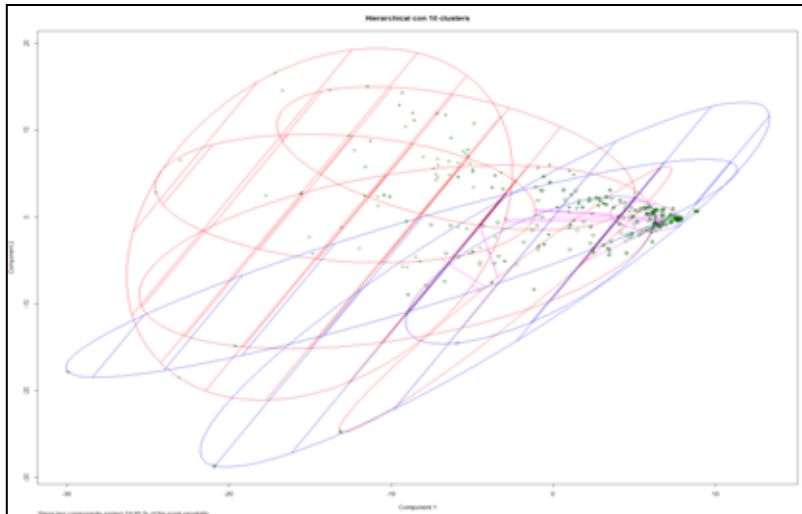


Figura E.52: Jerárquico con 10 clusters

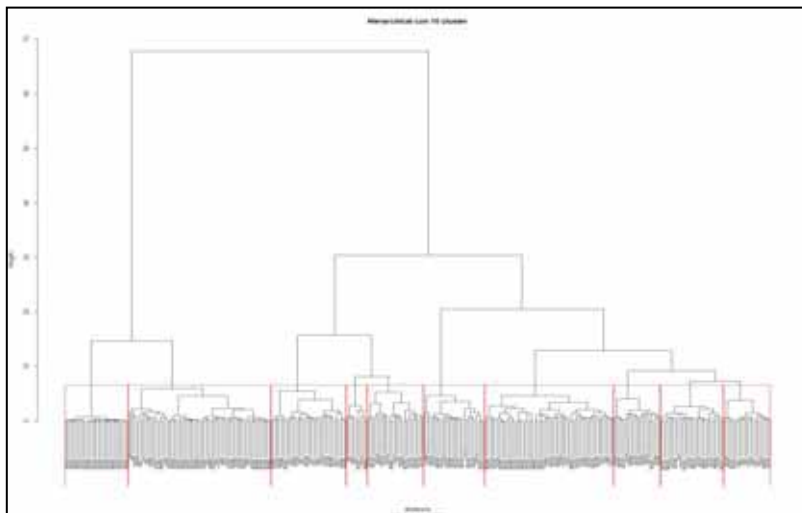


Figura E.53: Dendrograma. Jerárquico con 10 clusters

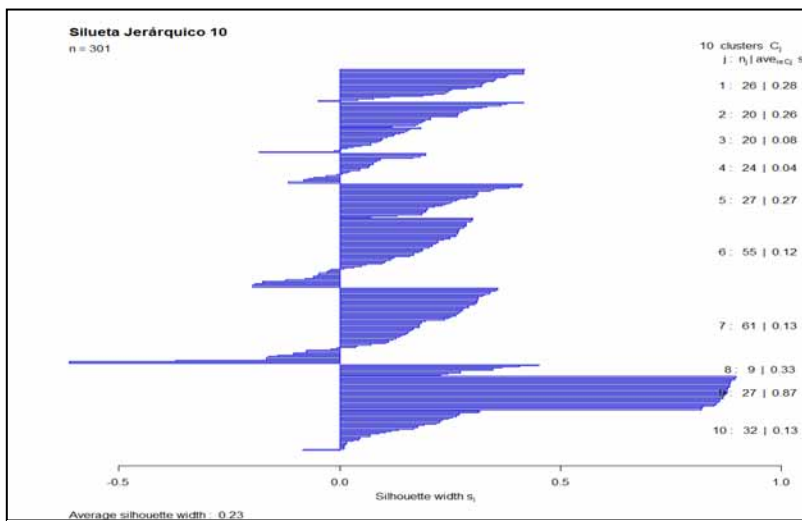


Figura E.54: “Silhouette” sobre Jerárquico con 10 clusters

Apéndice F

Resultados obtenidos al implementar el método Jerárquico con distancia similaridad “DTW” del inglés “Dynamic Time Warping”

F.1 Resultados obtenidos al implementar el método Jerárquico con distancia “DTW” sobre las 280 observaciones promediadas en escala logarítmica

- Jerárquico con distancia “DTW” y agrupando en 2 clusters

Tabla F.1: Jerárquico con 2 clusters

Hierarchical con distancia “DTW”, aglomerando en 2 clusters de tamaño: 213,67		
Muestras	Grupos	
	1	2
DisparoReal	82	55
DisparoSalva	42	2
EfectoSonido	49	0
Explosivo	40	10
Tiempo de ejecución 124.50 segundos		

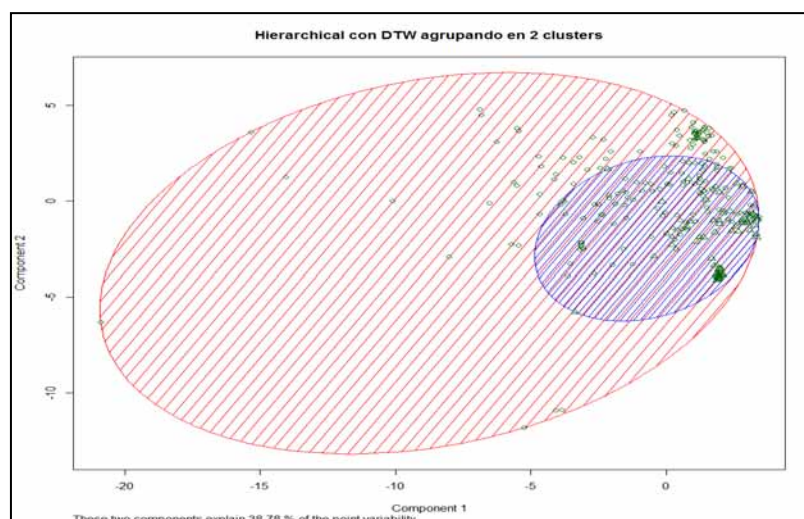


Figura F.1: Jerárquico con distancia “DTW”, con dos clusters

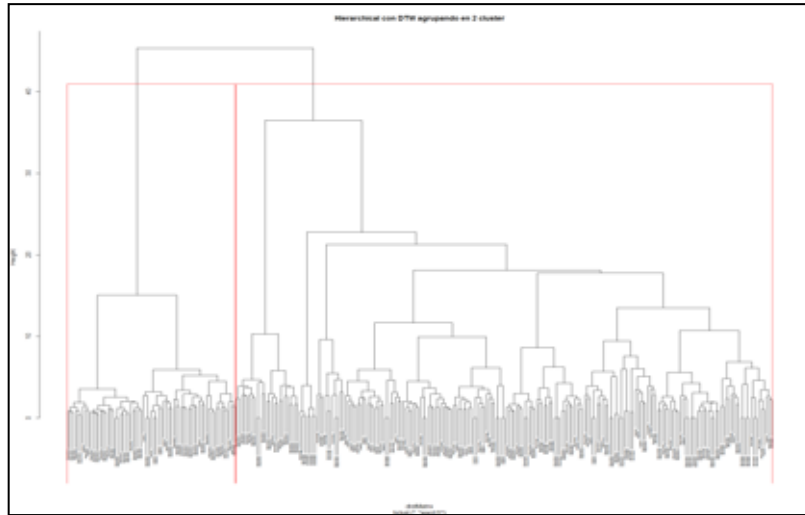


Figura F.2: Dendograma. Jerárquico con distancia “DTW” con 2 clusters

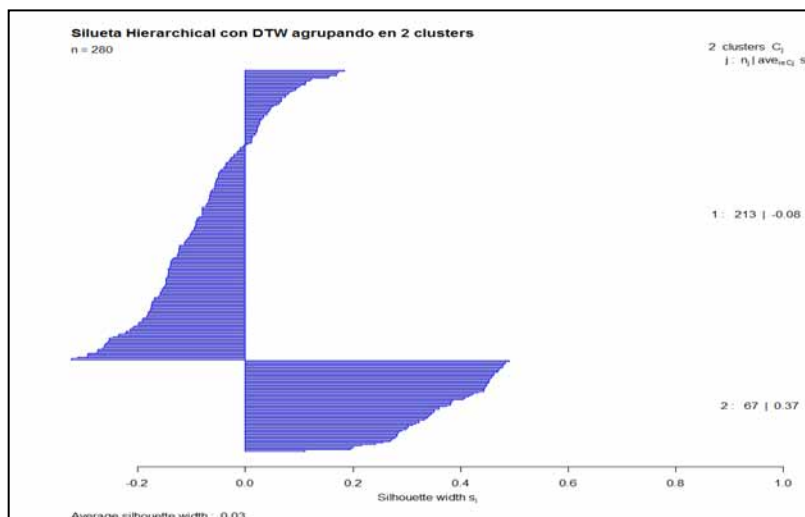


Figura F.3: “Silhouette” sobre jerárquico con distancia “DTW”, con 2 clusters

- Jerárquico con distancia “DTW” y agrupando en 3 clusters

Tabla F.2: Jerárquico con distancia “DTW” y agrupando con 3 clusters

Hierarchical con distancia “DTW”, aglomerando 3 clusters de tamaño: 188,67,25			
Muestras	Grupos		
	1	2	3
DisparoReal	58	55	24
DisparoSalva	42	2	0
EfectoSonido	49	0	0
Explosivo	39	10	1
Tiempo de ejecución 120.68 segundos			

