

**UNIVERSIDAD DE LAS PALMAS DE GRAN
CANARIA**

**ESCUELA DE INGENIERÍA DE
TELECOMUNICACIÓN Y ELECTRÓNICA**



PROYECTO FIN DE CARRERA

**ESTUDIO Y DESARROLLO DE LA CARACTERIZACIÓN
DE FIRMAS ONLINE MEDIANTE TRAZOS AÉREOS**

Titulación: Ingeniero de Telecomunicación

Autor: Aysse Ortega Pérez

Tutor: Dr. Carlos Manuel Travieso González
Elyoenai Guerra Segura

Fecha: Julio de 2017

UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA

ESCUELA DE INGENIERÍA DE TELECOMUNICACIÓN Y ELECTRÓNICA



PROYECTO FIN DE CARRERA

ESTUDIO Y DESARROLLO DE LA CARACTERIZACIÓN
DE FIRMAS ONLINE MEDIANTE TRAZOS AÉREOS

HOJA DE FIRMAS

Alumno/a

Fdo.: Aysse Ortega Pérez

Tutor/a

Tutor/a

Fdo.: Dr. Carlos Manuel Travieso González

Fdo.: Elyoenai Guerra Segura

Fecha: Julio de 2017

UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA

ESCUELA DE INGENIERÍA DE TELECOMUNICACIÓN Y ELECTRÓNICA



PROYECTO FIN DE CARRERA

ESTUDIO Y DESARROLLO DE LA CARACTERIZACIÓN DE FIRMAS ONLINE MEDIANTE TRAZOS AÉREOS

HOJA DE EVALUACIÓN

Calificación: _____

Presidente

Fdo.:

Vocal

Secretario/a

Fdo.:

Fdo.:

Fecha: Julio de 2017

Índice

Capítulo 1. Introducción	1
1.1. Definición de biometría	1
1.2. Firma biométrica	2
1.3. Leap Motion	3
1.4. Antecedentes	6
1.5. Novedad de la propuesta	6
1.6. Objetivos	8
1.7. Organización de la memoria	9
Capítulo 2. Base de datos	11
2.1. Descripción de la base de datos proporcionada	11
2.2. Ampliación de la base de datos	12
2.2.1. Captura de datos	12
2.2.2. Datos capturados	18
2.2.3. Almacenamiento en la base de datos	23
2.3. Características de los usuarios	26
Capítulo 3. Sistema de parametrización y clasificación	29
3.1. Parametrización	29
3.2. Parámetros	31
3.3. Descripción de un clasificador	33
3.4. Clasificador implementado	33
3.5. Máquinas de Soporte Vectorial	33
3.6. Clasificador seleccionado. LS-SVM	38
Capítulo 4. Metodología	41
4.1. Medidas para evaluar el rendimiento	41
4.2. Experimentos para el estudio del clasificador	42
Experimento 1. Análisis en detalle con el clasificador	43
Experimento 2. Ajuste del clasificador y selección del <i>kernel</i>	44
Experimento 3. Estudio de los datos añadiendo nuevos parámetros estadísticos .45	
Experimento 4. Estudio de la influencia de la tercera dimensión con respecto al sistema plano	46
Experimento 5. Fusión de los conjuntos	46
Experimento 6. Simulación del sistema con una base de datos ampliada a 70 usuarios	47
Experimento 7. Simulación con la base de datos final	47
Capítulo 5. Resultados	49
Experimento 1. Análisis en detalle con el clasificador	49

Experimento 2. Ajuste del clasificador y selección del <i>kernel</i>	51
Experimento 2.1. <i>Kernel</i> polinómico.....	51
Experimento 2.2. <i>Kernel</i> RBF.....	58
Experimento 3. Estudio de los datos añadiendo nuevos parámetros estadísticos.	61
Experimento 4. Estudio de la influencia de la tercera dimensión con respecto al sistema plano.	64
Experimento 5. Fusión de los conjuntos.	70
Experimento 6. Simulación del sistema con una base de datos ampliada a 70 usuarios.	71
Experimento 7. Simulación del sistema con la base de datos final de 100 usuarios.	74
Capítulo 6. Conclusiones y líneas futuras	81
6.1. Conclusiones	81
6.2. Comparación de resultados	83
6.3. Líneas futuras	84
Anexo A. Tablas de resultados	85
A.1. Resultados del experimento 2	85
A.2. Resultados del experimento 3	89
A.3. Resultados del experimento 4	91
A.4. Resultados del experimento 6 y del experimento 7.....	97
Anexo B. Boxplots	99
Bibliografía	111
PLANOS Y PROGRAMAS	115
PP.1. Distribución	115
PP.2. Algoritmos para la extracción de subconjuntos con sus respectivas características	116
PP.3. Algoritmo de fusión de subconjuntos	117
PP.4. Algoritmos de clasificación	118
PLIEGO DE CONDICIONES	121
PC.1. Requisitos hardware	121
PC.2. Requisitos software.....	122
PRESUPUESTO	123
P.1. Declaración jurada.....	123
P.2. Desglose del presupuesto	124
P.3. Recursos materiales.....	124
P.4. Recursos software	125
P.5. Recursos hardware	126
P.6. Trabajo tarifado por tiempo empleado	127
P.7. Costes de redacción	129

P.8. Material fungible	130
P.9. Derechos de visado del COIT	130
P.10. Gastos de tramitación y envío	131
P.11. Aplicación de impuestos.....	131

Índice de Figuras

Figura 1.1. Sensor Leap Motion	4
Figura 1.2. Ejes de coordenadas respecto a la superficie del sensor	4
Figura 1.3. Rango de interacción con el sensor.....	5
Figura 1.4. Interior de Leap Motion, donde se observan las cámaras de infrarrojo.....	5
Figura 1.5. Rango de interacción de Kinect	7
Figura 2.1. Distribución de edad de los usuarios de la base de datos de 39 usuarios	11
Figura 2.2. Ejecución del código de ejemplo en el entorno NetBeans 8.0.1	12
Figura 2.3. Extracto de la información capturada para una ejecución del archivo de ejemplo HTML.....	13
Figura 2.4. Archivo de texto creado para la captura de la firma de un usuario	14
Figura 2.5. Etapa 1 del protocolo de realización del trazo de la firma.....	15
Figura 2.6. Etapa 2 del protocolo de realización del trazo de la firma.....	15
Figura 2.7. Etapa 3 del protocolo de realización del trazo de la firma.....	16
Figura 2.8. Etapa 4 del protocolo de realización del trazo de la firma.....	17
Figura 2.9. Visualizador del SDK de Leap Motion para observar la mano y el trazo...	17
Figura 2.10. Código para la captura y escritura de los datos en el archivo de texto mayo_30.txt.....	18
Figura 2.11. Definición del esqueleto de la mano y los métodos tipPosition y tipVelocity.....	19
Figura 2.12. Definición del método direction	19
Figura 2.13. Definición de los métodos palmPosition y palmVelocity	20
Figura 2.14. Definición de los métodos palmNormal, direction.pitch, direction.yaw y direction.roll	20
Figura 2.15. Definición de los ángulos pitch, yaw y roll	21
Figura 2.16. Archivo de texto formateado a partir del archivo de texto creado para la captura de la firma	23
Figura 2.17. Representación de la matriz de costes del algoritmo Dynamic Time Warping para las secuencias A y B	24
Figura 2.18. Trazo tridimensional caracterizado por las coordenadas (x,y,z) de la palabra "trazo"	25

Figura 2.19. Proyecciones del trazo tridimensional de la figura 23 sobre los planos (x,y) , (x,z) y (y,z)	25
Figura 2.20. Distribución de edad de los usuarios de la base de datos de 70 usuarios	26
Figura 2.21. Distribución de edad de los usuarios de la base de datos de 100 usuarios	27
Figura 3.1. Medias de cada uno de los segmentos de las señales (x,y,z) del dedo índice	31
Figura 3.2. Ejemplos de curtosis.....	32
Figura 3.3. Ejemplos de asimetría (skewness)	32
Figura 3.1. Posibles hiperplanos de separación	34
Figura 3.2. Hiperplano de separación óptima.....	34
Figura 3.3. Hiperplano de separación óptima con vector ortogonal al hiperplano	36
Figura 3.4. Transformación de un caso no linealmente separable a un caso linealmente separable	37
Figura 4.1. Representación gráfica de las medidas de FRR, FAR y EER	42
Figura 5.1. <i>Kernel</i> polinómico para todas las señales XY con 5 muestras	53
Figura 5.2. <i>Kernel</i> polinómico para todas las señales XZ con 5 muestras	53
Figura 5.3. <i>Kernel</i> polinómico para todas las señales YZ con 5 muestras	54
Figura 5.4. <i>Kernel</i> polinómico todas las señales XYZ con 5 muestras	54
Figura 5.5. <i>Kernel</i> polinómico para todas las señales XY con 1 muestra.....	55
Figura 5.6. <i>Kernel</i> polinómico para todas las señales XZ con 1 muestra.....	56
Figura 5.7. <i>Kernel</i> polinómico para todas las señales YZ con 1 muestra.....	56
Figura 5.8. <i>Kernel</i> polinómico para todas las señales XYZ con 1 muestra.....	57
Figura 5.9. <i>Kernel</i> RBF para todas las señales XY con 5 muestras.....	58
Figura 5.10. <i>Kernel</i> RBF para todas las señales XYZ con 5 muestras.....	59
Figura 5.11. <i>Kernel</i> RBF para todas las señales XY con 1 muestra	60
Figura 5.12. <i>Kernel</i> RBF para todas las señales XYZ con 1 muestra	60
Figura 5.13. Variación del EER al añadir nuevos parámetros estadísticos, usando 5 muestras.	63
Figura 5.14. Variación del EER al añadir nuevos parámetros estadísticos usando 1 muestra.....	64

Figura 5.15. Comparación de los planos XY y XYZ, usando <i>kernel</i> RBF y 5 muestras de entrenamiento.	65
Figura 5.16. Comparación de los planos XY y XYZ, usando <i>kernel</i> RBF y 4 muestras de entrenamiento.	66
Figura 5.17. Comparación de los planos XY y XYZ, usando <i>kernel</i> RBF y 3 muestras de entrenamiento.	66
Figura 5.18. Comparación de los planos XY y XYZ, usando <i>kernel</i> RBF y 2 muestras de entrenamiento.	67
Figura 5.19. Comparación de los planos XY y XYZ, usando <i>kernel</i> RBF y 1 muestra de entrenamiento.	67
Figura 5.20. Comparación de los planos XY y XYZ, usando <i>kernel</i> polinómico y 5 muestras de entrenamiento.	68
Figura 5.21. Comparación de los planos XY y XYZ, usando <i>kernel</i> polinómico y 4 muestras de entrenamiento.	68
Figura 5.22. Comparación de los planos XY y XYZ, usando <i>kernel</i> polinómico y 3 muestras de entrenamiento.	69
Figura 5.23. Comparación de los planos XY y XYZ, usando <i>kernel</i> polinómico y 2 muestras de entrenamiento.	69
Figura 5.24. Comparación de los planos XY y XYZ, usando <i>kernel</i> polinómico y 1 muestra de entrenamiento.....	70
Figura 5.25. Conjunto de boxplots de resultados para 70 usuarios, usando <i>kernel</i> polinómico	73
Figura 5.27. Conjunto de boxplots de resultados para 100 usuarios, usando <i>kernel</i> polinómico	75
Figura 5.28. Conjunto de boxplots de resultados para 100 usuarios, usando <i>Kernel</i> RBF.....	75
Figura B.1. Explicación del boxplot	99
Figura B.2. Boxplot de resultados para 70 usuarios con 1 muestra entrenamiento usando <i>kernel</i> polinómico	99
Figura B.3. Boxplot de resultados para 70 usuarios con 2 muestras entrenamiento usando <i>kernel</i> polinómico	100
Figura B.4. Boxplot de resultados para 70 usuarios con 3 muestras entrenamiento usando <i>kernel</i> polinómico	100
Figura B.5. Boxplot de resultados para 70 usuarios con 4 muestras entrenamiento usando <i>kernel</i> polinómico	101

Figura B.6. Boxplot de resultados para 70 usuarios con 5 muestras entrenamiento usando <i>kernel</i> polinómico	101
Figura B.7. Boxplot de resultados para 70 usuarios con 1 muestra entrenamiento usando <i>kernel</i> RBF	102
Figura B.8. Boxplot de resultados para 70 usuarios con 2 muestras entrenamiento usando <i>kernel</i> RBF	102
Figura B.9. Boxplot de resultados para 70 usuarios con 2 muestras entrenamiento usando <i>kernel</i> RBF	103
Figura B.10. Boxplot de resultados para 70 usuarios con 4 muestras entrenamiento usando <i>kernel</i> RBF	103
Figura B.11. Boxplot de resultados para 70 usuarios con 5 muestras entrenamiento usando <i>kernel</i> RBF	104
Figura B.12. Boxplot de resultados para 100 usuarios con 1 muestra entrenamiento usando <i>kernel</i> polinómico	104
Figura B.13. Boxplot de resultados para 100 usuarios con 2 muestras entrenamiento usando <i>kernel</i> polinómico	105
Figura B.14. Boxplot de resultados para 100 usuarios con 3 muestras entrenamiento usando <i>kernel</i> polinómico	105
Figura B.15. Boxplot de resultados para 100 usuarios con 4 muestras entrenamiento usando <i>kernel</i> polinómico	106
Figura B.16. Boxplot de resultados para 100 usuarios con 5 muestras entrenamiento usando <i>kernel</i> polinómico	106
Figura B.17. Boxplot de resultados para 100 usuarios con 1 muestra entrenamiento usando <i>kernel</i> RBF	107
Figura B.18. Boxplot de resultados para 100 usuarios con 2 muestras entrenamiento usando <i>kernel</i> RBF	107
Figura B.19. Boxplot de resultados para 100 usuarios con 3 muestras entrenamiento usando <i>kernel</i> RBF	108
Figura B.20. Boxplot de resultados para 100 usuarios con muestras entrenamiento usando <i>kernel</i> RBF	108
Figura B.21. Boxplot de resultados para 100 usuarios con 5 muestras entrenamiento usando <i>kernel</i> RBF	109

Índice de Tablas

Tabla 1.1. Unidades de las magnitudes medidas	4
Tabla 2.1. Distribución por sexo de los usuarios de la base de datos de 39 usuarios. 11	
Tabla 2.2. Listado de los datos capturados junto a los métodos que los devuelven.. 22	
Tabla 2.3. Distribución por sexo de los usuarios de la base de datos de 70 usuarios. 26	
Tabla 2.4. Distribución por sexo de los usuarios de la base de datos de 100 usuarios	27
Tabla 4.1. Breve descripción de cada experimento	43
Tabla 5.1. Mejores resultados con el clasificador no ajustado, con 5 muestras usando <i>kernel RBF</i>	50
Tabla 5.1. Resultados de la simulación hecha con la BBDD de 70 usuarios	72
Figura 5.26. Conjunto de boxplots de resultados para 70 usuarios, usando <i>kernel RBF</i>	73
Tabla 5.2. Resultados de la simulación hecha con la BBDD de 100 usuarios.....	74
Tabla 5.3. Comparación de resultados para <i>kernel RBF</i>	78
Tabla 5.4. Comparación de resultados para <i>kernel RBF</i>	80
Tabla 6.1. Comparativa de rendimiento entre diferentes trabajos	83
Tabla A.1. Resultados del conjunto de todas señales del plano XY para 5 muestras entrenamiento con <i>kernel RBF</i>	85
Tabla A.2. Resultados del conjunto de todas señales del plano XYZ para 5 muestras entrenamiento con <i>kernel RBF</i>	86
Tabla A.3. Resultados del conjunto de todas señales del plano XY para 1 muestra entrenamiento con <i>kernel RBF</i>	86
Tabla A.4. Resultados del conjunto de todas señales del plano XYZ con 1 muestra entrenamiento para <i>kernel RBF</i>	87
Tabla A.5. Resultados del conjunto de todas señales del plano XY con 5 muestras entrenamiento para <i>kernel</i> polinómico	87
Tabla A.6. Resultados del conjunto de todas señales del plano XYZ con 5 muestras entrenamiento para <i>kernel</i> polinómico	88
Tabla A.7. Resultados del conjunto de todas señales del plano XY con 1 muestra entrenamiento para <i>kernel</i> polinómico	88
Tabla A.8. Resultados del conjunto de todas señales del plano XYZ con 1 muestra entrenamiento para <i>kernel</i> polinómico	89

Tabla A.9. Variación del EER al añadir nuevos parámetros estadísticos para 5 muestras.....	90
Tabla A.10. Variación del EER al añadir nuevos parámetros estadísticos para 1 muestra.....	90
Tabla A.11. Resultados del conjunto del plano XY para 4 muestras entrenamiento usando <i>kernel</i> polinómico	91
Tabla A.12. Resultados del conjunto del plano XYZ para 4 muestras entrenamiento usando <i>kernel</i> polinómico	91
Tabla A.13. Resultados del conjunto del plano XY para 3 muestras entrenamiento usando <i>kernel</i> polinómico	92
Tabla A.14. Resultados del conjunto del plano XYZ para 3 muestras entrenamiento usando <i>kernel</i> polinómico	92
Tabla A.15. Resultados del conjunto del plano XY para 2 muestras entrenamiento usando <i>kernel</i> polinómico	93
Tabla A.16. Resultados del conjunto del plano XYZ para 2 muestras entrenamiento usando <i>kernel</i> polinómico	93
Tabla A.17. Resultados del conjunto del plano XY para 4 muestras entrenamiento usando <i>kernel</i> RBF.....	94
Tabla A.18. Resultados del conjunto del plano XYZ para 4 muestras entrenamiento usando <i>kernel</i> RBF.....	94
Tabla A.19. Resultados del conjunto del plano XY para 3 muestras entrenamiento usando <i>kernel</i> RBF.....	95
Tabla A.20. Resultados del conjunto del plano XYZ para 3 muestras entrenamiento usando <i>kernel</i> RBF.....	95
Tabla A.21. Resultados del conjunto del plano XY para 2 muestras entrenamiento usando <i>kernel</i> RBF.....	96
Tabla A.22. Resultados del conjunto del plano XYZ para 2 muestras entrenamiento con <i>kernel</i> RBF.....	96
Tabla A.21. Resultados del experimento 6, para 70 usuarios	97
Tabla A.22. Resultados del experimento 7, para 100 usuarios.....	98
Tabla PR.1. Costes de los recursos software	125
Tabla PR.2. Costes de los recursos hardware	126
Tabla PR.3. Factor de corrección del COIT según las horas trabajadas.....	128
Tabla PR.4. Presupuesto: Software + Hardware + Tiempo empleado.	129

Tabla PR.5. Costes del material fungible..... 130

Tabla PR.6. Tabla final de costes con la aplicación de impuestos..... 131

Acrónimos

DTW:	Dynamic Time Warping (<i>Alineamiento Temporal Dinámico</i>).
EER:	Equal Error Rate (<i>Tasa de Error Igual</i>).
FAR:	False Acceptance Rate (<i>Tasa de falso Positivo</i>).
FRR:	False Rejection Rate (<i>Tasa de Falso Negativo</i>).
LS-SVM:	Least Squares Support Vector Machine (<i>Máquina de vectores de soporte de Mínimos Cuadrados</i>).
RBF:	Radial Basis Function (<i>Función de Base Radial</i>).
SVM:	Support Vector Machine (<i>Máquina de vectores de soporte</i>).
IDeTIC:	Instituto para el Desarrollo Tecnológico y la Innovación en Comunicaciones.

Capítulo 1. Introducción

Desde su inicio, el sector de la informática y de las telecomunicaciones ha evolucionado considerablemente. Ha habido un gran desarrollo en las tecnologías de la información y, debido a esto, nuestra sociedad está cada vez más conectada electrónicamente. Cada día hay más sistemas automatizados que realizan muchas de las tareas que antes realizaban las personas, como la administración y la gestión de la información. Como consecuencia de todo esto, la tarea de determinar la identidad de un individuo se ha convertido en una tarea muy importante [1].

Por esto, en las últimas décadas, se ha desarrollado lo que se conoce como *Biometría*.

1.1. Definición de biometría

El término biometría deriva de las palabras *bios* (vida) y *metron* (medida), y se define como estudio mensurativo o estadístico de fenómenos o procesos biológicos [2]. En particular, para este proyecto se puede definir la biometría como el método de reconocimiento de individuos basado en sus rasgos fisiológicos o de comportamiento [3].

Dependiendo del tipo de característica que se use para la identificación, existen dos grandes familias de técnicas de reconocimiento de personas [4]:

- Técnicas basadas en **rasgos físicos**: están basadas en una característica física de la persona. Por ejemplo, la huella dactilar, el iris, la mano, la cara, etc.
- Técnicas basadas en **rasgos de comportamiento**: se basan en cómo el usuario realiza una acción. Como por ejemplo la firma, la forma de caminar, etc.

Las técnicas basadas en rasgos físicos presentan un mayor nivel de seguridad debido a la variabilidad de los rasgos de comportamiento frente a las basadas en rasgos de comportamiento. Por ejemplo, una persona nunca va a firmar dos veces de la misma manera, sin embargo, el iris de su ojo o su huella dactilar siempre serán iguales. Así

Capítulo 1. Introducción

mismo, existen técnicas híbridas que combinan ambos tipos de rasgos, como es el reconocimiento por voz [4].

No existe un sistema biométrico idóneo. Se ha de seleccionar un sistema biométrico en base a las características del escenario en el que se vaya a utilizar. Por ejemplo, no es lógico pensar en el acceso a un gimnasio mediante técnicas de reconocimiento de iris, pero sí mediante una tarjeta identificativa a pesar de los accesos fraudulentos que esto pueda conllevar. Los diferentes sistemas muestran diferentes características, no existe un sistema biométrico que sobresalga sobre el resto a nivel de prestaciones. Por tanto, es importante conocer los diferentes entornos de aplicación y sus características fundamentales a la hora de seleccionar el sistema que mejor se adapte al escenario en cuestión [2] [5].

La firma manuscrita es muy utilizada y aceptada dentro de la línea del estudio de la biometría mediante técnicas de comportamiento, a pesar de no ser la técnica que mejores resultados ofrece. Por este motivo, y con el objetivo de aumentar la seguridad de esta técnica, en este proyecto se realiza el estudio de la biometría mediante el **trazo aéreo de la firma**.

1.2. Firma biométrica

La firma biométrica es una tecnología que permite capturar la firma manuscrita utilizando dispositivos especiales para ello y autenticar la identidad del firmante. Es decir, el sistema recoge las características físicas de la firma original y luego las compara con las subsiguientes firmas para proceder a su validación o repudio [6].

El reconocimiento de una persona mediante una firma es un proceso que consiste en, dada una firma realizada por un usuario, decidir si es una firma perteneciente a dicho usuario (firma genuina) o ha sido realizada por otro usuario (firma falsificada).

Cuando se realizan tareas de verificación de firmas, existen dos vertientes para realizar el análisis de las mismas. Por un lado técnicas llamadas *off-line*, y por otro las conocidas como técnicas *on-line* [7, 1] [8]. En ellas se intenta identificar y analizar los patrones que

caractericen la firma de cada usuario, intentando diferir entre las firmas de diferentes usuarios y/o si una firma es original o realizada por un impostor.

- La verificación **off-line** se refiere a las técnicas en las que se parte de la imagen escaneada de la firma manuscrita.
- En cuanto a las técnicas **on-line**, se utilizan para medir las secuencias de datos, como la velocidad [9], la presión [10], etc.

En general, las técnicas *on-line* ofrecen mejor resultado debido a las fuentes de información de las que dispone.

Por esta razón, en este proyecto se realizará un estudio de las firmas mediante **técnicas on-line**, donde el dispositivo que será usado para la recolección de las firmas será el *Leap Motion*.

1.3. Leap Motion

Los trazos de la firma de cada usuario son capturados mediante un sensor comercial, *Leap Motion* (figura 1.1), que realiza un barrido en infrarrojo y ofrece la posibilidad de utilizar diferentes datos, como la posición de los dedos, la posición de la palma de la mano, la velocidad de movimiento de ambos, etc., para cada *frame* capturado [11]. Las magnitudes medidas por el sensor se ofrecen de acuerdo a las unidades mostradas en la tabla 1.1.

En la figura 1.2 se pueden observar los ejes de coordenadas definidos sobre la superficie del sensor.



Figura 1.1. Sensor Leap Motion

Distancia	Milímetros
Tiempo	Microsegundos (modificable)
Velocidad	Milímetros / segundo
Ángulos	Radianes

Tabla 1.1. Unidades de las magnitudes medidas

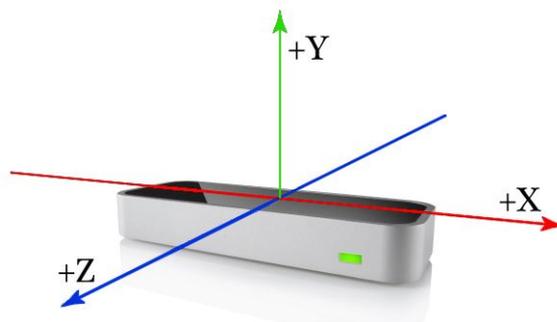


Figura 1.2. Ejes de coordenadas respecto a la superficie del sensor

El rango de interacción del dispositivo se puede observar en la figura 1.3, de acuerdo al fabricante. *Leap Motion* no necesita que el usuario se tenga que alejar para poder interactuar con él, a diferencia de otros sensores que utilizan el infrarrojo para detectar la profundidad de los objetos como *Kinect*.

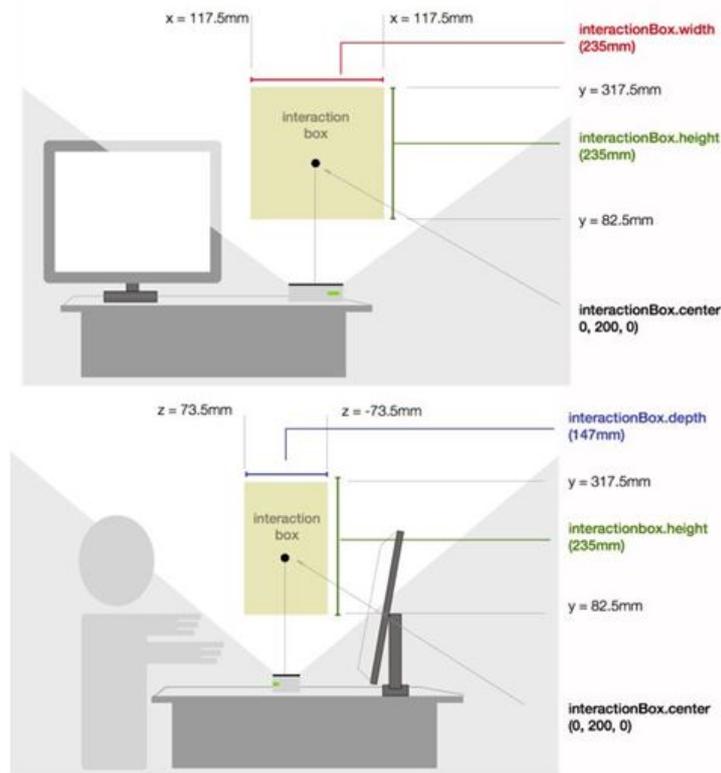


Figura 1.3. Rango de interacción con el sensor

Otra ventaja de este sensor, al utilizar el espectro infrarrojo, es la posibilidad de trabajar en condiciones de poca luminosidad (figura 1.4). Este hecho es una de las fortalezas que también poseen otros sensores como *Kinect* frente a diferentes aplicaciones de detección y seguimiento de los trazos que utilizan cámaras en el espectro visible [12] [13] [14].



Figura 1.4. Interior de Leap Motion, donde se observan las cámaras de infrarrojo

1.4. Antecedentes

Uno de los antecedentes más recientes es el Trabajo de Fin de Máster sobre el que se basa este Proyecto de Fin de Carrera. Se trata del Trabajo de Fin de Máster denominado **“Estudio de la biometría del trazo aéreo de la firma para la identificación personal”** [5]. En éste, a partir de las diferentes señales que se obtienen con *Leap Motion*, se prueba una serie de clasificadores (SVM y DTW) y se busca el que mejores resultados ofrece en la evaluación de la identidad de un usuario a partir de su firma realizada en el aire.

Por otro lado, recientemente se han desarrollado sistemas de análisis *on-line* que analizan la firma en más de dos dimensiones [3] [15]. Algunos de estos estudios caracterizan la presión que imprime el usuario en el trazo de la firma sobre algún tipo de superficie como una tercera dimensión de estudio, otros utilizan información extraída de bolígrafos u otros dispositivos con sensores al ser usados para realizar la firma [16] [17].

Como otra fuente de adquisición de datos, hay trabajos que muestran aplicaciones basadas en acelerómetros integrados para identificar al usuario mediante el gesto que realiza con el dispositivo [4] [18] [19] [20] [21]. A diferencia de estos trabajos, en esta propuesta no hay que sostener el sensor en la mano, sino que se realiza una captura del trazo descrito por el dedo en el aire, al igual que si de una cámara se tratara.

Como alternativa a los acelerómetros, existen trabajos que obtienen la trayectoria de la firma realizada mediante el procesado de las imágenes capturadas por una o varias cámaras, así como por diferentes sensores [12] [13] [14] [22] [23].

1.5. Novedad de la propuesta

Por un lado, a diferencia de los antecedentes que utilizan cámaras de espectro visible, *Leap Motion* ofrece la ventaja de no necesitar buenas condiciones lumínicas para su funcionamiento debido al uso que hace del espectro infrarrojo. Este espectro infrarrojo es también un punto fuerte en el uso del sensor *Kinect*. Sin embargo, este último posee

la debilidad, en el caso de los escenarios pensados para este trabajo, de fijar un rango de actuación alejado del dispositivo con el que se interactúa (figura 1.5).