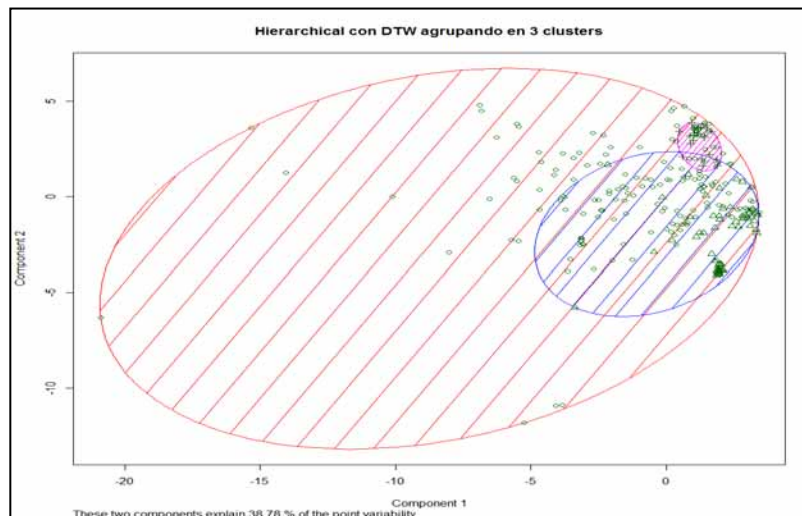


Figura F.4: Jerárquico con distancia “DTW”, con 3 clusters

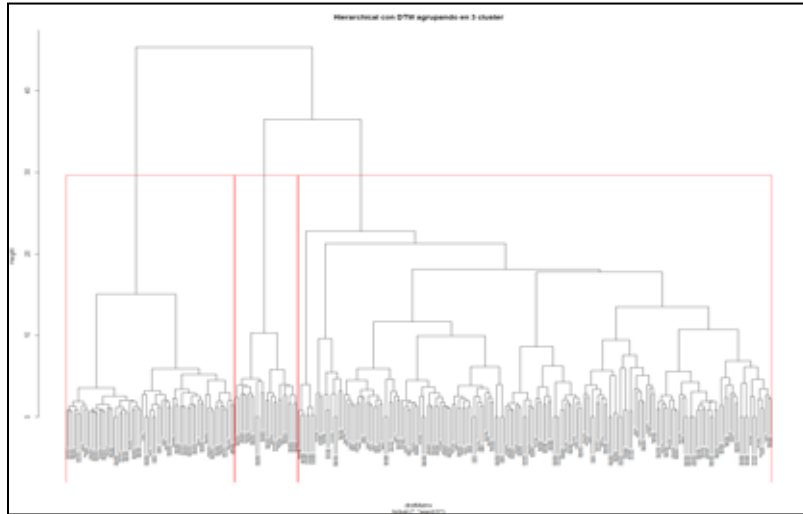


Figura F.5: Dendrograma. Jerárquico con distancia “DTW” , con 3 clusters

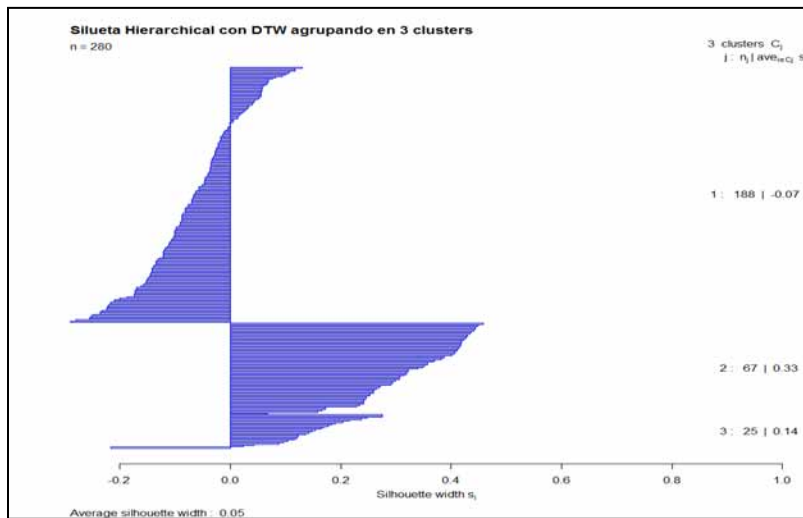


Figura F.6: “Silhouette” sobre jerárquico con distancia “DTW”, con 3 clusters

- Jerárquico con distancia “DTW” y agrupando en 4 clusters

Tabla F.3: Jerárquico con distancia “DTW” y agrupando con 4 clusters

Hierarchical con distancia “DTW”, aglomerando 4 clusters de tamaño:181,67,25, 7				
Muestras	Grupos			
	1	2	3	4
DisparoReal	58	55	24	0
DisparoSalva	42	2	0	0
EfectoSonido	42	0	0	7
Explosivo	39	10	1	0
Tiempo de ejecución 120.64 segundos				

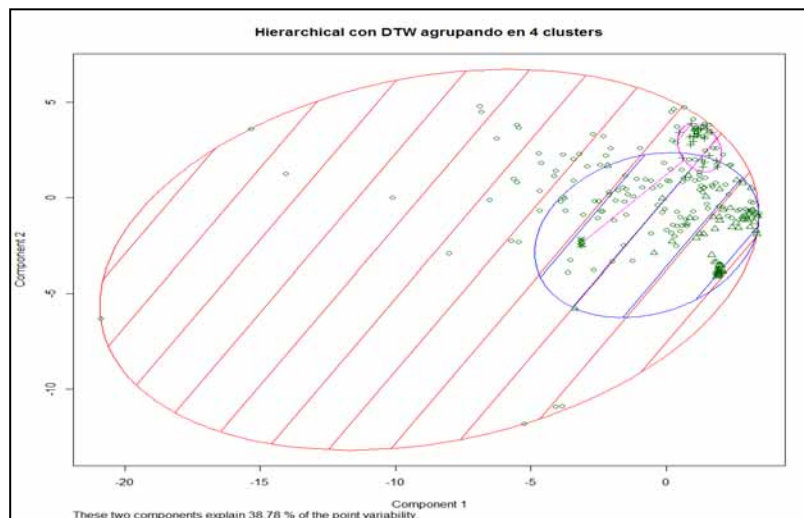


Figura F.7: Jerárquico con distancia “DTW”, con 4 clusters

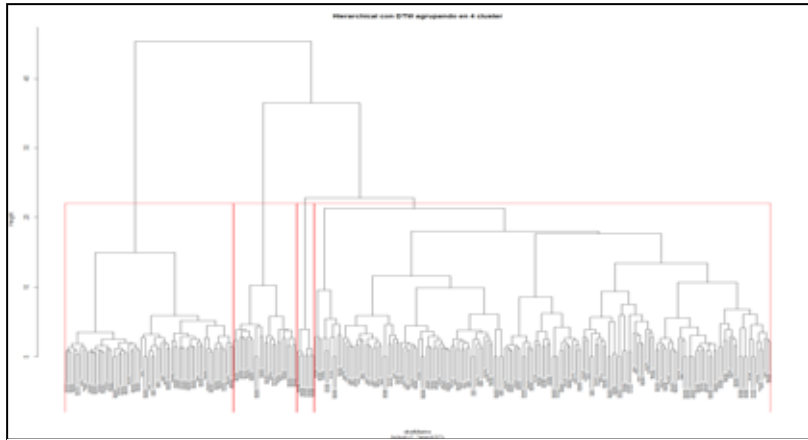


Figura F.8: Dendograma. Jerárquico con distancia “DTW” , con 4 clusters

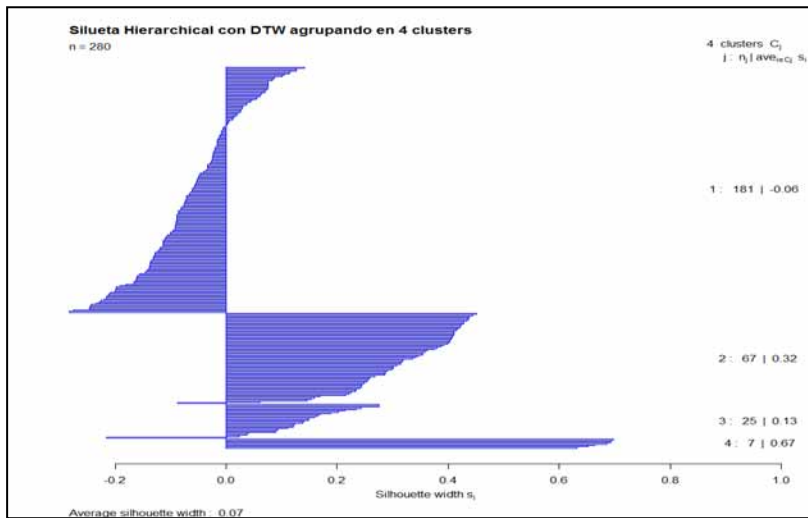


Figura F.9: “Silhouette” sobre jerárquico con distancia “DTW “,con 4 clusters

- Jerárquico con distancia “DTW” y agrupando en 5 clusters

Tabla F.4: Jerárquico con distancia “DTW” y agrupando con 5 clusters

Hierarchical con distancia “DTW”, aglomerando 5 clusters de tamaño: 170,67,25,11, 7					
Muestras	Grupos				
	1	2	3	4	5
DisparoReal	58	55	24	0	0
DisparoSalva	39	2	0	3	0
EfectoSonido	34	0	0	8	7
Explosivo	39	10	1	0	0
Tiempo de ejecución 123.16					

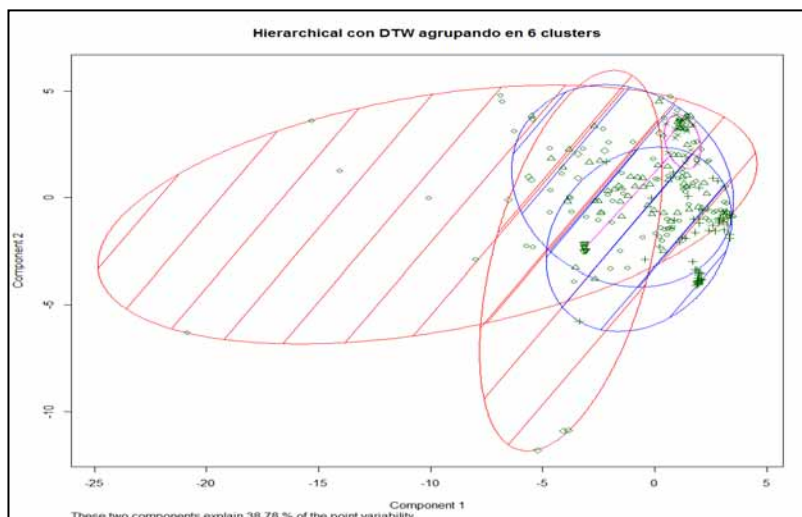


Figura F.10: Jerárquico con distancia “DTW”, con 5 clusters

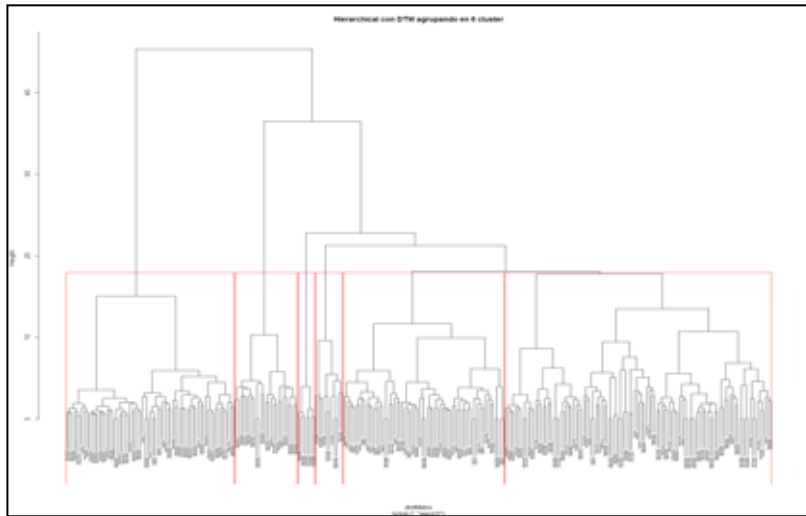


Figura F.11: Dendrograma. Jerárquico con distancia “DTW” , con 5 clusters

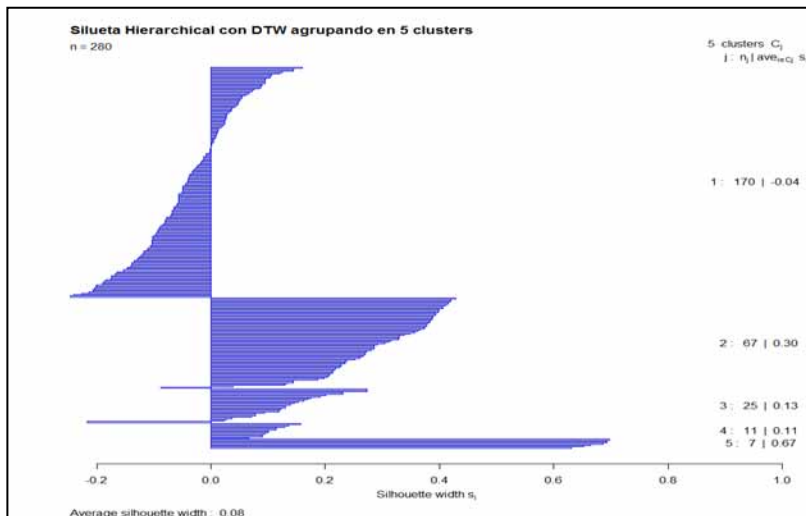


Figura F.12: “Silhouette” sobre jerárquico con distancia “DTW”, con 5 clusters

- Jerárquico con distancia “DTW” y agrupando en 6 clusters

Tabla F.5: Jerárquico con distancia “DTW” y agrupando con 6 clusters

Hierarchical con distancia “DTW”, aglomerando 6 clusters de tamaño: 106,64,67,25,11, 7						
Muestras	Grupos					
	1	2	3	4	5	6
DisparoReal	42	16	55	24	0	0
DisparoSalva	27	12	2	0	3	0
EfectoSonido	26	8	0	0	8	7
Explosivo	11	28	10	1	0	0
Tiempo de ejecución 126.42 segundos						

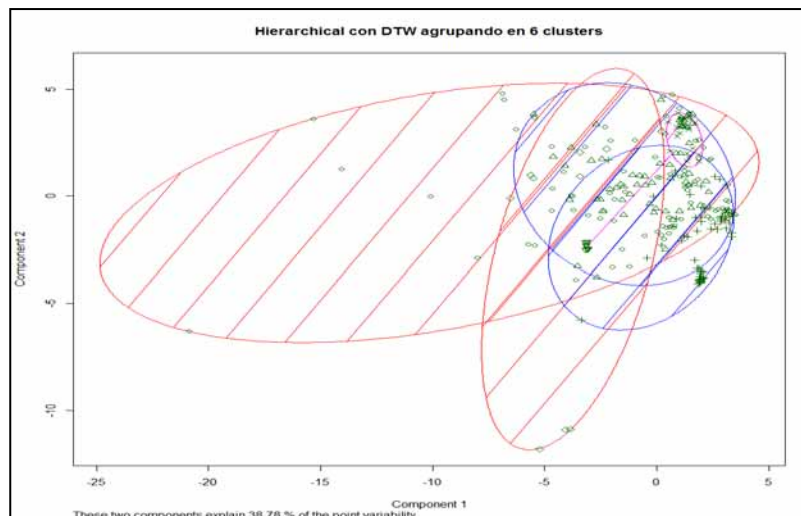


Figura E.55: Jerárquico con distancia “DTW”, con 6 clusters

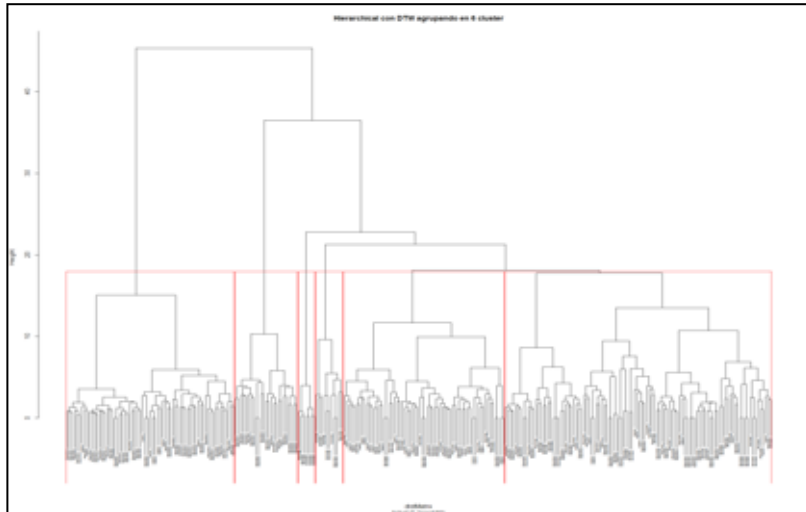


Figura E.56: Dendrograma. Jerárquico con distancia “DTW”, con 6 clusters

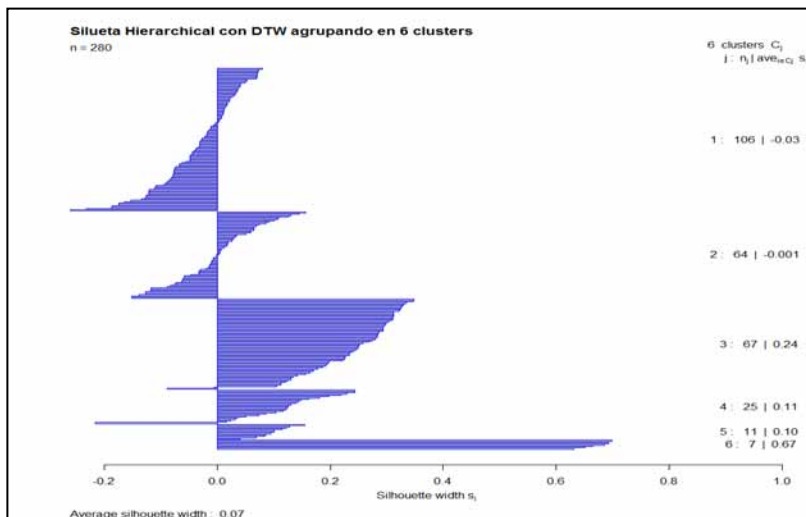


Figura E.57: “Silhouette” sobre jerárquico con distancia “DTW”, con 6 clusters

}

- Jerárquico con distancia “DTW” y agrupando en 7 clusters

Tabla F . 6: Jerárquico con distancia “DTW” y agrupando con 7 clusters

Hierarchical con distancia “DTW”, aglomerando 7 clusters de tamaño: 75, 64, 67, 25, 31, 11, 7							
Muestras	Grupos						
	1	2	3	4	5	6	7
DisparoReal	20	16	55	24	22	0	0
DisparoSalva	26	12	2	0	1	3	0
EfectoSonido	19	8	0	0	7	8	7
Explosivo	10	28	10	1	1	0	0
Tiempo de ejecución 127.43 segundos							

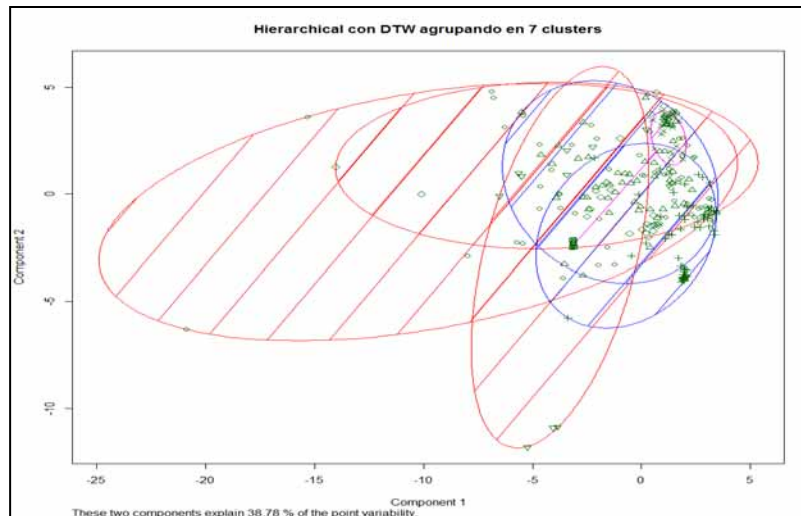


Figura F.13: Jerárquico con distancia “DTW”, con 7 clusters

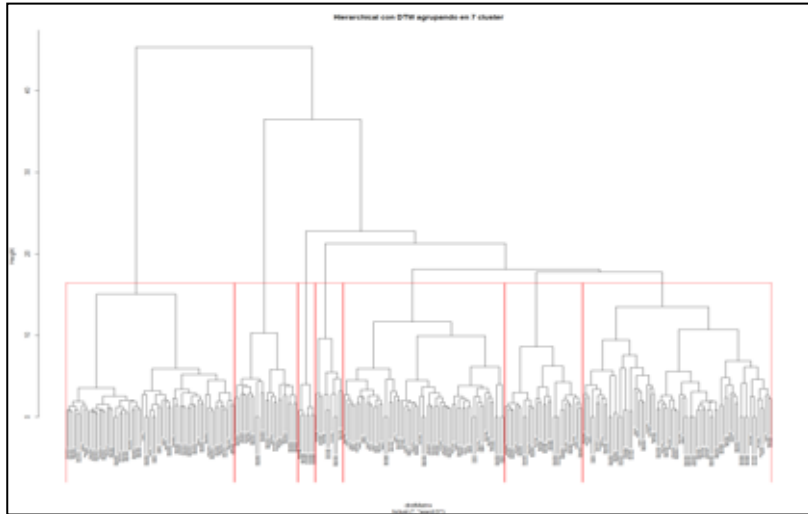


Figura F.14: Dendograma. Jerárquico con distancia “DTW”, con 7 clusters

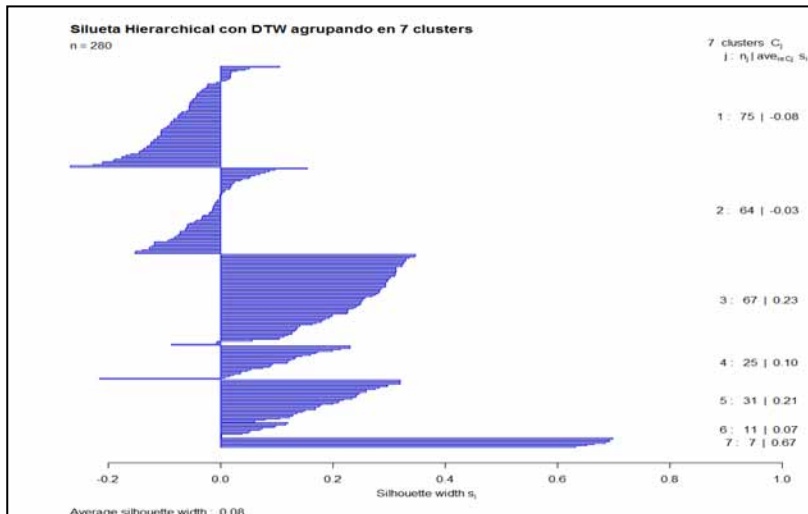


Figura F . 15: “Silhouette” sobre jerárquico con distancia “DTW”, con 7 clusters

- Jerárquico con distancia “DTW” y agrupando en 8 clusters

Tabla F.7: Jerárquico con distancia “DTW” y agrupando con 8 clusters

Hierarchical con distancia “DTW”, aglomerando 8 clusters de tamaño: 75,64,37,25,31,30,11,7								
Muestras	Grupos							
	1	2	3	4	5	6	7	8
DisparoReal	20	16	27	24	22	28	0	0
DisparoSalva	26	12	2	0	1	0	3	0
EfectoSonido	19	8	0	0	7	0	8	7
Explosivo	10	28	8	1	1	2	0	0
Tiempo de ejecución 122.76 segundos								

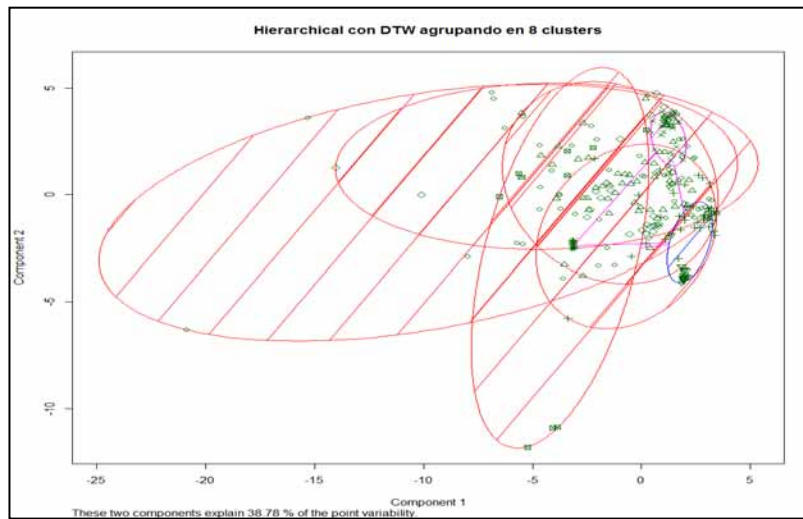


Figura F.16: Jerárquico con distancia “DTW”, con 8 clusters

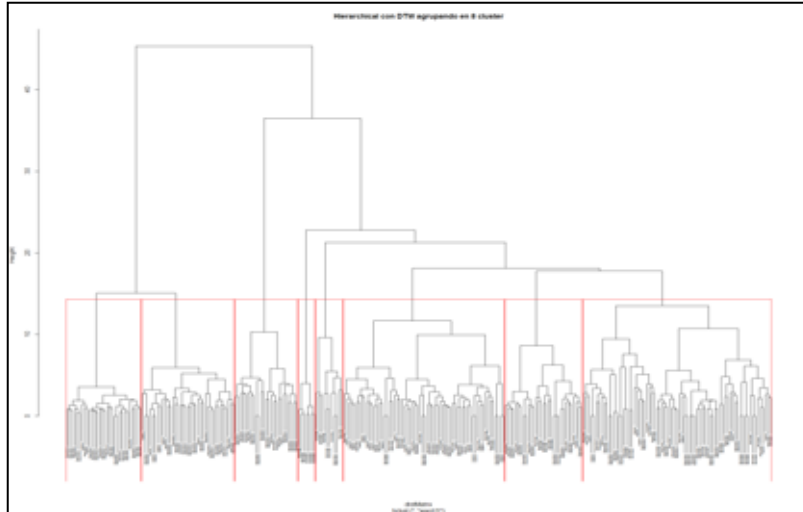


Figura F.17: Dendograma. Jerárquico con distancia “DTW” , con 8 clusters

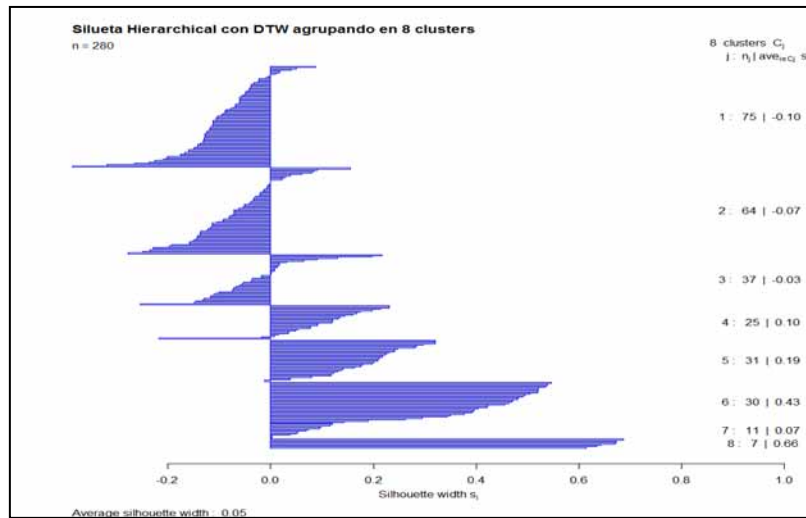


Figura F . 18: “Silhouette” sobre jerárquico con distancia “DTW “, con 8 clusters