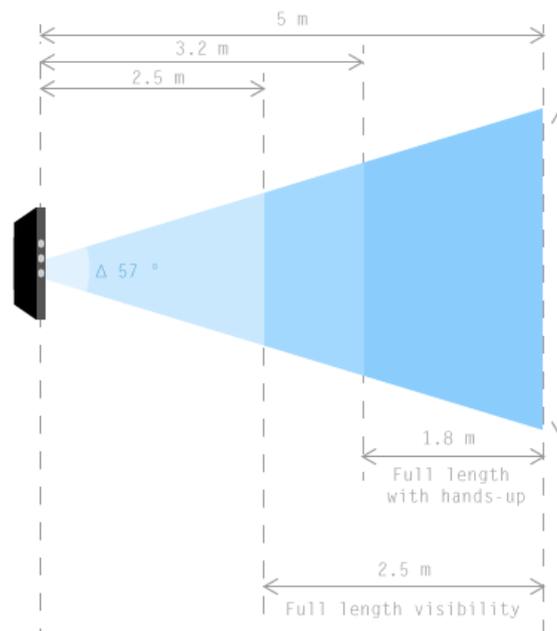


Figura 1.5. Rango de interacción de Kinect

En cuanto al proyecto *AirAuth*, mostrado en [24], el cual es el que más semejanzas presenta con este trabajo en cuanto a las características del sensor, las diferencias son varias. En primer lugar este trabajo utiliza un sensor de bajo coste dado que el precio del sensor es 2.2 veces inferior al usado en *AirAuth*, así como mayor variedad de sistemas operativos donde se pueda utilizar. Por otro lado, la necesidad de adaptarse a la localización del sensor (figura 1.6), tipo *webcam*, limita los escenarios de utilización dado que para su uso el usuario ha de levantarse o levantar el brazo a la altura del monitor del dispositivo con el que interactúe, mientras que *Leap Motion* presenta la ventaja de colocarse sobre la mesa, pudiendo ser utilizado sin necesidad de que el usuario se levante. Como última diferencia, *AirAuth* no especifica que se realice la escritura de la firma del usuario con un dedo, sino que captura el gesto realizado.

En resumen, la novedad de la propuesta que se pretende llevar a cabo con este trabajo se fundamenta en las características del sensor utilizado, así como la manera de caracterizar el trazo aéreo de la firma.

Capítulo 1. Introducción

Como se puede observar en los antecedentes mostrados, en ninguno de ellos se realiza la caracterización del trazo de la firma tal como aquí se propone, mediante la escritura aérea con el dedo de escritura.

Por otro lado, también hay novedades con respecto al Trabajo de Fin de Máster antes nombrado, sobre el cual se basa este proyecto. Se trabajará esta vez con un nuevo clasificador. En vez de usar los clasificadores DTW (Dynamic Time Warping) y el SVM (Support Vector Machine, Máquina de Vectores de Soporte), se trabajará con un LS-SVM (Least Square Support Vector Machine, Máquinas de Vectores de Soporte de Mínimos Cuadrados). Además se incluirán nuevos parámetros para intentar aportar una mayor información y así intentar realizar una mejor identificación de las firmas. También, el estudio se realizaba dividiendo los conjuntos en 5, 10 y 15 segmentos, sin embargo, ahora se hará dividiendo los conjuntos de 2 a 30 segmentos, obteniendo mucha más información. Por último, se ampliará la base de datos proporcionada de 39 usuarios llegando a un total de 100 firmantes.

1.6. Objetivos

Como bien se ha mencionado anteriormente, este proyecto trata sobre el estudio de firmas realizadas a partir de trazos aéreos proporcionadas a partir de una base de datos. Dicha base de datos ha sido proporcionada por el IDeTIC.

El **objetivo principal** de este proyecto es determinar si la tercera dimensión aporta información al estudio. Cuando una persona realiza su firma, las coordenadas que se “controlan” son las coordenadas X e Y, es decir, todas las personas realizan la firma sobre el plano XY (lo que vendría siendo el papel) y, sin embargo, la intensidad con la que se aprieta el bolígrafo es algo que se realiza de manera inconsciente.

Otro **objetivo importante** será ampliar la base de datos proporcionada de 39 usuarios a un total de 100 usuarios. Mientras más usuarios tenga la base de datos, mejores conclusiones se podrán obtener.

Como **objetivos secundarios**, también se quiere estudiar cómo se comporta el sistema cuando se aumenta o disminuye el número de muestras de entrenamiento. Además, se

evaluarán parámetros estadísticos diferentes a los que ya se usan en la parametrización de las firmas. Por otro lado, como último objetivo secundario, se analizará si los resultados pueden mejorar realizando fusiones de los distintos subconjuntos de señales.

1.7. Organización de la memoria

La memoria de este Proyecto de Fin de Carrera está compuesta por 6 capítulos y 2 anexos, así como los apartados de Planos y Programas, Pliego de Condiciones, Presupuesto y Bibliografía. A continuación se describe brevemente el contenido de ellos:

- **Capítulo 1.** Introducción.
Introducción a la biometría y al sensor utilizado. Se muestran los antecedentes a este trabajo, así como la novedad de la propuesta. Por último se listan los objetivos planteados para el desarrollo del trabajo.
- **Capítulo 2.** Base de Datos.
En este capítulo se especifican las características de la base de datos y se describe el proceso que se sigue para ampliarla.
- **Capítulo 3.** Clasificación.
Se explicará cómo se agrupan y dividen las señales obtenidas de cada firma, así como el clasificador que se usará en este proyecto. También se indicará qué parámetros se calculan para cada uno de los grupos de señales para su posterior evaluación y clasificación.
- **Capítulo 4.** Metodología.
Se mostrará la metodología seguida para el alcance de los objetivos planteados.
En este capítulo, además se describirán los distintos experimentos que se llevarán a cabo para el logro de estos objetivos.

Capítulo 1. Introducción

- **Capítulo 5. Resultados.**
En este capítulo se describe los resultados obtenidos para cada uno de los experimentos diseñados.
- **Capítulo 6. Conclusiones y líneas futuras.**
Se resumirá las conclusiones que se han obtenido en el desarrollo de este proyecto. Además se plantearán líneas futuras de trabajo.
- **Anexo A. Tablas de resultados**
Presentación en tablas de los resultados obtenidos en el capítulo 5 de resultados. En estas tablas se muestran los resultados de una manera más detallada.
- **Anexo B. Boxplots**
Presentación de los resultados obtenidos en diagramas de cajas.
- **Bibliografía.**
- **Planos y programas**
Se presentan los principales scripts desarrollados para la realización de este proyecto.
- **Pliego de condiciones**
Se presentan las herramientas hardware y software que han sido necesarias en el desarrollo del proyecto.
- **Presupuesto**
Cálculo del coste que ha supuesto el desarrollo de este proyecto.

Capítulo 2. Base de datos

En esta sección se detallan las características de la base de datos proporcionada y el proceso a seguir para la captura de los trazos y la ampliación de la misma, así como los datos a capturar para cada firma y las características de los usuarios que la componen.

2.1. Descripción de la base de datos proporcionada

La base de datos proporcionada consta de un total de 39 usuarios y 10 muestras para cada uno de ellos, lo que hace un total de 390 muestras. Los usuarios que forman la base de datos son de diferente sexo y edad, y quedan distribuidos como se detalla a continuación.

En la tabla 2.1 se muestra la distribución de los usuarios por sexo.

Sexo	Distribución
Masculino	64,10%
Femenino	35,90%

Tabla 2.1. Distribución por sexo de los usuarios de la base de datos de 39 usuarios

En la figura 2.1 se puede observar la distribución de los mismos por edad. Se observa que la media es de unos casi 26 años, con una desviación típica de poco menos de ± 7 años.

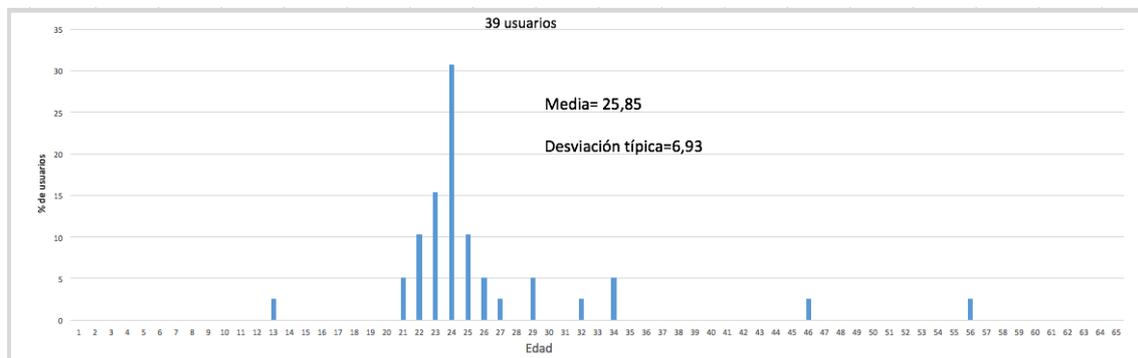


Figura 2.1. Distribución de edad de los usuarios de la base de datos de 39 usuarios

Capítulo 2. Base de datos

Se observa que el rango de edad donde se concentra la mayoría de los firmantes está entre 22 a 25 años.

2.2. Ampliación de la base de datos

Uno de los objetivos es ampliar la base de datos proporcionada. Para ello, como bien se ha explicado, se usa el sensor de infrarrojos *Leap Motion*, el cual será controlado a través de un código que se modificará a través del software *NetBeans 8*.

2.2.1. Captura de datos

Para la captura de los datos utilizando el sensor *Leap Motion*, se utiliza como fuente el código de ejemplo disponible para lenguaje *Java*. del *Kit de Desarrollo Software (SDK)* disponible en la web de desarrolladores de *Leap Motion* [11].

Con este código es posible observar los datos capturados para la mano (o para ambas manos) que se coloque sobre el sensor. En la figura 2.2 se puede observar el código de ejemplo en funcionamiento sobre el entorno de programación *NetBeans*.

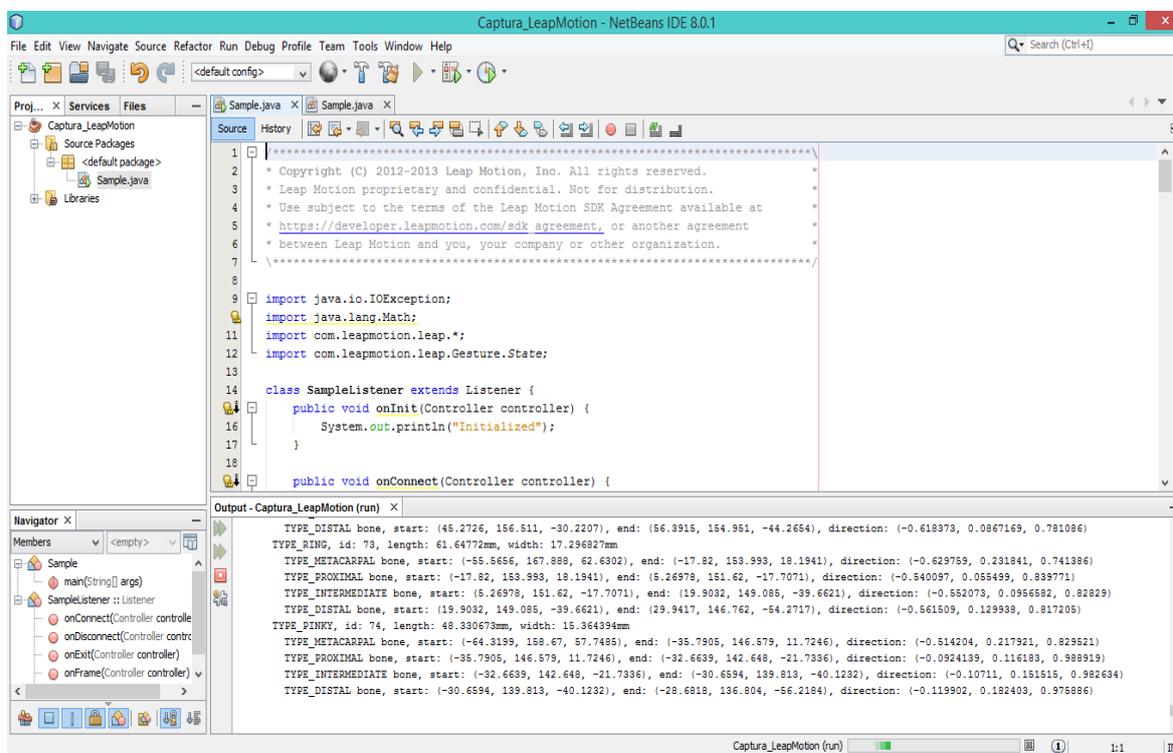


Figura 2.2. Ejecución del código de ejemplo en el entorno NetBeans 8.0.1

Para una visión más clara de la información capturada, se muestra en la figura 2.3 la ejecución del mismo código de ejemplo en lenguaje *HTML*.

Finger and tool data:

Pointable ID: 100	Pointable ID: 101	Pointable ID: 102	Pointable ID: 103	Pointable ID: 104
Type: Thumb	Type: Index finger	Type: Middle finger	Type: Ring finger	Type: Pinky finger
Belongs to hand with ID: 10	Belongs to hand with ID: 10	Belongs to hand with ID: 10	Belongs to hand with ID: 10	Belongs to hand with ID: 10
Classified as a finger	Classified as a finger	Classified as a finger	Classified as a finger	Classified as a finger
Length: 52.1 mm	Length: 58.8 mm	Length: 67.0 mm	Length: 64.4 mm	Length: 50.5 mm
Width: 20.3 mm	Width: 19.3 mm	Width: 19.0 mm	Width: 18.1 mm	Width: 16.1 mm
Direction: (0.99, 0.08, -0.08)	Direction: (0.65, -0.04, -0.76)	Direction: (0.56, -0.04, -0.83)	Direction: (0.51, -0.03, -0.86)	Direction: (0.02, -0.10, -1.00)
Extended?: true	Extended?: true	Extended?: true	Extended?: true	Extended?: true
Metacarpal bone	Metacarpal bone	Metacarpal bone	Metacarpal bone	Metacarpal bone
Center: (-37.2, 139.0, 95.9)	Center: (-6.7, 152.7, 60.5)	Center: (-17.5, 152.2, 47.3)	Center: (-31.2, 149.3, 37.2)	Center: (-45.2, 141.2, 30.4)
Direction: (0.9, 0.1, -0.3)	Direction: (0.8, -0.1, -0.6)	Direction: (0.7, -0.1, -0.7)	Direction: (0.6, -0.2, -0.8)	Direction: (0.5, -0.1, -0.8)
Up vector: (0.2, 0.6, 0.8)	Up vector: (0.0, 1.0, -0.2)	Up vector: (-0.1, 0.9, -0.3)	Up vector: (-0.2, 0.9, -0.3)	Up vector: (-0.3, 0.9, -0.3)
Proximal phalanx bone	Proximal phalanx bone	Proximal phalanx bone	Proximal phalanx bone	Proximal phalanx bone
Center: (-13.2, 137.7, 89.0)	Center: (37.5, 148.4, 23.8)	Center: (21.5, 147.3, 4.2)	Center: (-0.2, 144.3, -5.9)	Center: (-30.1, 136.0, -11.7)
Direction: (1.0, -0.1, -0.3)	Direction: (0.6, 0.0, -0.8)	Direction: (0.5, 0.0, -0.8)	Direction: (0.5, 0.0, -0.9)	Direction: (0.0, -0.1, -1.0)
Up vector: (0.3, 0.6, 0.8)	Up vector: (-0.1, 1.0, -0.1)	Up vector: (-0.2, 1.0, -0.1)	Up vector: (-0.3, 0.9, -0.2)	Up vector: (-0.3, 0.9, -0.1)
Intermediate phalanx bone	Intermediate phalanx bone	Intermediate phalanx bone	Intermediate phalanx bone	Intermediate phalanx bone
Center: (27.7, 137.7, 80.7)	Center: (59.1, 148.2, -1.9)	Center: (42.6, 146.8, -27.8)	Center: (18.2, 144.0, -37.0)	Center: (-29.6, 133.9, -39.1)
Direction: (1.0, 0.1, -0.1)	Direction: (0.6, -0.0, -0.8)	Direction: (0.6, -0.0, -0.8)	Direction: (0.5, -0.0, -0.9)	Direction: (0.0, -0.1, -1.0)
Up vector: (0.0, 0.6, 0.8)	Up vector: (-0.0, 1.0, -0.1)	Up vector: (-0.2, 1.0, -0.2)	Up vector: (-0.3, 0.9, -0.2)	Up vector: (-0.3, 0.9, -0.1)
Distal phalanx bone	Distal phalanx bone	Distal phalanx bone	Distal phalanx bone	Distal phalanx bone
Center: (54.9, 143.1, 83.4)	Center: (72.5, 146.8, -17.6)	Center: (55.9, 145.6, -47.3)	Center: (30.2, 143.2, -56.8)	Center: (-29.1, 131.8, -57.4)
Direction: (0.9, 0.3, 0.3)	Direction: (0.6, -0.1, -0.8)	Direction: (0.6, -0.1, -0.8)	Direction: (0.5, -0.1, -0.9)	Direction: (0.0, -0.1, -1.0)
Up vector: (-0.5, 0.4, 0.8)	Up vector: (-0.0, 1.0, -0.1)	Up vector: (-0.2, 1.0, -0.2)	Up vector: (-0.2, 0.9, -0.2)	Up vector: (-0.3, 0.9, -0.1)
Tip position: (60.4, 144.4, 85.2) mm	Tip position: (75.7, 146.1, -20.8) mm	Tip position: (59.0, 144.8, -51.2) mm	Tip position: (33.2, 142.5, -60.9) mm	Tip position: (-28.5, 131.0, -61.9) mm