- Jerárquico con distancia “DTW” y agrupando en 9 clusters

Tabla F.8: Jerárquico con distancia “DTW” y agrupando con 9 clusters

Hierarchical con distancia “DTW”, aglomerando 9 clusters de tamaño:6,64,37,25,29,31,30,11, 7									
Muestras	Grupos								
	1	2	3	4	5	6	7	8	9
DisparoReal	7	16	27	24	13	22	28	0	0
DisparoSalva	22	12	2	0	4	1	0	3	0
EfectoSonido	13	8	0	0	6	7	0	8	7
Explosivo	4	28	8	1	6	1	2	0	0
Tiempo de ejecución 124.84 segundos									

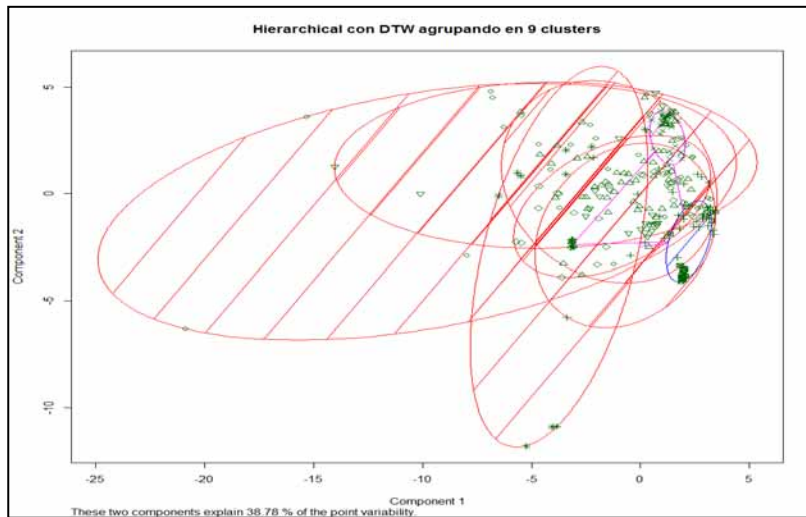


Figura F.19: Jerárquico con distancia “DTW”, con 9 clusters

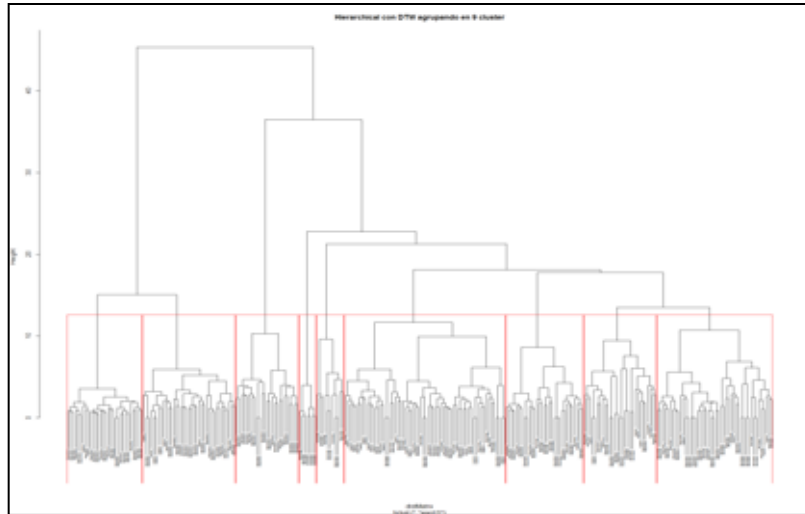


Figura F.20: Dendrograma. Jerárquico con distancia “DTW” , con 9 clusters

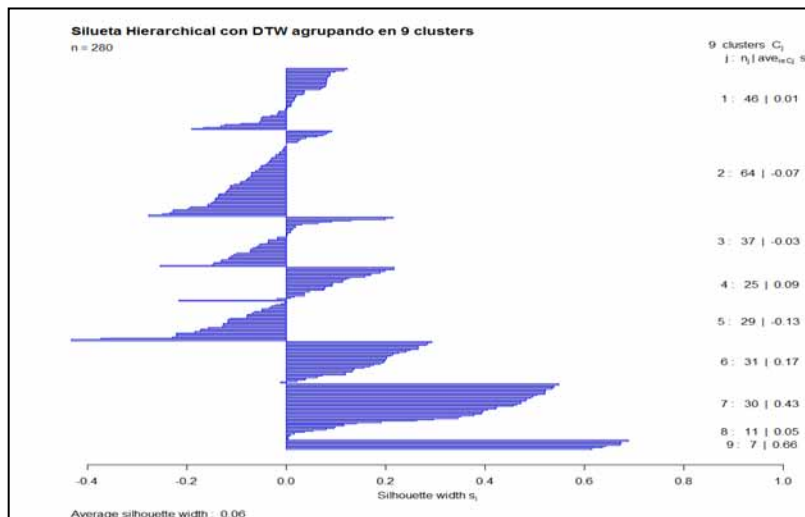


Figura F.21: “Silhouette” sobre jerárquico con distancia “DTW” , con 9 clusters

- Jerárquico con distancia “DTW” y agrupando en 10 clusters

Tabla F.9: Jerárquico con distancia “DTW” y agrupando con 10 clusters

Hierarchical con distancia “DTW”, aglomerando 10 clusters de tamaño:46,41,37,25,29,31,30,23, 11, 7										
Muestras	Grupos									
	1	2	3	4	5	6	7	8	9	10
DisparoReal	7	15	27	24	13	22	28	1	0	0
DisparoSalva	22	6	2	0	4	1	0	6	3	0
EfectoSonido	13	6	0	0	6	7	0	2	8	7
Explosivo	4	14	8	1	6	1	2	14	0	0
Tiempo de ejecución 125.13 segundos										

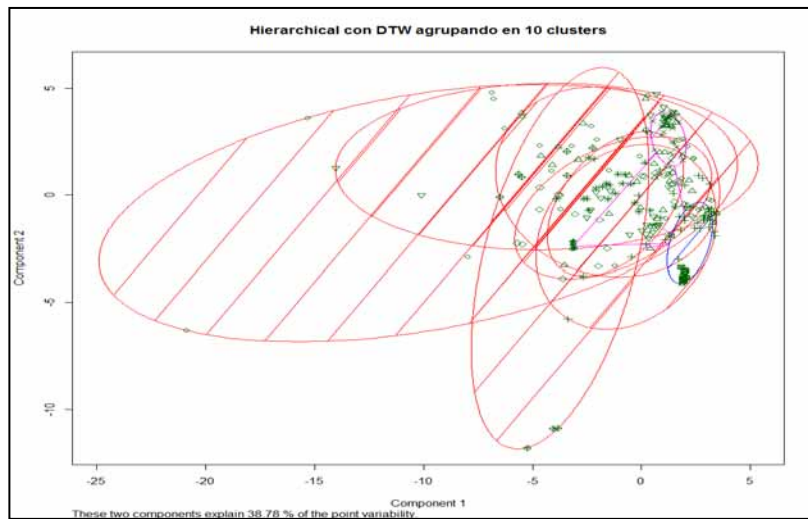


Figura F.22: Jerárquico con distancia “DTW”, con 10 clusters

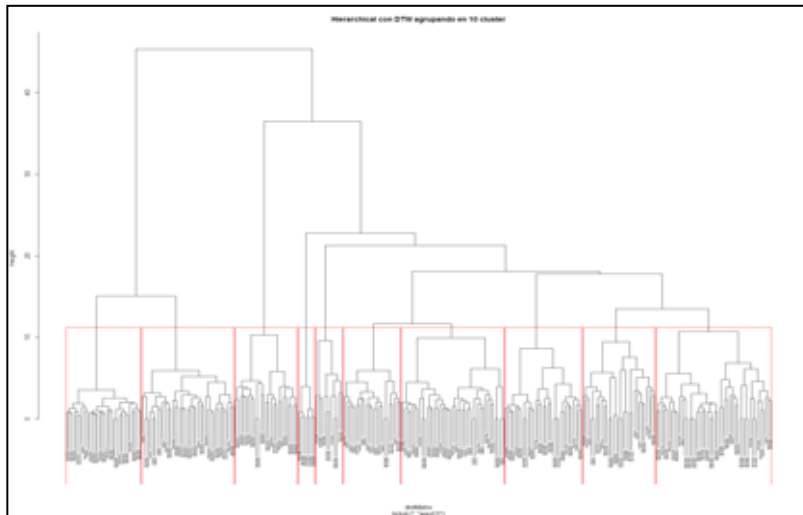


Figura F.23: Dendrograma. Jerárquico con distancia “DTW” , con 10 clusters

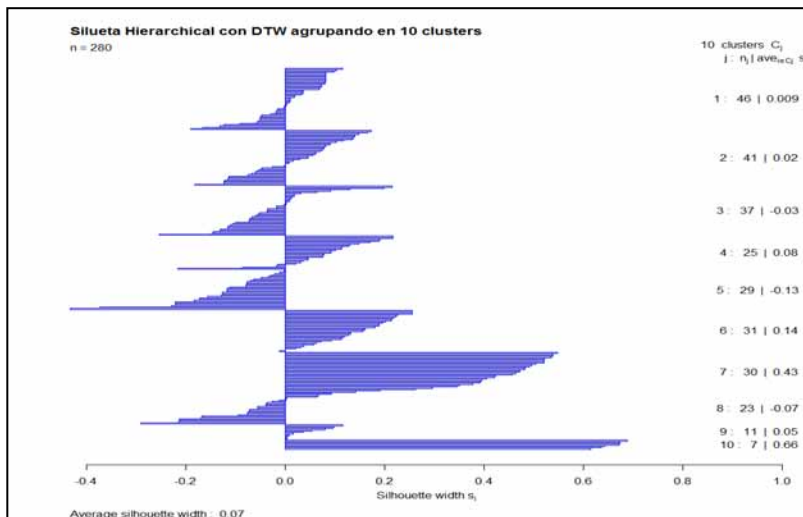


Figura F.24: “Silhouette” sobre jerárquico con distancia “DTW”, con 10 clusters

F.2 Resultados obtenidos al implementar el método Jerárquico con distancia “DTW” sobre 301 observaciones promediadas por bandas de frecuencia de 50 en 50 Hz

- Jerárquico con distancia “DTW” y agrupando en 2 clusters

Tabla F.10: Jerárquico con 2 clusters

Hierarchical con distancia “DTW”, aglomerando en 2 clusters de tamaño:119,182		
Muestras	Grupos	
	1	2
DisparoReal	74	78
DisparoSalva	29	21
EfectoSonido	0	49
Explosivo	16	34
Tiempo de ejecución 478.28 segundos		