Figura 2.3. Extracto de la información capturada para una ejecución del archivo de ejemplo HTML

Este código de ejemplo se modifica para que capture los datos de acuerdo a un protocolo de captura diseñado con el fin de identificar de forma automática el inicio y el fin de la firma en función de los gestos realizados por el usuario. Los datos capturados se almacenan en un archivo de texto con el formato que se puede ver en la figura 2.4 [5].

Capítulo 2. Base de datos

```
(-36.5497, 222.289, -75.1226) (4.49074, 5.04888, 2.09448) (-0.075988, 0.0596664, -0.995322) (-7.10609, 201.933, 11.5613) (-0.664622, -6.02258, 1.72171) (0.0516093, -0.
(-36.4801, 222.26, -75.0689) (4.4006, 3.1159, 2.93865) (-0.0763852, 0.0603872, -0.995248) (-7.10276, 201.779, 11.6113) (-0.319339, -8.11142, 2.49206) (0.0459172, -0.89
(-36.4382, 222.221, -75.0217) (3.69239, 1.32923, 3.2515) (-0.0767574, 0.0607283, -0.995199) (-7.11337, 201.641, 11.6543) (-0.501717, -9.01394, 2.76211) (0.0408854, -0.
(-36.4276, 222.174, -74.9695) (2.42744, -0.284967, 3.57557) (-0.0773587, 0.0607587, -0.99515) (-7.13317, 201.496, 11.6935) (-0.881596, -9.80321, 2.87306) (0.0357292, -0.
(-36.4205, 222.128, -74.9235) (1.52662, -1.23212, 3.58097) (-0.0779252, 0.0608823, -0.995098) (-7.15156, 201.364, 11.7276) (-1.07365, -9.87949, 2.76477) (0.0310742, -0.
(-36.4068, 222.069, -74.8787) (1.15578, -2.13091, 3.56161) (-0.0782931, 0.0609238, -0.995067) (-7.17501, 201.244, 11.7698) (-1.34138, -9.61584, 2.93527) (0.0268576, -0.
(-36.42, 221.987, -74.8284) (0.140736, -3.32537, 3.69105) (-0.0790002, 0.0609728, -0.995008) (-7.21945, 201.069, 11.8067) (-2.10145, -10.9716, 2.88197) (0.020915, -0.9
(-36.4403, 221.836, -74.7682) (-0.72821, -6.16462, 4.03586) (-0.0796314, 0.0604172, -0.994992) (-7.28219, 200.871, 11.8651) (-3.09327, -12.4652, 3.45608) (0.0146696, -0.
(-36.4565, 221.593, -74.7354) (-1.0172, -10.9768, 3.40404) (-0.0801197, 0.0585649, -0.995063) (-7.34232, 200.675, 11.9093) (-3.63813, -13.3554, 3.41226) (0.00831985, -0.
(-36.4811, 221.296, -74.7208) (-1.42423, -15.784, 2.44618) (-0.0804647, 0.0585508, -0.995192) (-7.40923, 200.497, 11.946) (-4.15395, -13.347, 3.15492) (0.00272582, -0.
(-36.5122, 220.94, -74.7023) (-1.81448, -20.8152, 1.93749) (-0.0809643, 0.0520873, -0.995355) (-7.45406, 200.326, 11.9838) (-3.84022, -13.0979, 3.02401) (-0.00236964, -0.
(-36.5424, 220.592, -74.697) (-2.01733, -23.5186, 1.28598) (-0.0812395, 0.0484087, -0.995518) (-7.50569, 200.177, 12.0131) (-3.83375, -12.3075, 2.69747) (-0.00663379, -0.
(-36.5445, 220.278, -74.6993) (-1.44252, -23.9533, 0.710948) (-0.0814788, 0.0450649, -0.995656) (-7.53134, 200.054, 12.0363) (-3.08065, -11.0361, 2.31188) (-0.0103132, -0.
(-36.5405, 220.009, -74.6921) (-0.909284, -22.516, 0.692454) (-0.081708, 0.0422763, -0.995759) (-7.5477, 199.942, 12.0628) (-2.33899, -9.88419, 2.15629) (-0.0135937, -0.
(-36.5707, 219.72, -74.6826) (-1.56333, -22.0481, 0.732425) (-0.0824063, 0.0394816, -0.995816) (-7.56462, 199.802, 12.0779) (-1.88727, -9.88292, 1.72603) (-0.0174074, -0.
(-36.5977, 219.435, -74.6806) (-1.92529, -21.367, 0.57909) (-0.0831164, 0.0366529, -0.995866) (-7.57709, 199.672, 12.0873) (-1.47325, -9.51035, 1.29158) (-0.0208506, -0.
(-36.6394, 219.14, -74.7045) (-2.58535, -20.7856, -0.228093) (-0.084018, 0.0339843, -0.995885) (-7.59358, 199.531, 12.0968) (-1.31461, -9.44543, 0.269913) (-0.0243594, -0.
(-36.6921, 218.835, -74.7311) (-3.16166, -20.327, -0.753605) (-0.0849282, 0.031221, -0.995898) (-7.61545, 199.389, 12.0488) (-1.34652, -9.27636, -0.419069) (-0.0277518, -0.
(-36.7589, 218.567, -74.7508) (-3.80537, -18.5499, -0.847361) (-0.0859346, 0.0291541, -0.995874) (-7.64049, 199.247, 12.0272) (-1.43093, -9.02666, -0.818725) (-0.030895, -0.
(-36.8042, 218.326, -74.8076) (-3.54554, -16.3124, -1.85813) (-0.0868652, 0.0273637, -0.995844) (-7.65016, 199.129, 11.9749) (-1.03676, -8.05322, -1.84019) (-0.0337592, -0.
(-36.8659, 218.185, -74.838) (-3.78518, -12.0861, -1.68656) (-0.0879889, 0.0267364, -0.995763) (-7.65965, 199.015, 11.9259) (-0.805274, -7.33851, -2.28384) (-0.0365176, -0.
(-36.9273, 218.1, -74.9036) (-3.76401, -8.13764, -2.48999) (-0.0889779, 0.0265803, -0.995679) (-7.67645, 198.912, 11.8358) (-0.857041, -6.48315, -3.53863) (-0.0390921, -0.
(-36.9558, 217.901, -75.0259) (-2.95641, -9.22744, -4.36383) (-0.089509, 0.0252648, -0.995666) (-7.69797, 198.796, 11.7208) (-1.00457, -6.38182, -4.83147) (-0.0411455, -0.
(-36.9738, 217.765, -75.1288) (-2.22749, -7.95342, -4.83249) (-0.090062, 0.0246477, -0.995631) (-7.70346, 198.703, 11.6195) (-0.662029, -5.73498, -5.13813) (-0.0429497, -0.
(-36.995, 217.676, -75.1984) (-1.8839, -6.01501, -4.1583) (-0.0906447, 0.0242753, -0.995587) (-7.70295, 198.636, 11.5447) (-0.323112, -4.62632, -4.54975) (-0.0444222, -0.
(-37.0116, 217.606, -75.2234) (-1.68086, -4.50667, -2.63449) (-0.0913167, 0.0238831, -0.995535) (-7.68574, 198.598, 11.5122) (0.281407, -3.32914, -3.12452) (-0.0455397, -0.
(-37.0105, 217.487, -75.2567) (-1.02743, -5.11909, -2.04984) (-0.0918191, 0.0227599, -0.995516) (-7.66445, 198.562, 11.4834) (0.687038, -2.57941, -2.29792) (-0.0465318, -0.
(-37.0052, 217.339, -75.2756) (-0.671851, -6.28654, -1.40367) (-0.0924099, 0.0212477, -0.995494) (-7.63068, 198.533, 11.4721) (1.20991, -2.03526, -1.43454) (-0.0474157, -0.
(-36.9975, 217.27, -75.2723) (-0.352469, -5.78087, -0.510272) (-0.093007, 0.0201382, -0.995462) (-7.58897, 198.515, 11.4775) (1.66492, -1.44356, -0.57127) (-0.0482543, -0.
(-36.9621, 217.089, -75.2625) (0.488145, -6.25807, 0.107086) (-0.0936864, 0.0186109, -0.995428) (-7.51886, 198.494, 11.4925) (2.61204, -1.27017, 0.102882) (-0.0494073, -0.
(-36.8903, 216.931, -75.2619) (1.86079, -6.9908, 0.163027) (-0.0944564, 0.0169017, -0.995386) (-7.40753, 198.47, 11.5007) (4.12341, -1.23291, 0.261242) (-0.0509077, -0.
(-36.8267, 216.806, -75.272) (2.3313, -6.39183, -0.0802336) (-0.0951091, 0.0155068, -0.995346) (-7.30646, 198.471, 11.5001) (4.59509, -0.568137, 0.111779) (-0.0518261, -0.
```

Figura 2.4. Archivo de texto creado para la captura de la firma de un usuario

En cada fila del archivo de texto se almacenan los datos capturados en cada *frame*. Cada uno de los valores corresponde a cada uno de los datos capturados, los cuales se detallarán más adelante. Como se puede observar, algunos de los valores se almacenan en grupos. Normalmente, cada uno de estos grupos está formado por las tres coordenadas de una misma característica, como por ejemplo los 3 primeros, que corresponden a las posiciones respecto a las coordenadas (x,y,z) de la punta del dedo índice.

El protocolo de captura de datos diseñado consta de las siguientes 4 etapas, de acuerdo a [5]:

- **Etapla 1.** Detección de la mano del usuario.

El usuario ha de colocar la mano con la palma hacia abajo y los dedos bien separados para que el sensor identifique los dedos y la palma. A continuación el programa de captura pasa a estado de espera y muestra el texto “Esperando”, como se puede ver en la figura 2.5.

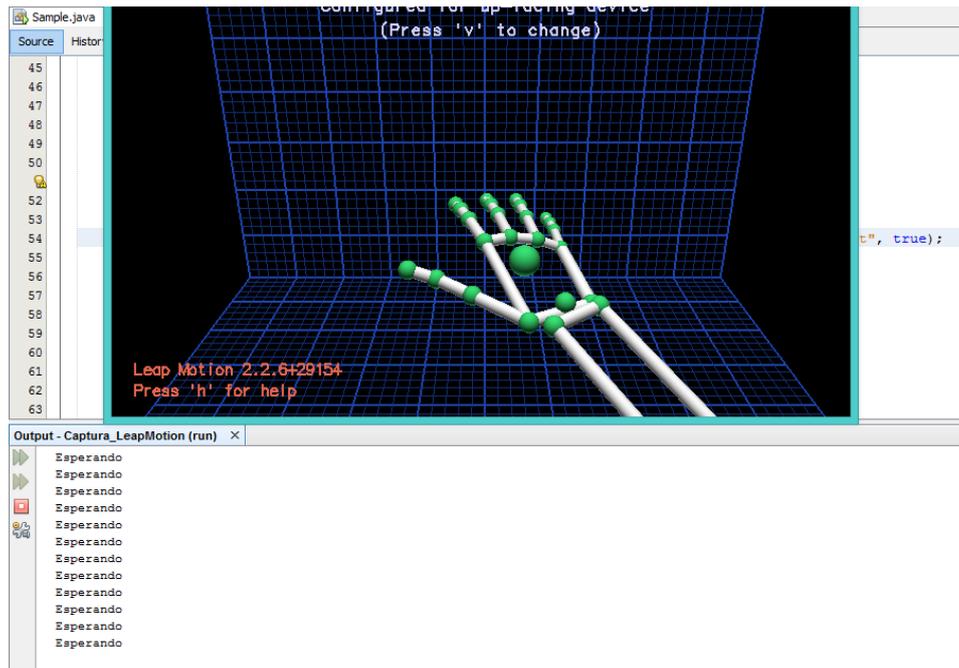


Figura 2.5. Etapa 1 del protocolo de realización del trazo de la firma

- **Etapa 2.** Posición de inicio de la firma.

El usuario ha de colocar la mano en la posición mostrada en la figura 2.6, como si de una pistola se tratara. El programa sigue en estado de espera.

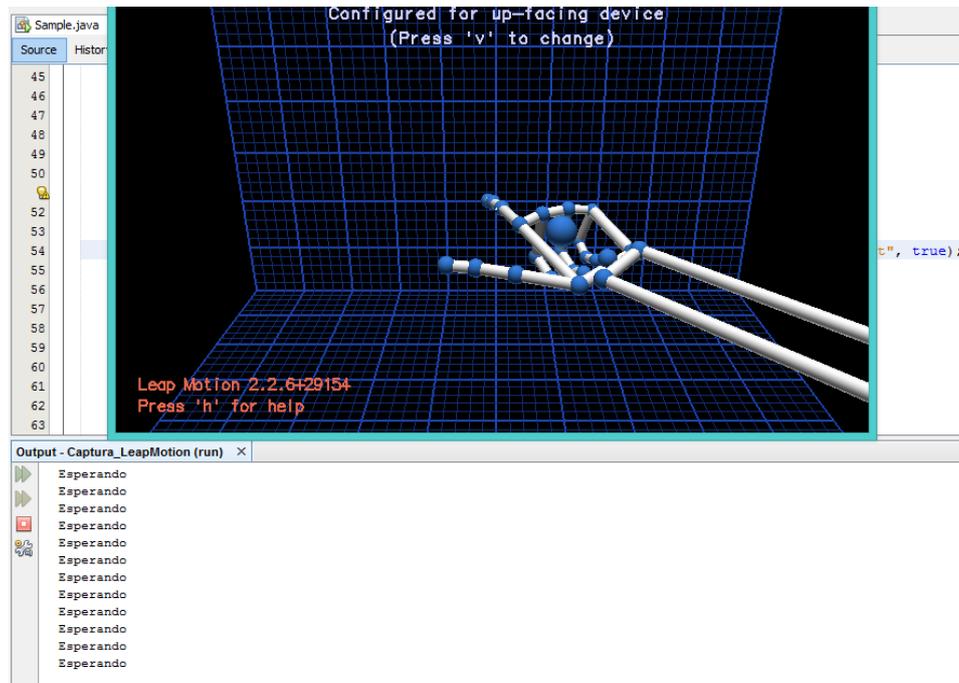


Figura 2.6. Etapa 2 del protocolo de realización del trazo de la firma

- **Etapa 3.** Inicio de la firma.

El usuario ha de cerrar el dedo pulgar cuando desee comenzar el trazo y el programa pasa a estado de captura, mostrando por pantalla el texto “Almacenando”, como se observa en la figura 2.7. En este momento el usuario ha de realizar el trazo correspondiente a su firma.

- **Etapa 4.** Fin de la firma.

Cuando el usuario finaliza el trazo correspondiente a su firma ha de volver a abrir el dedo pulgar, con lo que el programa vuelve a mostrar por la pantalla el texto “Esperando”, como se puede observar en la figura 2.8.

El sensor en todo momento se encuentra capturando datos. Todos aquellos datos capturados por el sensor mientras el texto “Esperando” aparece en pantalla son descartados. Todos aquellos datos capturados mientras el texto “Almacenando” aparece son escritos por el programa de forma secuencial en el archivo de texto correspondiente al usuario que firma.

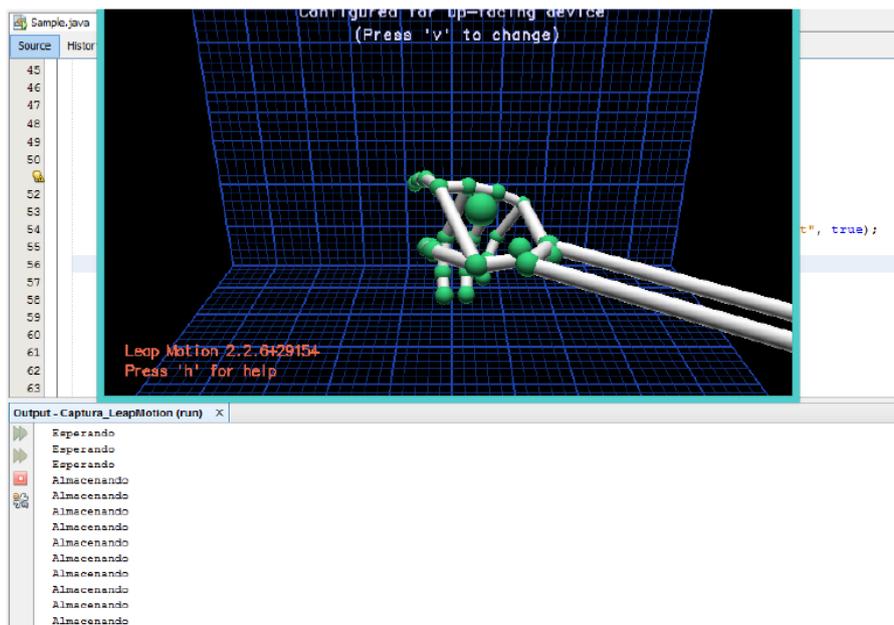


Figura 2.7. Etapa 3 del protocolo de realización del trazo de la firma

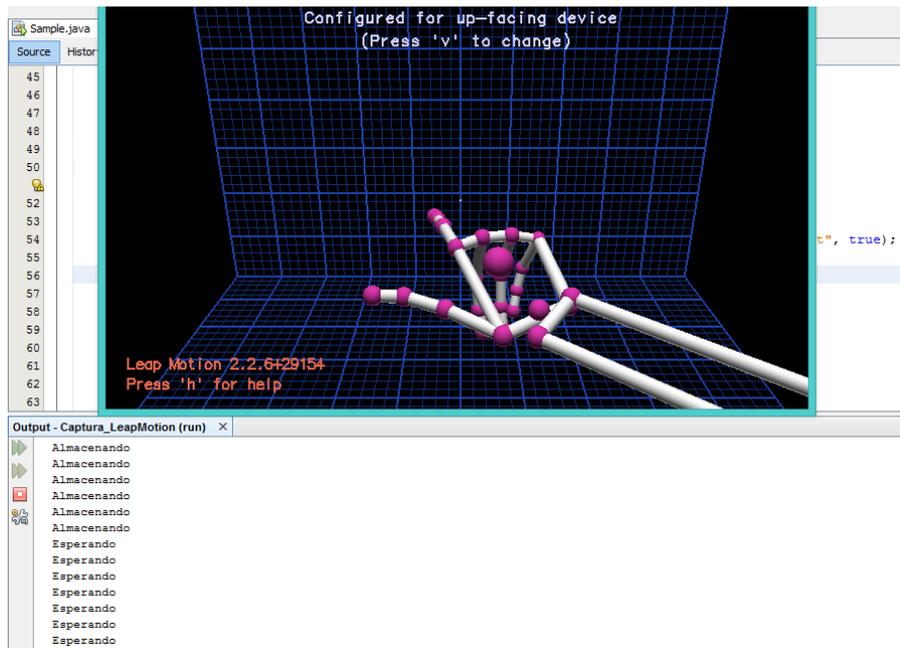


Figura 2.8. Etapa 4 del protocolo de realización del trazo de la firma

Como guía visual para que el usuario realice la firma, se muestra su mano y parte del trazo en el Visualizador del SDK de *Leap Motion*, como se puede ver en la figura 2.9.

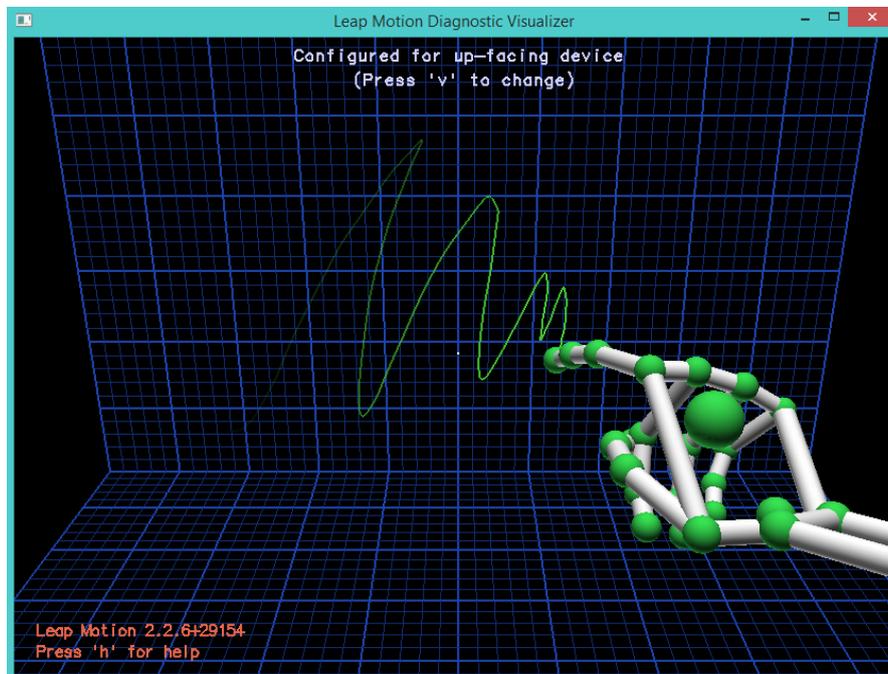


Figura 2.9. Visualizador del SDK de Leap Motion para observar la mano y el trazo

2.2.2. Datos capturados

A continuación, se detalla los datos capturados en cada uno de los *frames* de los que consta cada trazo realizado. Para ello, en la figura 2.10 se muestra un extracto del código en *Java* utilizado para la captura de los datos y su escritura en el archivo de texto correspondiente.

```
52         try
53         {
54             fichero = new FileWriter("mayo_30.txt", true);
55             pw = new PrintWriter(fichero);
56             pw.println(finger.tipPosition()
57                 +" "+finger.tipVelocity()
58                 +" "+finger.direction()
59                 +" "+hand.palmPosition()
60                 +" "+hand.palmVelocity()
61                 +" "+hand.palmNormal()
62                 +" "+hand.direction().pitch()
63                 +" "+hand.direction().yaw()
64                 +" "+hand.direction().roll());
```

Figura 2.10. Código para la captura y escritura de los datos en el archivo de texto mayo_30.txt

Como se observa en la figura 2.10, en la escritura en el archivo de texto se realizan 9 llamadas a métodos propios de *Leap Motion* que devuelven distintos datos. Éstos hacen referencia a los objetos *finger* (dedo) y *hand* (mano). En el caso particular de este proyecto, el código ha sido modificado para utilizar solamente las llamadas a *finger* que tengan como objeto particular el **dedo índice**.

Los dos primeros métodos capturan los valores de la posición y la velocidad del extremo más alejado de la falange distal del dedo índice, de acuerdo a la figura 2.11. En este caso *pointable* refiere al objeto *finger*.

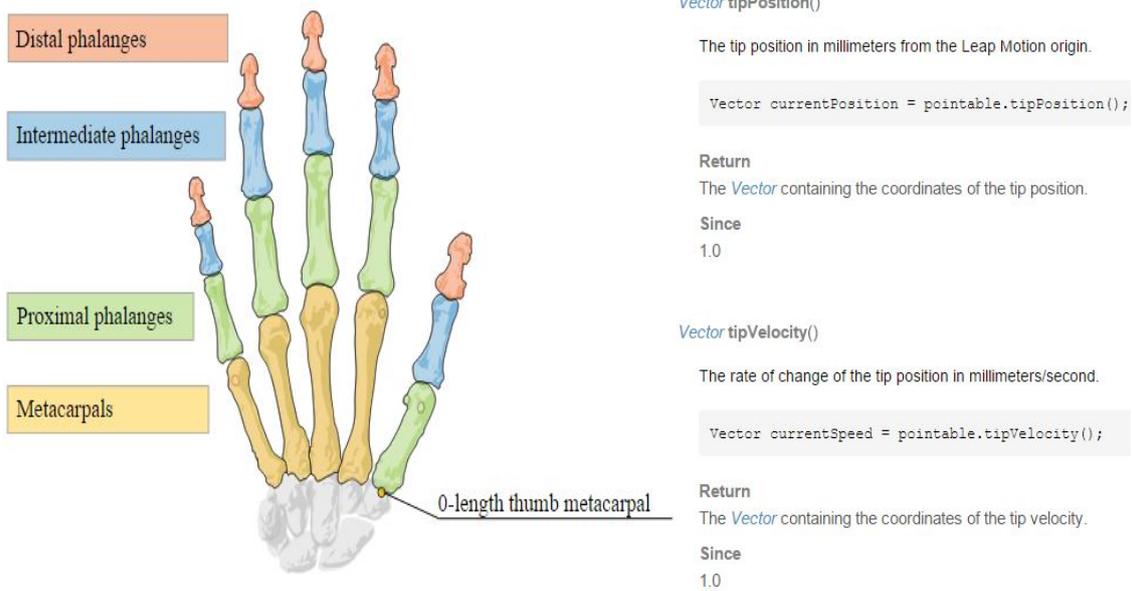


Figura 2.11. Definición del esqueleto de la mano y los métodos tipPosition y tipVelocity

El tercero de los métodos que aparecen en la figura 2.10, queda definido de acuerdo a la figura 2.12. Al igual que en el caso anterior, *pointable* refiere al objeto *finger*.

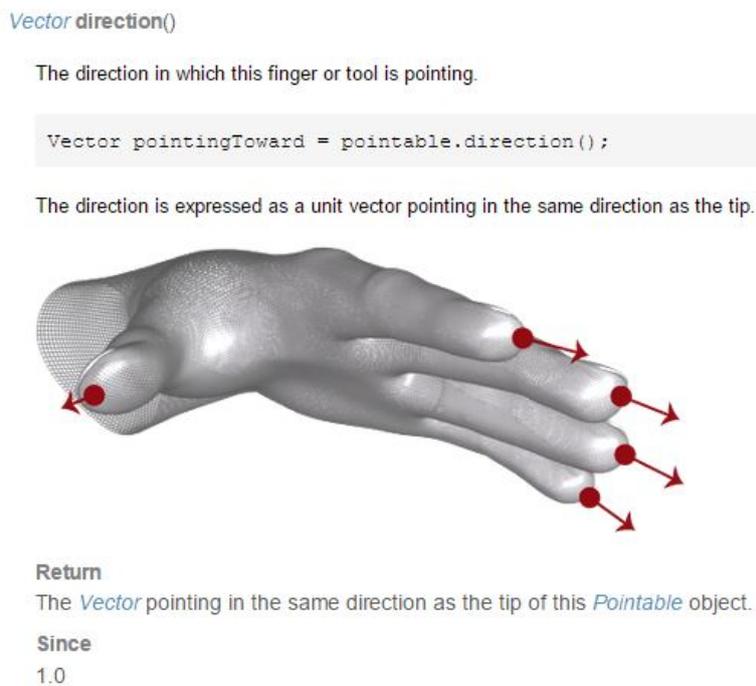


Figura 2.12. Definición del método direction

Capítulo 2. Base de datos

Los siguientes 6 métodos se refieren a métodos que devuelven información sobre la palma de la mano. Estos quedan definidos de acuerdo a las figuras 2.13 y 2.14.

Vector palmPosition()

The center position of the palm in millimeters from the Leap Motion *Controller* origin.

```
Vector handCenter = hand.palmPosition();
```

Return

The *Vector* representing the coordinates of the palm position.

Since

1.0

Vector palmVelocity()

The rate of change of the palm position in millimeters/second.

```
Vector handSpeed = hand.palmVelocity();
```

Return

The *Vector* representing the coordinates of the palm velocity.

Since

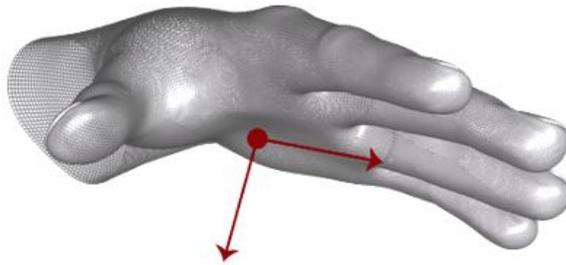
1.0

Figura 2.13. Definición de los métodos palmPosition y palmVelocity

Vector palmNormal()

The normal vector to the palm.

If your hand is flat, this vector will point downward, or "out" of the front surface of your palm.



The direction is expressed as a unit vector pointing in the same direction as the palm normal (that is, a vector orthogonal to the palm).