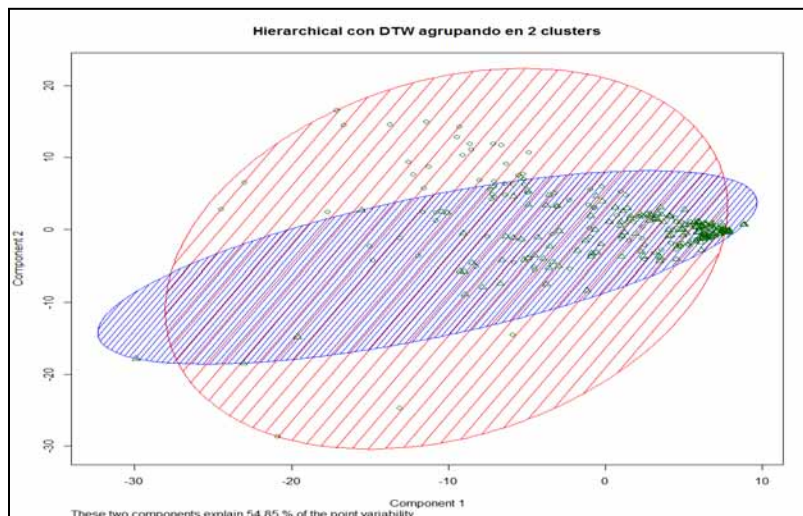


Figura F.25: Jerárquico con distancia “DTW”, con dos clusters

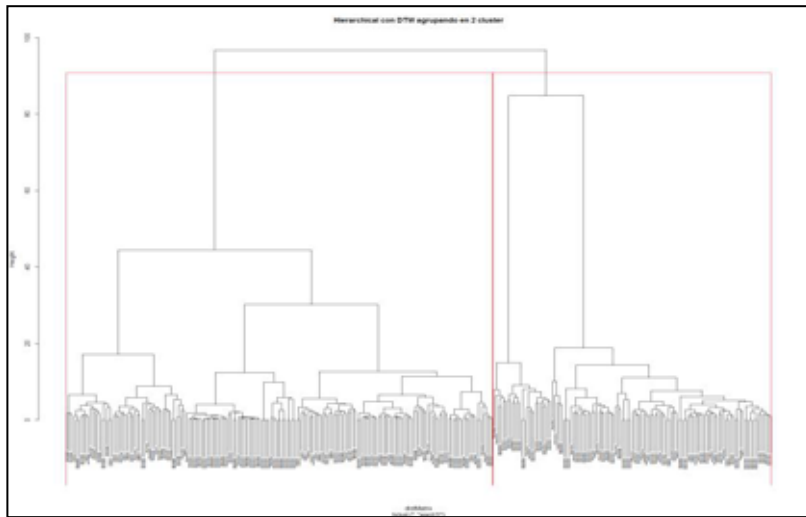


Figura F.26: Dendrograma. Jerárquico con distancia “DTW” con 2 clusters



Figura F.27: “Silhouette” sobre jerárquico con distancia “DTW”, con 2 clusters

- Jerárquico con distancia “DTW” y agrupando en 3 clusters

Tabla F.11: Jerárquico con distancia “DTW” y agrupando con 3 clusters

Hierarchical con distancia “DTW”, aglomerando 3 clusters de tamaño:94,182, 25			
Muestras	Grupos		
	1	2	3
DisparoReal	49	78	25
DisparoSalva	29	21	0
EfectoSonido	0	49	0
Explosivo	16	34	0
Tiempo de ejecución 486.84 segundos			

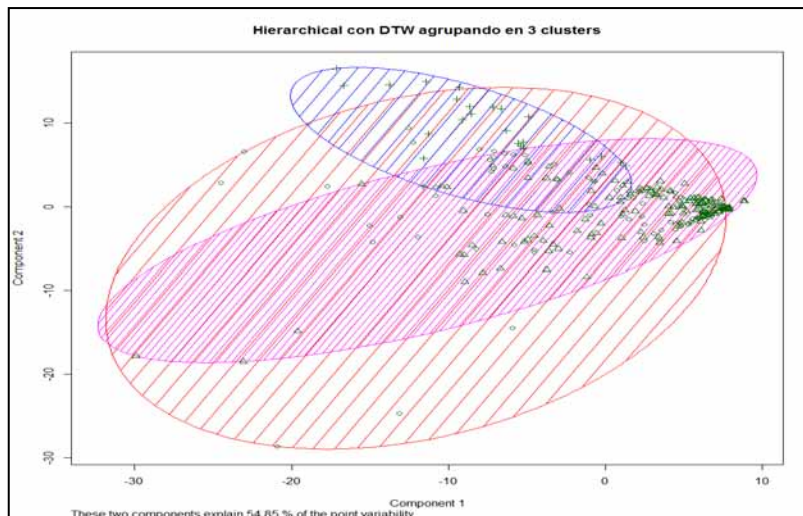


Figura F.28: Jerárquico con distancia “DTW”, con 3 clusters

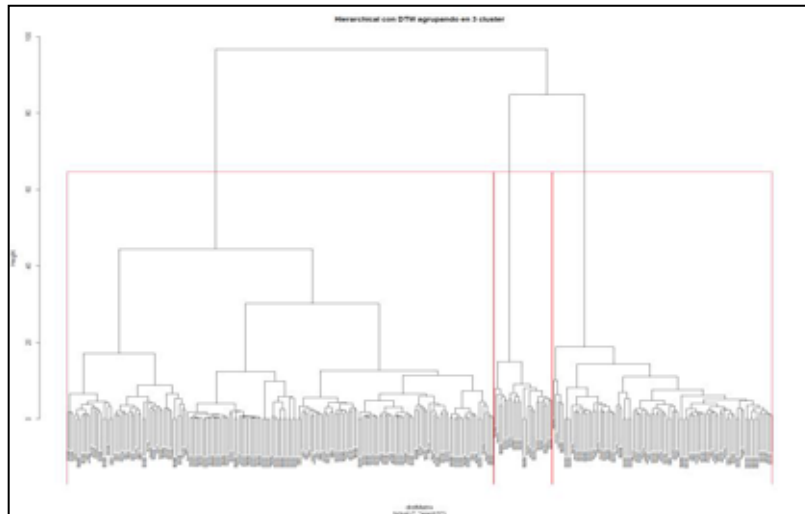


Figura F.29: Dendrograma. Jerárquico con distancia “DTW” , con 3 clusters

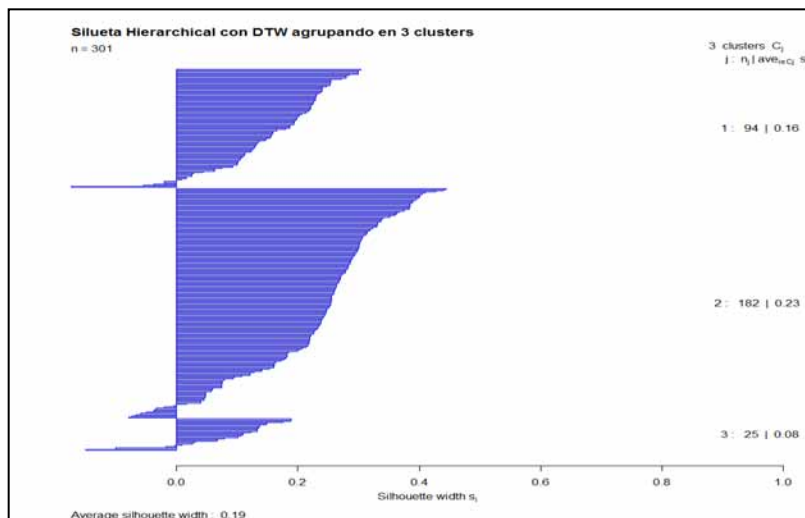


Figura F . 30: “Silhouette” sobre jerárquico con distancia “DTW “,con 3 clusters

- Jerárquico con distancia “DTW” y agrupando en 4 clusters

Tabla F.12: Jerárquico con distancia “DTW” y agrupando con 4 clusters

Hierarchical con distancia “DTW”, aglomerando 4 clusters de tamaño: 94,131,25, 51				
Muestras	Grupos			
	1	2	3	4
DisparoReal	49	76	25	2
DisparoSalva	29	9	0	12
EfectoSonido	0	28	0	21
Explosivo	16	18	0	16
Tiempo de ejecución 485.48 segundos				

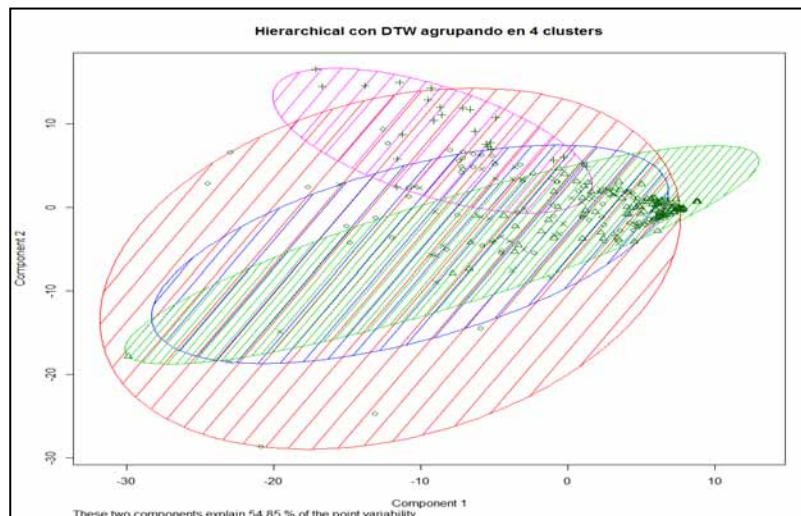


Figura F.31: Jerárquico con distancia “DTW”, con 4 clusters

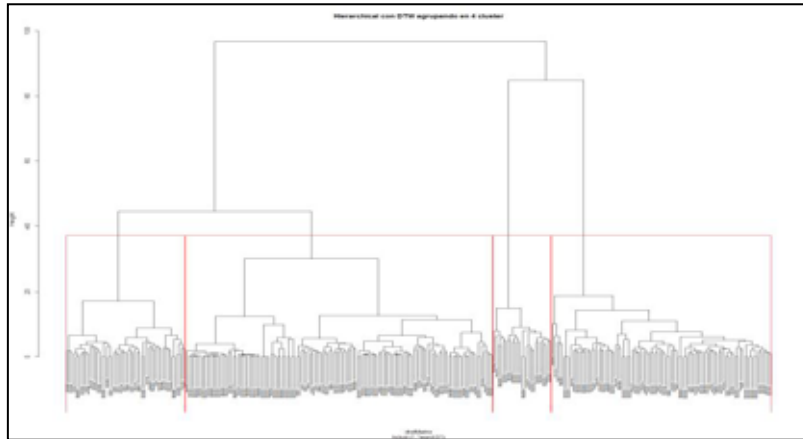


Figura F.32: Dendrograma. Jerárquico con distancia “DTW” , con 4 clusters

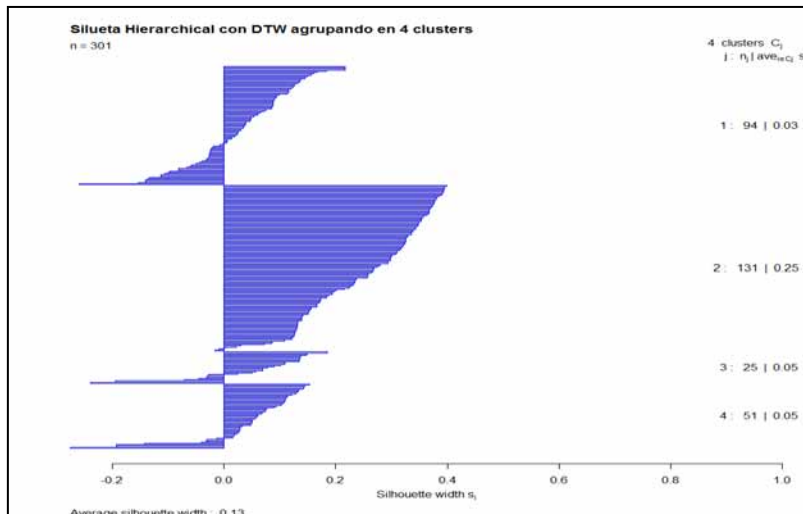


Figura F.33: “Silhouette” sobre jerárquico con distancia “DTW”, con 4 clusters

- Jerárquico con distancia “DTW” y agrupando en 5 clusters

Tabla F.13: Jerárquico con distancia “DTW” y agrupando con 5 clusters

Hierarchical con distancia “DTW”, aglomerando 5 clusters de tamaño: 94,84,47,25,51					
Muestras	Grupos				
	1	2	3	4	5
DisparoReal	49	29	47	25	2
DisparoSalva	29	9	0	0	12
EfectoSonido	0	28	0	0	21
Explosivo	16	18	0	0	16
Tiempo de ejecución 484.12					