You can use the palm normal vector to compute the roll angle of the palm with respect to the horizontal plane:

```
float pitch = hand.direction().pitch();  
float yaw = hand.direction().yaw();  
float roll = hand.palmNormal().roll();
```

Return

The *Vector* normal to the plane formed by the palm.

Since

1.0

Figura 2.14. Definición de los métodos palmNormal, direction.pitch, direction.yaw y direction.roll

Además, en la figura 2.14, se puede observar que los valores de los ángulos *pitch* (cabeceo), *yaw* (guiñada) y *roll* (alabeo) se obtienen a partir del vector de la normal de la palma. Estos tres ángulos se definen para un vector como se muestra en la figura 2.15.

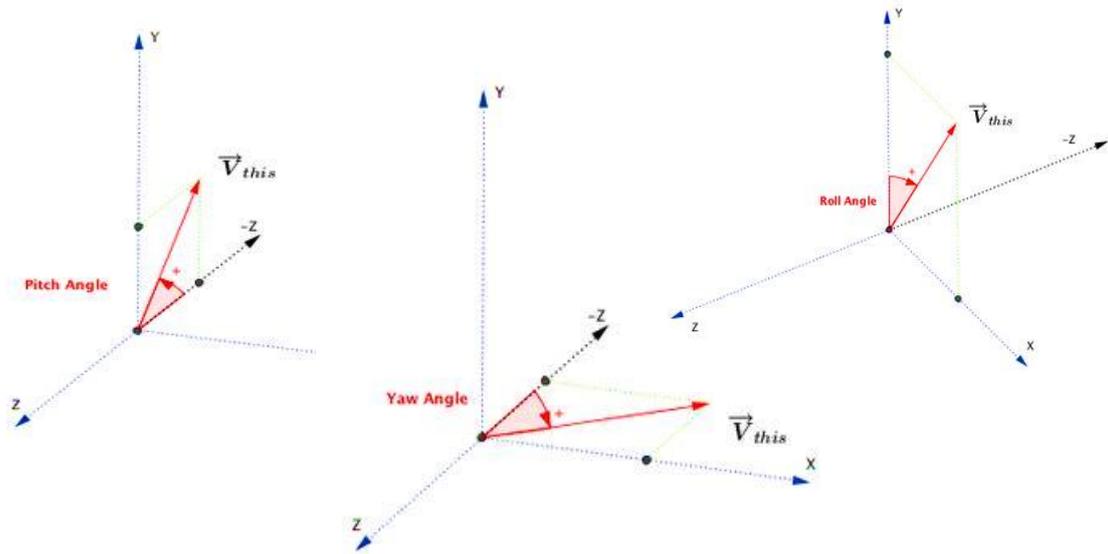


Figura 2.15. Definición de los ángulos pitch, yaw y roll

En total se capturan 21 valores por *frame*, que en el archivo de texto aparecen agrupados en función al método que los devuelve. A modo de resumen de los valores capturados, en la tabla 2.2 se muestra una lista de los mismos y de las unidades correspondientes, agrupados en función al método utilizado para obtenerlos.

Método	Datos	Unidades
finger.tipPosition()	coordenada x	milímetros
	coordenada y	milímetros
	coordenada z	milímetros
finger.tipVelocity()	velocidad coordenada x	milímetros / segundos
	velocidad coordenada y	milímetros / segundos
	velocidad coordenada z	milímetros / segundos
finger.direction()	coordenada x del vector dirección del índice	
	coordenada y del vector dirección del índice	
	coordenada z del vector dirección del índice	
hand.palmPosition()	coordenada x	milímetros
	coordenada y	milímetros
	coordenada z	milímetros
hand.palmVelocity()	velocidad coordenada x	milímetros / segundos
	velocidad coordenada y	milímetros / segundos
	velocidad coordenada z	milímetros / segundos
Hand.palmNormal()	coordenada x del vector normal a la palma	
	coordenada y del vector normal a la palma	
	coordenada z del vector normal a la palma	
Hand.direction().pitch()	ángulo pitch de la palma	radianes
Hand.direction().yaw()	ángulo yaw de la palma	radianes
Hand.direction().roll()	ángulo roll de la palma	radianes

Tabla 2.2. Listado de los datos capturados junto a los métodos que los devuelven

2.2.3. Almacenamiento en la base de datos

A la hora de recoger las muestras de cada usuario, se les solicita a los mismos que realicen entre 15 y 20 firmas, dependiendo del manejo del sensor.

Una vez se obtienen todas las muestras o firmas, se deben formatear los archivos de texto para su lectura y procesado automático en *MatLab*. Haciendo referencia a la figura 2.4, se puede observar que los valores numéricos se encuentran entre comas y paréntesis, que son caracteres que dificultan la lectura automática de los mismos. Para eliminar estos caracteres que no pertenecen a los valores de los datos capturados, se realiza un formato de los archivos de texto. De esta manera los archivos de texto formateados quedan como se presenta en la figura 2.16.

```
|-25.451500 163.575000 -26.987900 3.876110 -2.466950 9.942600 -0.118613 -0.001633 -0.992939 8.499800 155.550000 55.521600 3.573790 -1.526920 10.666800 0.04551
-25.420200 163.581000 -26.920300 3.675800 -1.152110 9.350700 -0.119664 -0.001495 -0.992813 8.591700 155.552000 55.569500 5.438170 -1.064540 9.290960 0.04328
-25.396300 163.612000 -26.861000 3.248380 0.729426 8.672430 -0.120647 -0.000825 -0.992695 8.670150 155.553000 55.618000 6.399290 -0.733598 8.067330 0.041323
-25.365300 163.635000 -26.816200 3.274910 1.697300 7.723310 -0.121374 -0.000404 -0.992607 8.743130 155.559000 55.641100 6.925280 -0.368687 6.855800 0.039609
-25.339700 163.658000 -26.777800 3.114870 2.279470 6.840910 -0.122009 -0.000052 -0.992529 8.805100 155.564000 55.667200 6.977600 -0.126684 5.825440 0.038155
-25.321500 163.686000 -26.745100 2.740500 2.834010 6.035080 -0.122520 0.000355 -0.992466 8.855240 155.591000 55.690900 6.514020 0.067281 4.999220 0.037013
-25.297600 163.725000 -26.711400 2.582690 3.719150 5.470920 -0.122500 0.000944 -0.992468 8.855240 155.591000 55.737800 4.915710 0.735055 5.101380 0.036162
-25.294500 163.750000 -26.687500 1.866170 3.891980 4.753870 -0.122990 0.001327 -0.992407 8.883720 155.612000 55.756700 4.477700 1.171930 4.322290 0.035206
-25.290000 163.788000 -26.664600 1.404800 4.527840 4.190660 -0.123472 0.001915 -0.992346 8.913900 155.641000 55.774800 4.208490 1.758750 3.723460 0.034288
-25.294600 163.847000 -26.641700 0.628869 5.770220 3.810230 -0.124115 0.002917 -0.992264 8.940900 155.667000 55.791000 3.913480 2.094130 3.225930 0.033468
-25.302800 163.894000 -26.617200 -0.065873 6.138280 3.577190 -0.124761 0.003725 -0.992180 8.965420 155.695000 55.809900 3.621280 2.367920 2.945640 0.032867
-25.308300 163.925000 -26.595900 -0.486244 5.720420 3.301800 -0.125408 0.004241 -0.992096 8.994380 155.720000 55.825700 3.543230 2.507370 2.643900 0.032389
-25.297400 163.948000 -26.572900 -0.249441 5.117250 3.145410 -0.125902 0.004434 -0.992033 9.028060 155.757000 55.849600 3.630800 2.979150 2.672420 0.031841
-25.282300 163.988000 -26.544900 0.028820 5.281710 3.196930 -0.126611 0.004845 -0.991941 9.000480 155.790000 55.871200 4.269500 3.205930 2.620880 0.031025
-25.271900 164.022000 -26.505900 0.156415 5.049880 3.567190 -0.127296 0.004873 -0.991853 9.130790 155.833000 55.902300 4.672960 3.653730 2.874690 0.030240
-25.249000 164.065000 -26.460300 0.593137 5.229240 4.055590 -0.127794 0.005091 -0.991788 9.174720 155.868000 55.947100 4.773260 3.759360 3.481440 0.029394
-25.343500 164.049000 -26.883600 -2.236630 3.819460 -9.805850 -0.132133 0.015114 -0.991117 9.219470 154.461000 55.326100 4.872010 -40.386300 -16.489300 0.00:
-25.302300 164.206000 -26.657400 -0.185522 6.185930 1.572060 -0.134391 0.015291 -0.990810 9.329290 154.517000 55.545000 4.432790 -14.441700 -1.267560 -0.0050
-25.275800 164.238000 -26.566800 0.578592 5.725010 3.952440 -0.135493 0.014913 -0.990666 9.419810 154.556000 55.618500 6.026140 -9.377910 1.326330 -0.007346
-25.238300 164.262000 -26.482100 1.559560 5.005090 5.510860 -0.136322 0.014376 -0.990560 9.505800 154.592000 55.690700 7.054900 -5.771020 3.183970 -0.009345
-25.208100 164.301000 -26.416000 2.034720 4.877500 6.093670 -0.137275 0.014044 -0.990433 9.598410 154.628000 55.735500 8.012830 -3.128750 3.708410 -0.010922
-25.173100 164.365000 -26.358200 2.586820 5.641860 6.267580 -0.137622 0.014272 -0.990382 9.650130 154.658000 55.788100 7.460690 -1.365670 4.333980 -0.012156
-25.131400 164.442000 -26.314900 3.194580 6.601520 5.949540 -0.137935 0.014376 -0.990337 9.709410 154.718000 55.827400 7.287940 0.828390 4.380560 -0.013006
-25.100000 164.550000 -26.291700 3.302940 8.316890 5.108970 -0.138279 0.014848 -0.990282 9.760890 154.787000 55.843600 6.922430 2.725000 3.710120 -0.013733
-25.046200 164.702000 -26.242000 4.072540 10.840400 5.296530 -0.138862 0.014722 -0.990202 9.853540 154.937000 55.881800 7.916440 6.591100 3.892290 -0.014668
-24.999000 164.865000 -26.198500 4.362270 13.008000 5.264840 -0.139345 0.015094 -0.990129 9.928390 155.063000 55.916000 8.099790 8.699990 3.903340 -0.015449
-24.953600 165.061000 -26.156500 4.489530 15.766800 5.224700 -0.139923 0.016512 -0.990025 10.012300 155.149000 55.935900 8.511150 9.008570 3.471270 -0.015788
-24.903500 165.290000 -26.100500 4.643550 18.655600 5.618420 -0.141154 0.017123 -0.989840 10.152400 155.332000 55.956200 10.535500 12.223100 3.167430 -0.0166
-24.846500 165.519000 -26.063400 5.007330 20.888700 5.314410 -0.141804 0.017989 -0.989731 10.252100 155.507000 55.977600 10.783900 14.341500 2.979520 -0.0167
-24.820000 165.789000 -26.002500 4.389890 23.834600 5.801050 -0.142357 0.018328 -0.989646 10.324300 155.791000 56.029100 10.121400 19.227000 3.761620 -0.0165
-24.852200 166.117000 -25.995800 2.082050 28.046000 4.541350 -0.143174 0.020209 -0.989491 10.350900 156.012000 56.008700 8.235310 20.875300 2.132850 -0.0155
-24.889300 166.539000 -25.979700 0.233492 34.490200 3.943360 -0.144327 0.023420 -0.989253 10.403700 156.258000 55.977800 7.658320 22.840500 0.617329 -0.0149
-25.021900 166.984000 -25.965100 -3.983370 39.787600 3.462880 -0.145448 0.026764 -0.989004 10.367300 156.522000 55.940400 4.496810 24.821300 -0.693240 -0.01:
-25.330200 167.645000 -25.996900 -12.582600 50.683800 1.702130 -0.148318 0.030369 -0.988473 10.289300 157.006000 55.803000 0.906355 33.056300 -4.722810 -0.0:
-25.718100 168.336000 -25.965500 -21.201000 59.881300 2.368050 -0.150191 0.035584 -0.988016 10.049100 157.400000 55.748200 -6.699560 36.305400 -5.142900 -0.0:
```

Figura 2.16. Archivo de texto formateado a partir del archivo de texto creado para la captura de la firma

Como se puede observar, ahora se muestran todos los valores de cada *frame* en una fila separados por un espacio, obteniendo una estructura que facilita su lectura automática. El siguiente paso, es cargar los archivos automáticamente en matrices, manteniendo el formato, es decir, estas matrices poseen tantas filas como *frames* y tantas columnas como datos capturados en cada *frame*.

Para cada una de las muestras se guarda un archivo *.mat* que contiene la matriz de datos.

A continuación, se seleccionan las 10 mejores muestras de todas las realizadas por el usuario. Para ello se utiliza el algoritmo *Dynamic Time Warping* (DTW) sobre los valores de las coordenadas (x,y,z) del extremo del dedo índice, que correspondería a la imagen que veríamos de la firma pero en 3 dimensiones, es decir, sobre las 3 primeras columnas de las matrices. Este algoritmo es bastante utilizado en problemas similares al estudiado [18] [21] [25].

A modo de breve explicación sobre el algoritmo *Dynamic Time Warping*, cabe mencionar que su objetivo es evaluar el coste de alinear dos secuencias temporales. En otras palabras, trata de medir la similitud entre dos señales, en este caso los trazos tridimensionales de las diferentes firmas del usuario, que no tienen por qué estar alineadas temporalmente (figura 2.17).

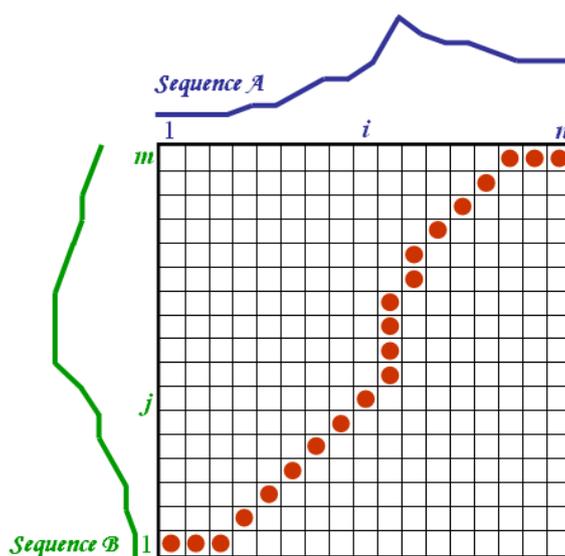


Figura 2.17. Representación de la matriz de costes del algoritmo Dynamic Time Warping para las secuencias A y B

El DTW evalúa el coste de alinear cada punto de la secuencia A con cada punto de la secuencia B, creando una matriz de salida donde se muestran dichos costes. En este caso el coste se evalúa respecto a la distancia Euclídea de los puntos (ecuación 2.1) [5].

$$d_E(P, Q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + (p_3 - q_3)^2} \quad (2.1)$$

En la ecuación 2.1 se particulariza la distancia Euclídea d_E entre dos puntos P y Q definidos en un espacio tridimensional.

El camino de menor coste de alineamiento es el marcado en rojo en la figura 2.17. Este camino hace referencia al menor coste para alinear las dos secuencias. Para el objetivo buscado de evaluar el coste, con el cálculo del camino es suficiente. Sin embargo, en función del resultado buscado al aplicar el algoritmo, se pueden procesar los datos de la matriz de costes para alinear ambas secuencias.

Para mostrar la idea del uso de las coordenadas xyz en la evaluación de la similitud de los trazos, se muestra un trazo aéreo caracterizado por las mismas, así como sus proyecciones bidimensionales para facilitar su identificación visual. En la figura 2.18 se muestra un ejemplo del trazo de la palabra "trazo". Solamente se muestra el trazo caracterizado por las coordenadas (x,y,z) del extremo del dedo índice.

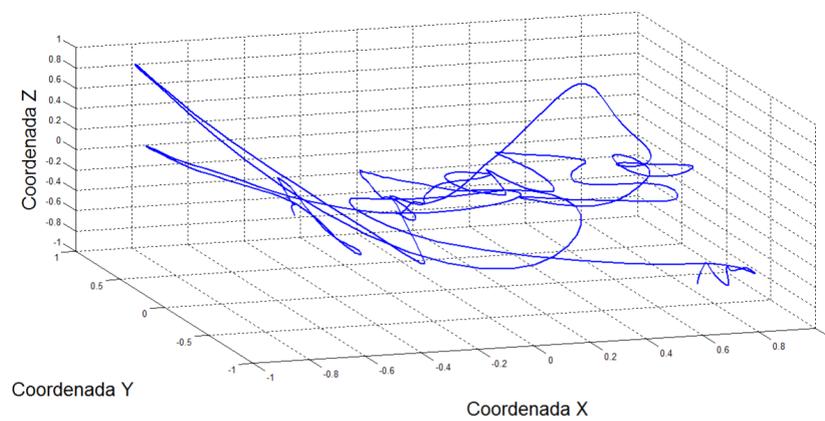


Figura 2.18. Trazo tridimensional caracterizado por las coordenadas (x,y,z) de la palabra "trazo"

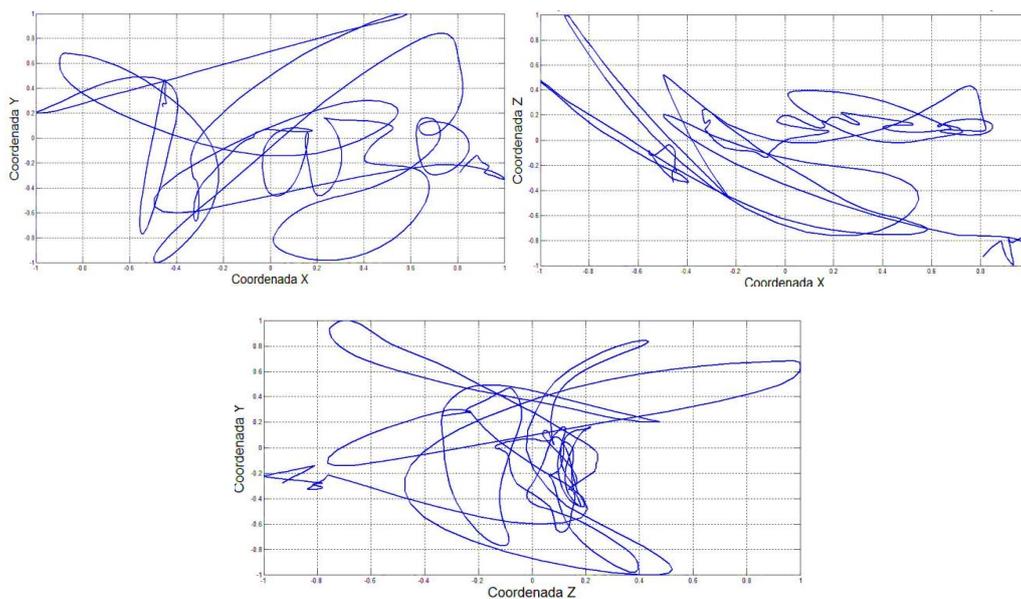


Figura 2.19. Proyecciones del trazo tridimensional de la figura 23 sobre los planos (x,y) , (x,z) y (y,z)

En la figura 2.19 se puede observar las proyecciones del trazo tridimensional de la figura 2.18 sobre los planos (x,y) , (x,z) y (y,z) , respectivamente.

2.3. Características de los usuarios

La base de datos proporcionada consta de un total 39 usuarios, como bien se ha mencionado en capítulos anteriores. Luego, se realiza una primera fase de ampliación de la misma hasta llegar a 70 usuarios y, por último, en una segunda etapa, se alcanzarán los 100 usuarios que se tienen como objetivo.

Para la base de datos de 70 usuarios, cada uno de éstos consta de 10 muestras, lo que hace un total de 700 muestras. Los usuarios que forman la base de datos son de diferente sexo y edad, y quedan distribuidos como se detalla a continuación.

En la tabla 2.3 se muestra la distribución de los usuarios por sexo.

Sexo	Distribución
Masculino	65,71%
Femenino	34,29%

Tabla 2.3. Distribución por sexo de los usuarios de la base de datos de 70 usuarios

En la figura 2.20 se puede observar la distribución de los mismos por edad. Se observa que la media es de unos casi 28 años, con una desviación típica de poco más de ± 11 años.

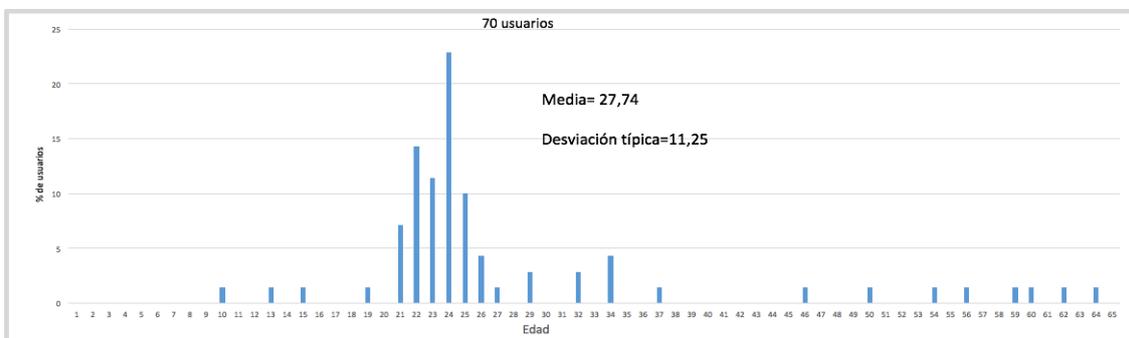


Figura 2.20. Distribución de edad de los usuarios de la base de datos de 70 usuarios

Se observa que el rango de edad donde se concentra la mayoría de los firmantes está entre 21 a 26 años.

Por último, como se ha comentado en los objetivos del proyecto, la base de datos final, tras la ampliación de la misma en una segunda etapa, consta de un total de 100 usuarios y 10 muestras para cada uno de ellos, lo que hace un total de 1000 muestras. Los usuarios que forman la base de datos son de diferente sexo y edad, y quedan distribuidos como se detalla a continuación.

En la tabla 2.4 se muestra la distribución de los usuarios por sexo.

Sexo	Distribución
Masculino	65%
Femenino	35%

Tabla 2.4. Distribución por sexo de los usuarios de la base de datos de 100 usuarios

En la figura 2.21 se puede observar la distribución de los mismos por edad. Se observa que la media es de unos 29 años, con una desviación típica de poco más de ± 12 años.

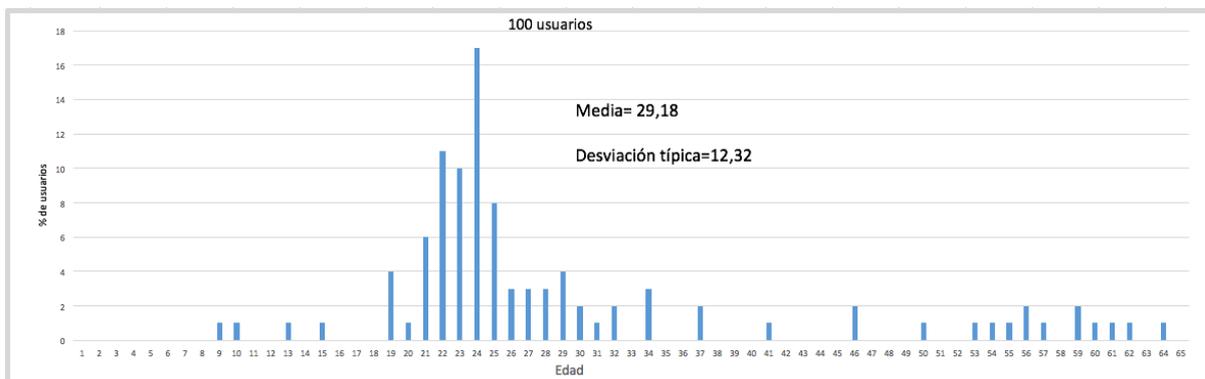


Figura 2.21. Distribución de edad de los usuarios de la base de datos de 100 usuarios

Se observa que la base de datos final tiene una mayor distribución de edades (entre 9 y 64 años) con respecto a la base de datos inicial (entre 13 y 57), pero sigue habiendo una distribución por sexo muy similar y más del 50% de la distribución se encuentra entre los 21 y 25 años.

Capítulo 3. Sistema de parametrización y clasificación

En este capítulo se explica el funcionamiento del clasificador que se usa en este proyecto y se indica cómo se extrae y se trata la información de las señales de las muestras correspondientes a la base de datos.

3.1. Parametrización

Los vectores de características son extraídos de las señales que forman cada uno de los grupos estudiados para representar el patrón formado por cada firma, siendo un patrón, una entidad a la que se puede dar un nombre y que está representada por un conjunto de propiedades medidas y las relaciones entre ellas (vector de características). Un patrón puede ser una señal sonora y su vector de características, el conjunto de coeficientes espectrales extraída de ellas.

Inicialmente, para la realización de los primeros experimentos, cada conjunto de señales se divide en segmentos y a cada segmento se le calcula la media, desviación típica y la correlación entre los posibles pares de señales del mismo tipo. A esto se le denominará subconjunto.

Por ejemplo, para el conjunto de estudio formado por las señales generadas a partir de las coordenadas (x,y,z) del dedo índice, se extraen las medias y las desviaciones típicas de cada una de las 3 señales, así como las correlaciones entre los 3 posibles pares de señales que se pueden formar $((x,y), (x,z)$ y $(y,z))$.

Para conjuntos con un mayor número de señales no se calculan las correlaciones de todos los posibles pares, sino que se calcula la correlación entre los pares del mismo tipo de señal. Por ejemplo, para el conjunto de estudio formado por las 21 señales se obtienen las medias y las desviaciones típicas de cada señal, y se calculan las correlaciones entre los pares de señales cuyos indicadores son $(1,2), (1,3), (2,3), (4,5), (4,6), (5,6)$, etc. Donde los 3 primeros pares corresponden a las señales de posición del

Capítulo 3. Sistema de parametrización y clasificación

dedo índice, los 3 siguientes a las señales de las velocidades de las coordenadas anteriores, etc. (ver tabla 2.2).

Como se ha comentado, en cuanto a la parametrización mostrada, no se realiza sobre toda la señal a la vez, dado que de esta forma se puede perder información característica. Para intentar caracterizar cada una de las señales de la mejor forma, estas son divididas en varios segmentos, por lo que para una señal dada se calculan varias medias y varias desviaciones típicas en función del número de segmentos.

De la misma forma se procede con la correlación, la cual se calcula para los pares de segmentos y no para los pares de señales directamente, por lo que para un par de señales se obtienen tantas correlaciones como segmentos se utilicen para cada una de ellas.

En la figura 3.1 se puede observar las señales de las coordenadas (x,y,z) de una muestra divididas en 10 segmentos a la vez que se representa la media de cada uno de estos segmentos [5].

La parametrización indicada se realiza dividiendo las señales de 2 a 30 segmentos de igual longitud. Por tanto, para cada conjunto de estudio se obtienen varios vectores de características diferentes.

Por otro lado, uno de los experimentos será añadir otra serie de parámetros para evaluar si aportan información discriminante. Estos parámetros serán la entropía de Shannon, la curtosis y la asimetría (*skewness*). Éstos se calculan de igual manera que el resto de parámetros antes mencionados. Es decir, se calculan sobre cada segmento.

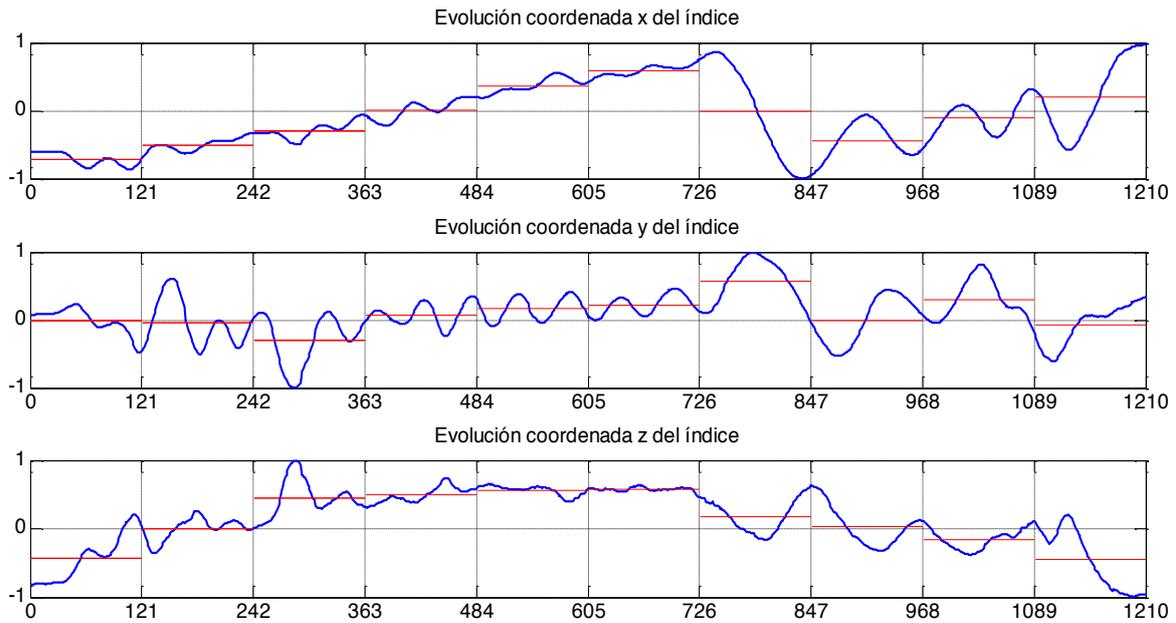


Figura 3.1. Medias de cada uno de los segmentos de las señales (x,y,z) del dedo índice

3.2. Parámetros

Como se mencionó en el apartado anterior, en primer lugar la parametrización consistirá en calcular la media, desviación típica y correlación de cada segmento para realizar los primeros experimentos y, luego, se añadirán además la entropía de Shannon, la curtosis, y la asimetría (*skewness*) con el fin de saber si añaden o no información. A continuación se realiza una breve descripción de cada uno de estos parámetros:

- **Media.** Es una medida de centralización. Se define como la suma de todos los datos de un conjunto dividida entre el número total de datos.

$$\bar{x} = \frac{x_1+x_2+\dots+x_n}{N} \quad (3.1)$$

- **Desviación típica.** Medida de dispersión asociada a la media. Muestra la desviación que presentan los datos en su distribución respecto de la media de dicha distribución.

$$\sigma = \sqrt{\frac{\sum x_i^2 \cdot f_i}{N} - (\bar{x})^2} \quad (3.2)$$

- **Correlación.** Es una medida que establece la relación o dependencia que existe entre dos variables.