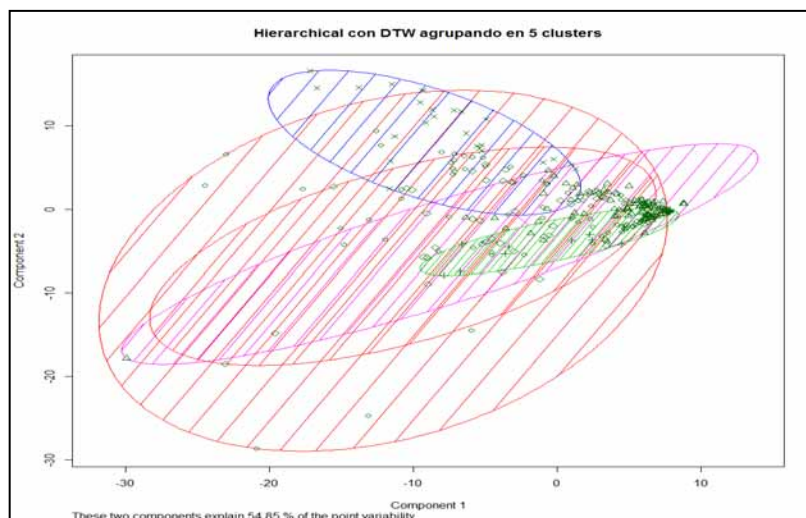


Figura F.34: Jerárquico con distancia “DTW”, con 5 clusters

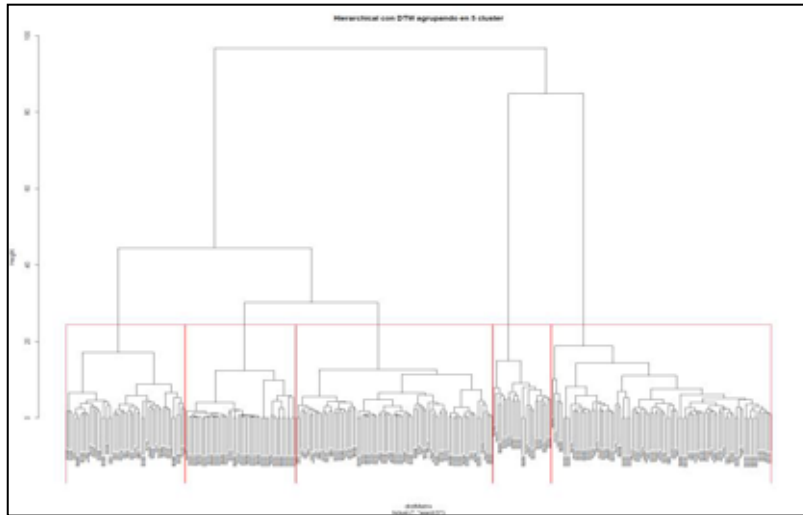


Figura F.35: Dendograma. Jerárquico con distancia “DTW” , con 5 clusters

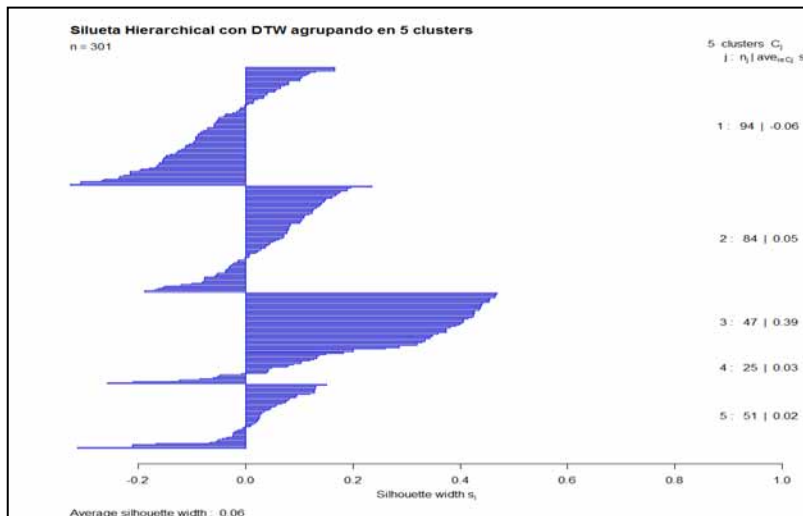


Figura F.36: “Silhouette” sobre jerárquico con distancia “DTW”, con 5 clusters

- Jerárquico con distancia “DTW” y agrupando en 6 clusters

Tabla F.14: Jerárquico con distancia “DTW” y agrupando con 6 clusters

Hierarchical con distancia “DTW”, aglomerando 6 clusters de tamaño: 5, 84, 47, 25, 89, 51						
Muestras	Grupos					
	1	2	3	4	5	6
DisparoReal	4	29	47	25	45	2
DisparoSalva	0	9	0	0	29	12
EfectoSonido	0	28	0	0	0	21
Explosivo	1	18	0	0	15	16
Tiempo de ejecución 480.33segundos						

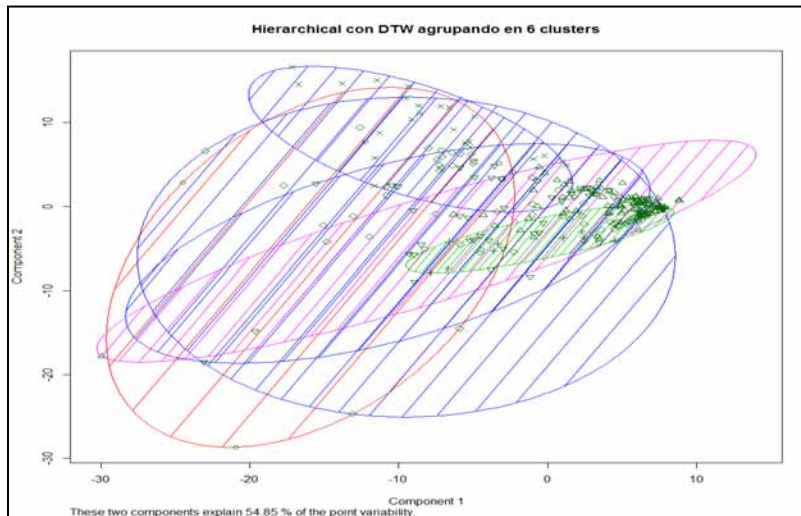


Figura F.37: Jerárquico con distancia “DTW”, con 6 clusters

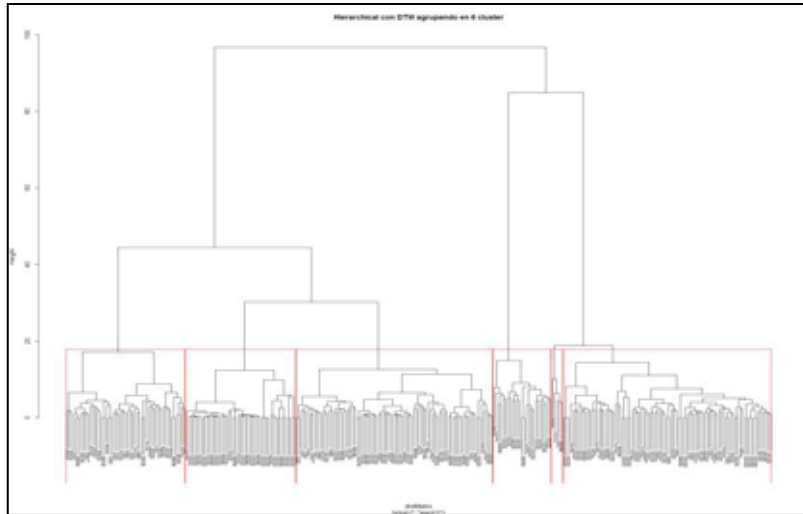


Figura F.38: Dendrograma. Jerárquico con distancia “DTW” , con 6 clusters

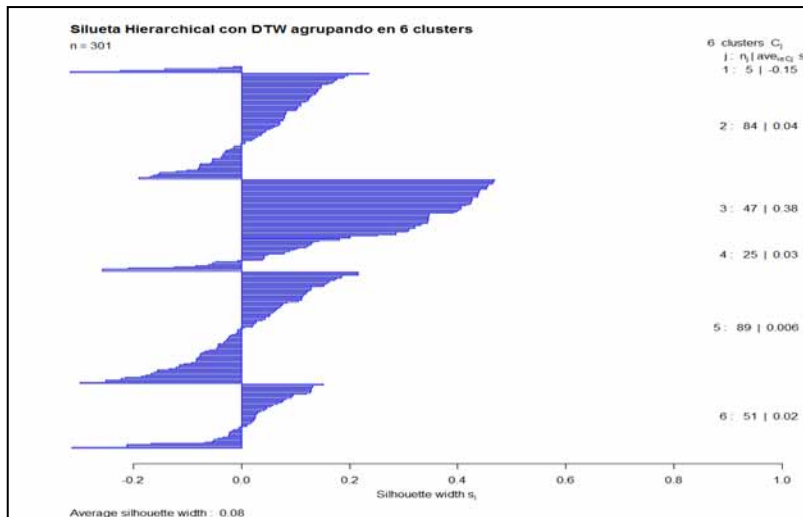


Figura F . 39: “Silhouette” sobre jerárquico con distancia “DTW “, con 6 clusters

- Jerárquico con distancia “DTW” y agrupando en 7 clusters

Tabla F.15: Jerárquico con distancia “DTW” y agrupando con 7 clusters

Hierarchical con distancia “DTW”, aglomerando 7 clusters de tamaño: 5,84,47,25,89,20,31							
Muestras	Grupos						
	1	2	3	4	5	6	7
DisparoReal	4	29	47	25	45	2	0
DisparoSalva	0	9	0	0	29	2	10
EfectoSonido	0	28	0	0	0	21	9
Explosivo	1	18	0	0	15	4	12
Tiempo de ejecución 480.24segundos							

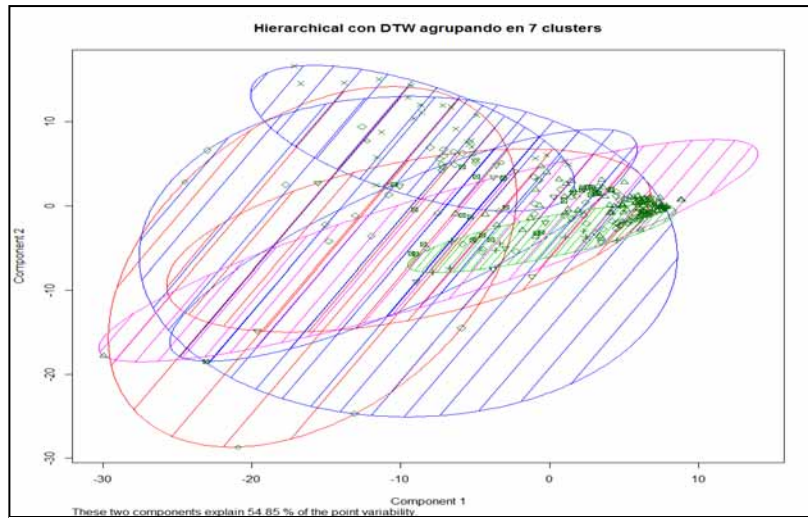


Figura F.40: Jerárquico con distancia “DTW”, con 7 clusters

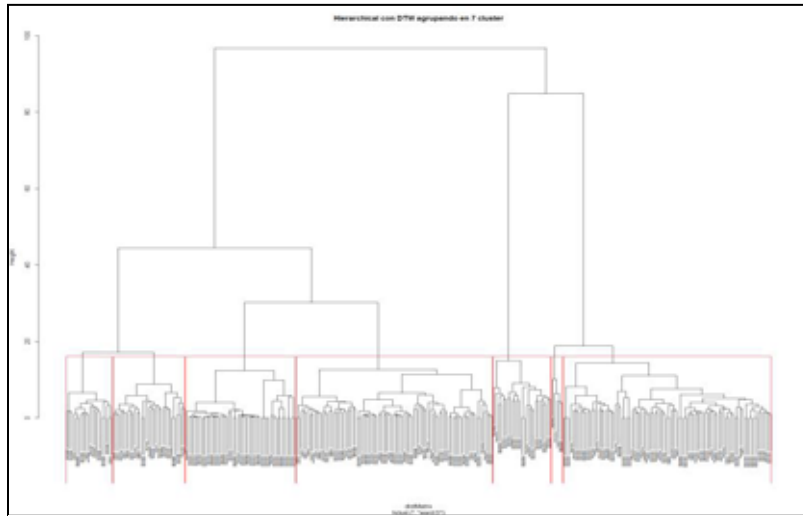


Figura F.41: Dendrograma. Jerárquico con distancia “DTW” , con 7 clusters

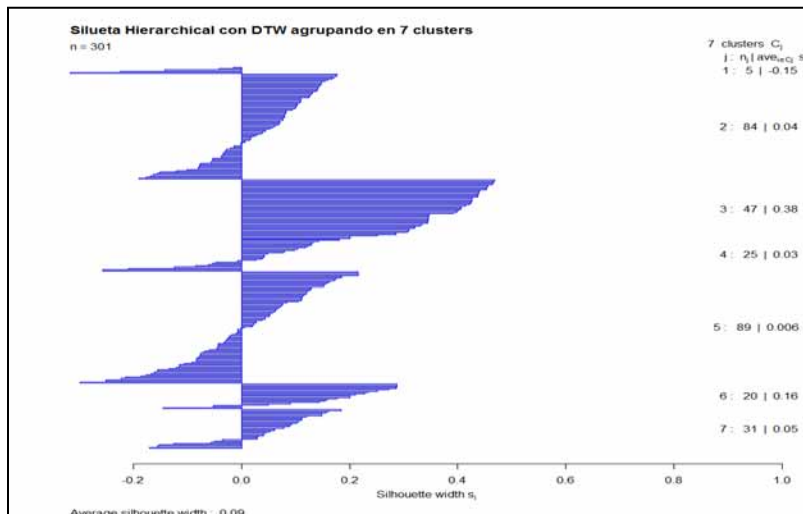


Figura F.42: “Silhouette” sobre jerárquico con distancia “DTW “, con 7 clusters

- Jerárquico con distancia “DTW” y agrupando en 8 clusters

Tabla F.16: Jerárquico con distancia “DTW” y agrupando con 8 clusters

Hierarchical con distancia “DTW”, aglomerando 8 clusters de tamaño: 5,84,47,19,6,89,20,31								
Muestras	Grupos							
	1	2	3	4	5	6	7	8
DisparoReal	4	29	47	19	6	45	2	0
DisparoSalva	0	9	0	0	0	29	2	10
EfectoSonido	0	28	0	0	0	0	12	9
Explosivo	1	18	0	0	0	15	4	12
Tiempo de ejecución 474.84segundos								

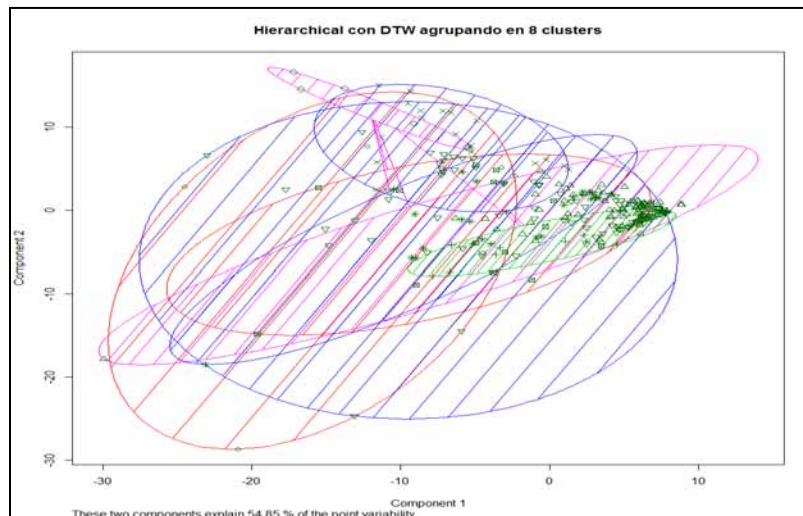


Figura F.43: Jerárquico con distancia “DTW”, con 8 clusters

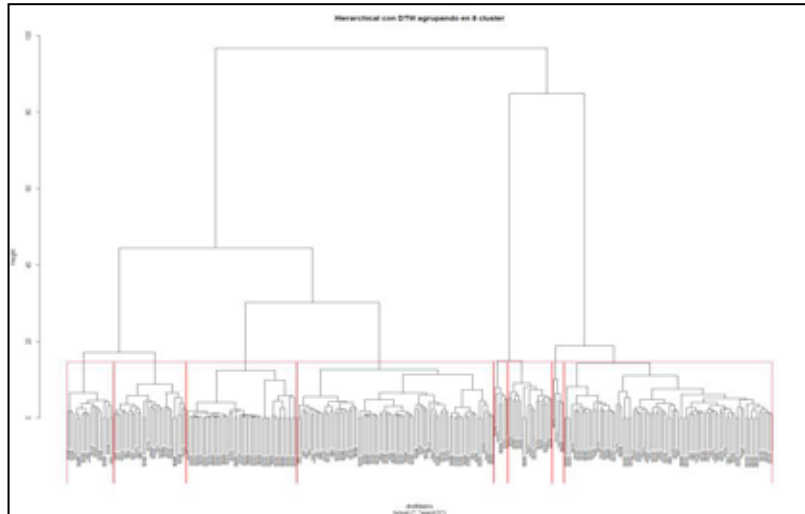


Figura F.44: Dendograma. Jerárquico con distancia “DTW” , con 8 clusters

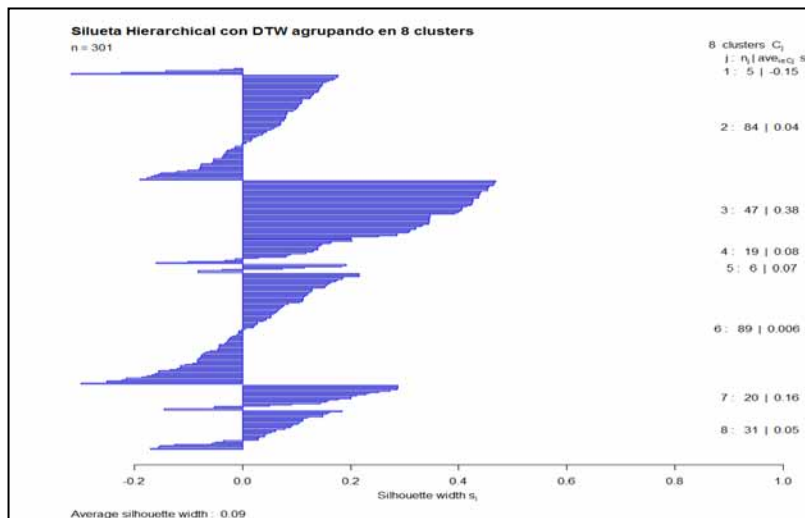


Figura F . 45: “Silhouette” sobre jerárquico con distancia “DTW “, con 8 clusters

- Jerárquico con distancia “DTW” y agrupando en 9 clusters

Tabla F.17: Jerárquico con distancia “DTW” y agrupando con 9 clusters

Hierarchical con distancia “DTW”, aglomerando 9 clusters de tamaño: 5,84,47,19,6,22,67,20,31									
Muestras	Grupos								
	1	2	3	4	5	6	7	8	9
DisparoReal	4	29	47	19	6	21	24	2	0
DisparoSalva	0	9	0	0	0	0	29	2	10
EfectoSonido	0	28	0	0	0	0	0	12	9
Explosivo	1	18	0	0	0	1	14	4	12
Tiempo de ejecución 475.97segundos									

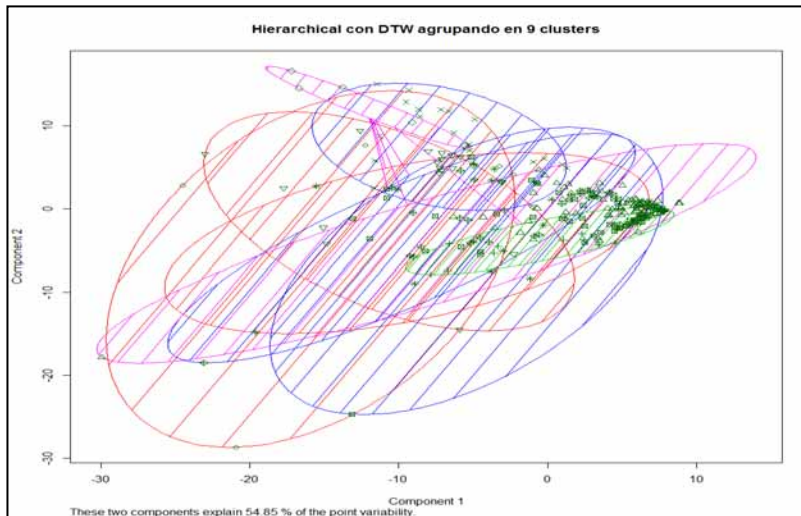


Figura F.46: Jerárquico con distancia “DTW”, con 9 clusters

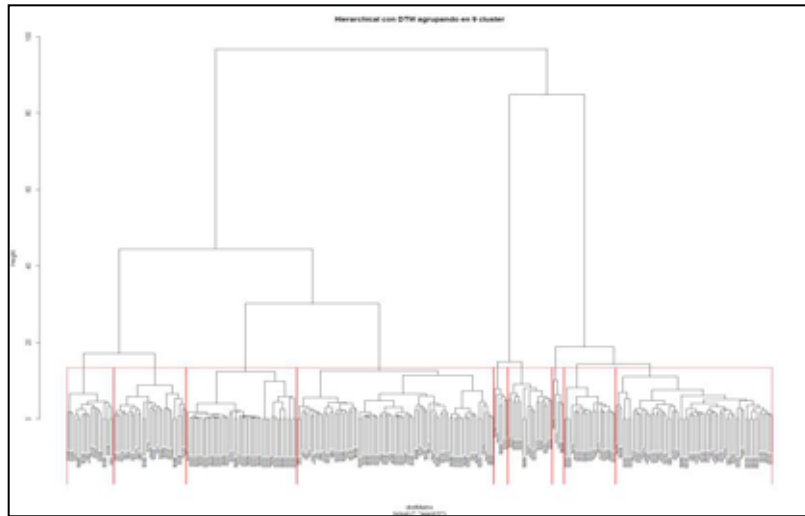


Figura F.47: Dendrograma. Jerárquico con distancia “DTW”, con 9 clusters

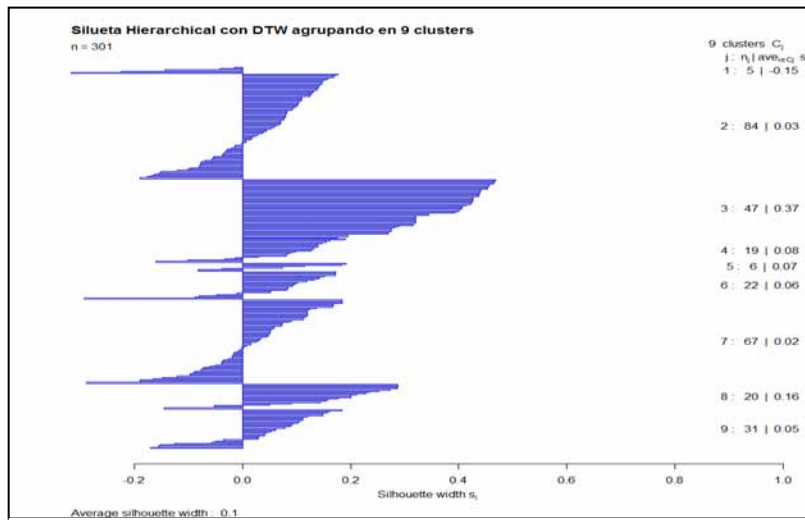


Figura F.48: “Silhouette” sobre jerárquico con distancia “DTW”, con 9 clusters

- Jerárquico con distancia “DTW” y agrupando en 10 clusters

Tabla F.18: Jerárquico con distancia “DTW” y agrupando con 10 clusters

Hierarchical con distancia “DTW”, aglomerando 10 clusters de tamaño: 5,58,47,19,6,22,67,26,20, 31										
Muestras	Grupos									
	1	2	3	4	5	6	7	8	9	10
DisparoReal	4	23	47	19	6	21	24	6	2	0
DisparoSalva	0	5	0	0	0	0	29	4	2	10
EfectoSonido	0	25	0	0	0	0	0	3	12	9
Explosivo	1	5	0	0	0	1	14	13	4	12
Tiempo de ejecución 482.62segundos										

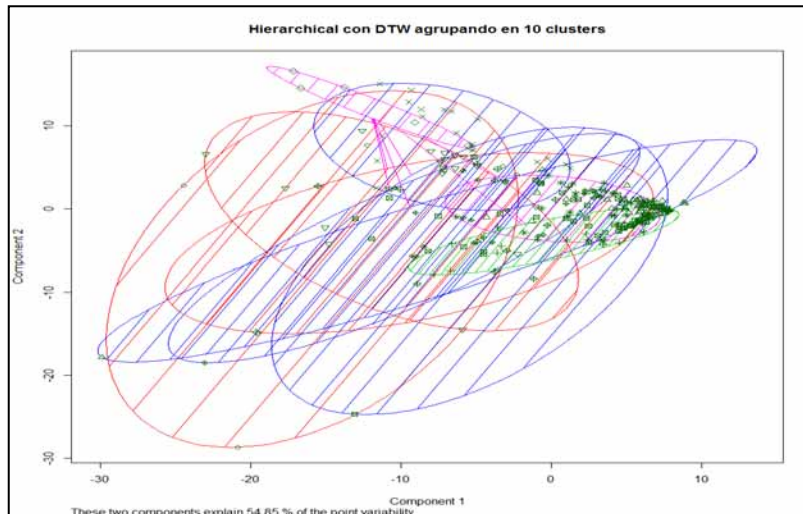


Figura F.49: Jerárquico con distancia “DTW”, con 10 clusters

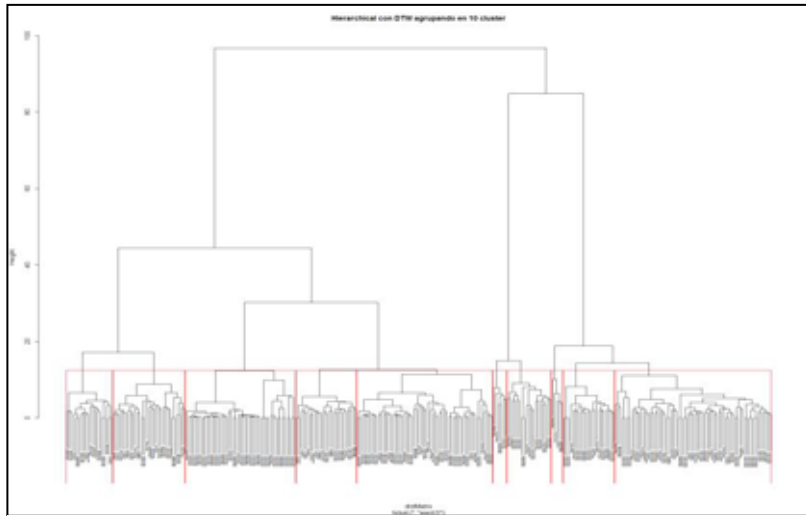


Figura F.50: Dendograma. Jerárquico con distancia “DTW” , con 10 clusters

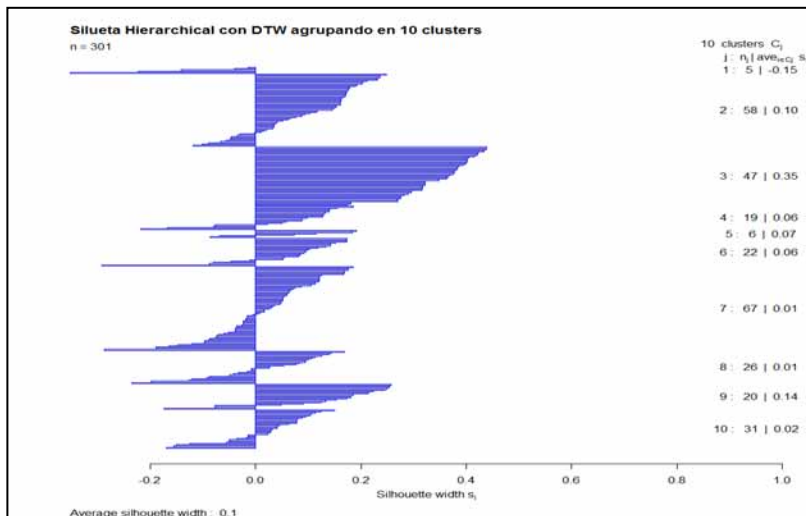


Figura F.51: “Silhouette” sobre jerárquico con distancia “DTW”, con 10 clusters

Apéndice G

Resultados obtenidos al implementar el método “SVM” del inglés “Support Vector Machines”

G.1 Resultados obtenidos al implementar el método “SVM” sobre 280 observaciones promediadas en escala logarítmica

- Método *SVM* con *kernel* lineal

La **¡Error! No se encuentra el origen de la referencia.** muestra un resumen de la clasificación obtenida a través del *SVM* con *kernel* lineal, así también los vectores de soporte, que son los puntos que tocan el límite del margen y por otro lado la **¡Error! No se encuentra el origen de la referencia.** exhibe la precisión que es la proporción del número de predicciones que son correctas, la sensibilidad que representa la proporción de malos riesgos predichos como malos, F y la exactitud que indica la probabilidad de obtener resultados correctos en la prueba.

Tabla G.1: SVM con kernel lineal

Clasificación con el algoritmo SVM utilizando kernel lineal				
Muestras	DisparoRelal	DisparoSalva	EfectosSonido	Explosivo
DisparoReal	133	2	0	2
DisparoSalva	1	43	0	0
EfectoSonido	0	0	49	0
Explosivo	4	1	0	45
Number of Support Vectors: 120 (40 26 33 21)				
Number of Classes: 4				
Levels: DisparoReal DisparoSalva EfectoSonido Explosivo				
Tiempo de ejecución 0.42 segundos				

Tabla G. 2: Evaluación del método svm con kernel lineal

Medidas	Disparos reales	Disparos de salva	Efectos de sonido	explosivo
precisión	0.97080292	0.97727273	1	0.9
sensibilidad	0.96376812	0.93478261	1	0.95744681
F	0.96727273	0.95555556	1	0.92783505
Exactitud	0.96428571			

- **Método SVM con kernel polinomial**

Tabla G. 3: Clasificación SVM usando kernel polinomial

Clasificación con el algoritmo SVM utilizando kernel polynomial				
Muestras	Grupos			
	DisparoRelal	DisparoSalva	EfectosSonido	Explosivo
DisparoReal	137	0	0	0
DisparoSalva	40	4	0	0
EfectoSonido	5	0	44	0
Explosivo	33	0	1	16
Number of Support Vectors: 192 (70 44 49 29)				
Number of Classes: 4				
Levels:				
DisparoReal DisparoSalva EfectoSonido Explosivo				
Tiempo de ejecución 0.42 segundos				

Tabla G. 4: Evaluación del método svm con kernel polinomial

Medidas	Disparos reales	Disparos salva	efectos	explosivo
precisión	1	0.09090909	0.89795918	0.32
sensibilidad	0.6372093	1	0.97777778	1
F	0.77840909	0.16666667	0.93617021	0.48484848
Exactitud	0.71785714			

- **Método SVM con kernel radial Gaussiana**

Tabla G. 5: Clasificación SVM usando kernel radial Gaussiana

Clasificación con el algoritmo SVM utilizando kernel radial Gaussiana				
Muestras	DisparoRelal	DisparoSalva	EfectosSonido	Explosivo
DisparoReal	135	1	0	1
DisparoSalva	5	38	1	0
EfectoSonido	1	0	48	0
Explosivo	6	3	1	40
Number of Support Vectors: 195 (69 43 49 34)				
Number of Classes: 4				
Levels: DisparoReal DisparoSalva EfectoSonido Explosivo				
Tiempo de ejecución 0.49 segundos				

Tabla G. 6: Evaluación del método svm con kernel radial Gaussiana

Medidas	Disparos reales	Disparos salva	efectos	explosivo
precisión	0.98540146	0.86363636	0.97959184	0.8
sensibilidad	0.91836735	0.9047619	0.96	0.97560976
F	0.95070423	0.88372093	0.96969697	0.87912088
Exactitud	0.93214286			

- **Método SVM con kernel sigmoidal**

Tabla G. 7: Clasificación SVM usando kernel sigmoidal

Clasificación con el algoritmo SVM utilizando kernel sigmoid				
Muestras	DisparoRelal	DisparoSalva	EfectosSonido	Explosivo
DisparoReal	133	2	0	2
DisparoSalva	29	11	2	2
EfectoSonido	2	0	43	4
Explosivo	22	4	8	16
Number of Support Vectors: 190 (64 44 49 33)				
Number of Classes: 4				
Levels: DisparoReal DisparoSalva EfectoSonido Explosivo				
Tiempo de ejecución 0.35 segundos				

Tabla G. 8: Evaluación del método svm con kernel sigmoidal

Medidas	Disparos reales	Disparos salva	efectos	explosivo
precisión	0.97080292	0.25	0.87755102	0.32
sensibilidad	0.71505376	0.64705882	0.81132075	0.66666667
F	0.82352941	0.36065574	0.84313725	0.43243243
Exactitud	0.725			

G.2 Resultados obtenidos al implementar el método “SVM” sobre 301 observaciones promediadas por bandas de frecuencia de 50 en 50 Hz

- Clasificación empleando el método *SVM* usando un *kernel* lineal

Tabla G. 9: Clasificación SVM usando kernel lineal

Clasificación con el algoritmo SVM utilizando kernel lineal				
	DisparoReal	DisparoSalva	EfectoSonido	Explosivo
DisparoReal	152	0	0	0
DisparoSalva	0	50	0	0
EfectoSonido	0	0	49	0
Explosivo	1	1	0	48
Number of Support Vectors: 151 (51 30 38 32)				
Number of Classes: 4				
Levels:				
DisparoReal DisparoSalva EfectoSonido Explosivo				
Tiempo de ejecución 0.75 segundos				

Tabla G. 10: Evaluación del método svm con kernel lineal

Medidas	Disparos reales	Disparos salva	efectos	explosivo
precisión	1	1	1	0.96
sensibilidad	0.99346405	0.98039216	1	1
F	0.99672131	0.99009901	1	0.97959184
Exactitud	0.99335548			

- **Clasificación empleando el método SVM usando un kernel polinomial**

Tabla G. 11: Clasificación SVM usando kernel polinomial

Clasificación con el algoritmo SVM utilizando kernel polynomial				
	DisparoReal	DisparoSalva	EfectoSonido	Explosivo
DisparoReal	152	0	0	0
DisparoSalva	40	10	0	0
EfectoSonido	37	0	12	0
Explosivo	44	0	0	6
Number of Support Vectors: 237 (98 45 49 45) Number of Classes: 4 Levels: DisparoReal DisparoSalva EfectoSonido Explosivo				

Tabla G. 12: Evaluación del método svm con kernel polinomial

Medidas	Disparos reales	Disparos salva	efectos	explosivo
precisión	1	0.2	0.24489796	0.12
sensibilidad	0.55677656	1	1	1
F	0.71529412	0.33333333	0.39344262	0.21428571
Exactitud	0.59800664			

- **Clasificación empleando el algoritmo SVM usando un *kernel* radial Gaussiana**

Tabla G. 13: Clasificación SVM usando *kernel* radial Gaussiana

Clasificación con el algoritmo SVM utilizando kernel radial Gaussiana				
	DisparoReal	DisparoSalva	EfectoSonido	Explosivo
DisparoReal	149	3	0	0
DisparoSalva	4	45	0	1
EfectoSonido	13	1	35	0
Explosivo	15	2	2	31
Number of Support Vectors: 242 (100 47 47 48)				
Number of Classes: 4				
Levels:				
DisparoReal DisparoSalva EfectoSonido Explosivo				
Tiempo de ejecución 0.94 segundos				

Tabla G. 14: Evaluación del método svm con *kernel* radial Gaussiana

Medidas	Disparos reales	Disparos salva	efectos	explosivo
precisión	0.98026316	0.9	0.71428571	0.62
sensibilidad	0.82320442	0.88235294	0.94594595	0.96875
F	0.89489489	0.89108911	0.81395349	0.75609756
Exactitud	0.86378738			

- **Clasificación empleando el algoritmo SVM usando un *kernel* sigmoidal**

Tabla G. 15: Clasificación SVM usando kernel sigmoidal

Clasificación con el algoritmo SVM utilizando kernel sigmoid				
	DisparoReal	DisparoSalva	EfectoSonido	Explosivo
DisparoReal	137	7	8	0
DisparoSalva	26	21	1	2
EfectoSonido	32	0	12	5
Explosivo	28	7	1	14
Number of Support Vectors: 223 (75 49 50 49)				
Number of Classes: 4				
Levels: DisparoReal DisparoSalva EfectoSonido Explosivo				
Tiempo de ejecución 1.08 segundos				

Tabla G. 16: Evaluación del método svm con kernel sigmoidal

Medidas	Disparos reales	Disparos salva	efectos	explosivo
precisión	0.90131579	0.42	0.24489796	0.28
sensibilidad	0.61434978	0.6	0.54545455	0.66666667
F	0.73066667	0.49411765	0.33802817	0.3943662
Exactitud	0.61129568			