- Entropía de Shannon. Hace referencia a la cantidad media de información que contiene una variable aleatoria [26].

$$H(X) = - \sum P_i \log_2 x_i \quad (3.3)$$

- Curtosis. Describe el grado de “punta” o “achatamiento” de la distribución de una variable con relación a la distribución normal.

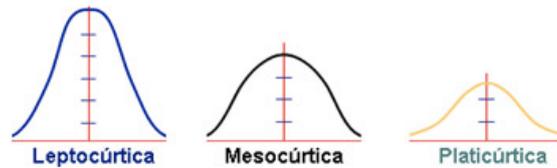


Figura 3.2. Ejemplos de curtosis

- Skewness o asimetría. Describe la simetría de la distribución alrededor de la media [27].



Figura 3.3. Ejemplos de asimetría (skewness)

Esto se hace individual y conjuntamente, es decir, a cada subconjunto estudiado se le incluye por separado estos tres parámetros, y también se les añaden de dos en dos:

- Entropía de Shannon y curtosis
- Entropía de Shannon y skewness
- Curtosis y *skewness*

Por último, se añaden los tres parámetros a la vez:

- Entropía de Shannon, curtosis y *skewness*

3.3. Descripción de un clasificador

Dado un conjunto de datos, o firmas en este caso particular, pertenecientes a distintos usuarios, el objetivo general de un clasificador es el de establecer una norma que permita determinar de forma unívoca a qué usuario pertenece una firma o un conjunto de firmas de entrada a dicho clasificador. Como ya se ha mencionado, para este proyecto se cuenta con 10 firmas parametrizadas para cada uno de los usuarios de la base de datos.

3.4. Clasificador implementado

Actualmente, las máquinas de soporte vectorial (SVM) se han convertido en una herramienta poderosa para resolver problemas de clasificación no lineal. **En este proyecto**, el proceso de clasificación se desarrollará aplicando una reformulación conocida como **Máquina de Soporte Vectorial de Mínimos Cuadrados** (*Least Squares Support Vector Machine* o **LS-SVM**), la cual trabaja con un modelo de minimización basada en funciones y polinomios de Lagrange [28].

3.5. Máquinas de Soporte Vectorial

Las Máquinas de Soporte Vectorial (SVM) son algoritmos de aprendizaje supervisado, es decir, el sistema aprende a partir de ejemplos etiquetados de forma que se provee al sistema de la respuesta correcta para una entrada dada [29]. O, dicho de otro modo, se conoce la existencia de un conjunto de clases a partir de las cuales se determinarán las reglas para asignar cada nueva entrada a la clase a la que pertenece. Esto implica contar con una serie de ejemplos debidamente etiquetados para realizar la etapa de aprendizaje.

Estas máquinas fueron desarrolladas como una herramienta robusta y sólida para regresión y clasificación en dominios complejos, por Vladimir Vapnik y su equipo en los laboratorios AT&T [30].

La metodología base de SVM, cuando no se puede separar linealmente un conjunto de

datos, se puede resumir de la siguiente forma: Si se requiere clasificar un conjunto de datos (representados en un plano n-dimensional) no separables linealmente, se toma dicho conjunto de datos y se mapea a un espacio de mayor dimensión donde si sea posible la separación lineal (esto se realiza mediante funciones llamadas *kernel*). En este nuevo plano se busca un hiperplano que sea capaz de separar en 2 clases los datos de entrada; el plano debe tener la mayor distancia posible a los puntos de ambas clases (los puntos más cercanos a este hiperplano de separación son los vectores de soporte) como se observa en la figura 3.1. Si lo que se requiere es hacer una regresión, se toma el conjunto de datos y se transforma a un espacio de mayor dimensión (donde sí se pueda hacer una regresión lineal) y en este nuevo espacio se realiza la regresión lineal pero sin penalizar errores pequeños.

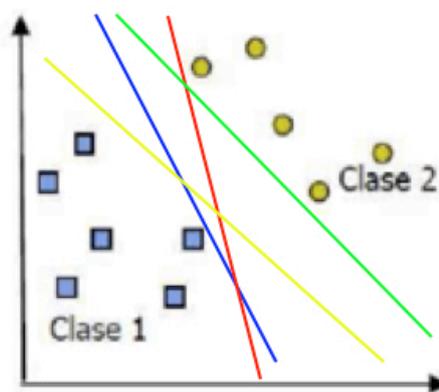


Figura 3.1. Posibles hiperplanos de separación

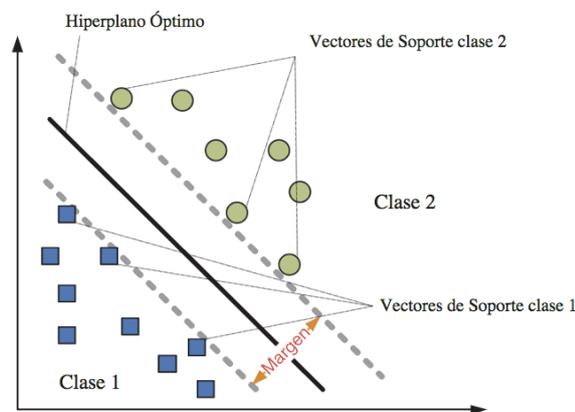


Figura 3.2. Hiperplano de separación óptima

En definitiva, el SVM es un clasificador lineal que se basa en el concepto de margen máximo a partir de los vectores de soporte. Este margen nos describe una región del espacio de características en el que no hay muestras y el plano intermedio de esta región es la solución del SVM [31].

Matemáticamente, la solución del SVM se puede obtener como solución a un problema de optimización, donde la función que queremos optimizar es el margen.

Definiendo X como un espacio de representación para las variables de entrada e $Y_i = \{+1, -1\}$ como las dos posibles clases a clasificar, la solución al hiperplano se expresa de la siguiente manera:

$$w^T x_i + b = 0 \quad (3.4)$$

con

$$w = (w_1, \dots, w_n) \quad (3.5)$$

siendo,

$w \rightarrow$ Vector ortogonal al hiperplano

$b \rightarrow$ Coeficiente de intersección

Las muestras positivas serán las que cumplen:

$$w^T x_i + b > 0 \quad (3.6)$$

y las negativas

$$w^T x_i + b < 0 \quad (3.7)$$

De esta manera, la función de clasificación se podrá expresar como se muestra en la ecuación (3.8).

$$f(x) = \text{sgn}(w^T x_i + b) \quad (3.8)$$

Teniendo en cuenta que las muestras positivas tienen etiqueta $y_i = +1$ y que las muestras negativas tienen etiqueta $y_i = -1$, estas dos condiciones se pueden juntar

dando lugar a la siguiente condición de clasificación que tienen que cumplir todas las muestras de entrenamiento:

$$y_i(w^T x_i + b) \geq 1 \quad (3.9)$$

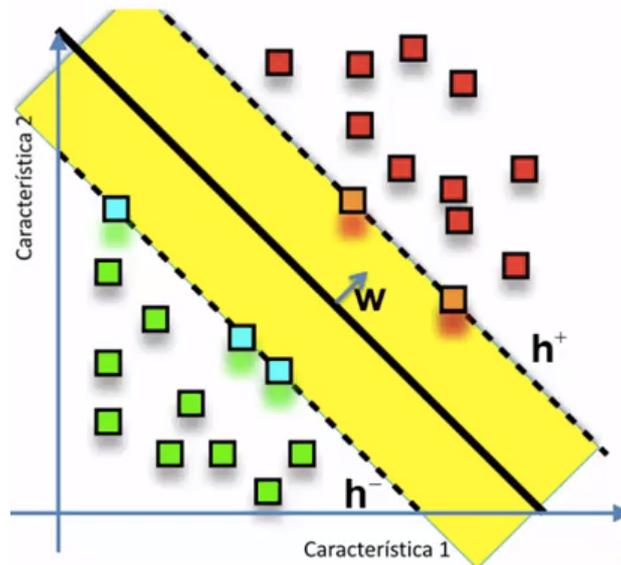


Figura 3.3. Hiperplano de separación óptima con vector ortogonal al hiperplano

De esta manera, el margen, que es la separación entre los dos hiperplanos, se puede definir como se indica en la ecuación (3.10).

$$margin = 2 \frac{1}{\|w\|} \quad (3.10)$$

Como lo que se busca es maximizar el margen, es la ecuación (3.10) la que se tiene que maximizar o, lo que sería equivalente, minimizar la ecuación (3.11).

$$\Phi(w) = \frac{1}{2} \|w\|^2 \quad (3.11)$$

sujeto a la condición de clasificación dada en la función (3.9) y siendo

$$\|w\|^2 = w^T w \quad (3.12)$$

Por lo tanto, se obtiene un problema de optimización cuadrática que se resolverá planteando una función auxiliar conocida como Lagrangiano:

$$L(x, \alpha) = f(x) + \sum_i \alpha g_i(x) \quad \forall \alpha_i \geq 0 \quad (3.13)$$

donde,

$\alpha \rightarrow$ Multiplicadores de Lagrange

$f(x) \rightarrow$ Función a optimizar

$g_i(x) \rightarrow$ Restricciones

Entonces,

$f(x) \rightarrow \Phi(w)$

$g_i(x) \rightarrow y_i(w^T x_i + b) - 1 \geq 0$

Obteniendo finalmente los valores óptimos de los multiplicadores α_i que permiten encontrar la formulación del hiperplano solución:

$$w = \sum \alpha_i y_i x_i \quad (3.14)$$

y, por lo tanto, definir la función de clasificación:

$$f(x) = \text{sgn}(\sum y_i \alpha_i x^T x_i + b) \quad (3.15)$$

Para el caso no linealmente separable, como se observa en la figura 3.4, se mapea el conjunto de características desde el espacio no linealmente separable a un espacio de dimensión superior en el que los datos sí serán linealmente separables y así resolverlo usando SVM. Para ello será necesario usar una función kernel.

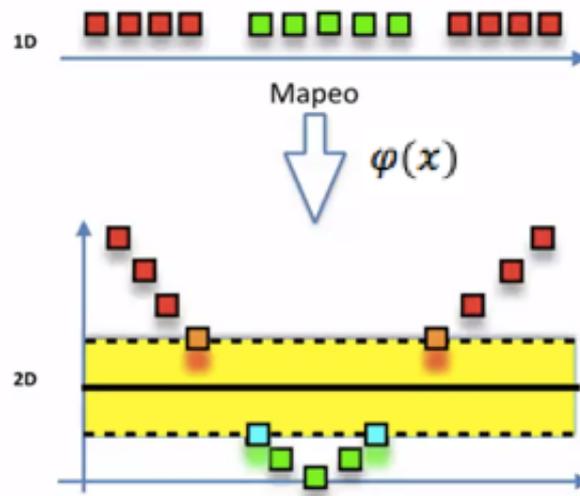


Figura 3.4. Transformación de un caso no linealmente separable a un caso linealmente separable

La función kernel será de la siguiente manera:

$$K(x, z) = \varphi(x)^T \varphi(z) \quad (3.16)$$

La función de clasificación, finalmente quedará definida como se indica en la ecuación (3.14).

$$f(x) = \text{sgn}(\sum y_i \alpha_i K(x, x_i) + b) \quad (3.17)$$

3.6. Clasificador seleccionado. LS-SVM

La forma como se soluciona el problema de encontrar un hiperplano maximizando los márgenes entre éste y los puntos de ambas clases (del conjunto de datos de entrada) define si la SVM es tradicional o es LS-SVM. SVM soluciona dicho problema mediante el principio de *structural risk minimization* mientras que LS-SVM lo soluciona mediante un conjunto de ecuaciones lineales [32]. Mientras que en SVM tradicional muchos valores de soporte son cero (valores diferentes a cero corresponden a los vectores de soporte), en LS-SVM los valores de soporte son proporcionales a los errores [33].

Como se mencionó en el apartado anterior, se usan funciones *kernel* cuando se necesita transformar los datos de entrada a un espacio de mayor dimensión donde sea posible su separación lineal, como se muestra en la figura 3.4.

Para el caso particular de este proyecto, se ha escogido un LS-SVM en el que se toman las dos clases siguientes:

- La primera clase corresponde al usuario i con etiqueta $Y_1 = +1$.
- La segunda, a todos los usuarios menos el usuario i con etiqueta $Y_1 = -1$.

Es decir, una clase corresponde al usuario a identificar y la otra clase al resto de usuarios.

De las 10 firmas pertenecientes a cada usuario, se seleccionan las pertinentes (las que sean necesarias en cada experimento) para entrenar al clasificador. El resto de firmas se

usa para la clasificación. El clasificador devolverá el vector Z con valores cercanos a +1 y -1 para cada firma que tenga que clasificar según el clasificador “crea” que la firma pertenece al usuario correspondiente (+1) o no (-1).

La elección de la función *kernel* apropiada para un problema concreto es aún motivo de investigación [33]. En este proyecto se usan dos funciones *kernel*:

- **Polynomial** (Polinómico). Utiliza una ecuación polinómica no homogénea de grado d .

$$K(x, z) = \langle x, z \rangle^d \quad (3.15)$$

- **RBF** (*Radial Base Function* o Gaussiano). Es una ecuación cuyos valores dependen solo de la distancia a un punto tomado como referencia central.

$$K(x, z) = \exp\left(-\frac{\|x-z\|^2}{2\sigma}\right) \quad (3.16)$$

Por último, será necesario ajustar los parámetros de sintonía del clasificador, que son:

- Coeficiente de dispersión, *gamma* (γ).
- Varianza (σ^2 o *sig2*). Parámetro de la función *kernel*.

Capítulo 4. Metodología

En este capítulo se indican los pasos que se siguen para evaluar el rendimiento del clasificador. Por un lado, se muestran las medidas que se realizarán para este fin y, por otro lado, se explican los experimentos que se llevarán a cabo en este proyecto.

4.1. Medidas para evaluar el rendimiento

El rendimiento de los clasificadores se mide en relación a los parámetros de *False Rejection Rate* (FRR), *False Acceptance Rate* (FAR) y *Equal Error Rate* (EER) [5].

- **False Rejection Rate.** Muestra el porcentaje de veces que el clasificador rechaza a un usuario positivo. Es decir, que no reconoce una firma que pertenece al usuario.
- **False Acceptance Rate.** Mide el número de veces que el clasificador acepta a un usuario negativo. En otras palabras, admite una firma que no pertenece al usuario.
- **Equal Error Rate.** Es el punto en el que tanto FRR y FAR poseen el mismo valor. Esta medida es la que suele caracterizar al sistema.

Tanto FRR y FAR se evalúan en función a un umbral de decisión utilizado por el clasificador, de esta manera se obtienen gráficas como la mostrada en la figura 4.1, donde se observa FRR, FAR y EER.

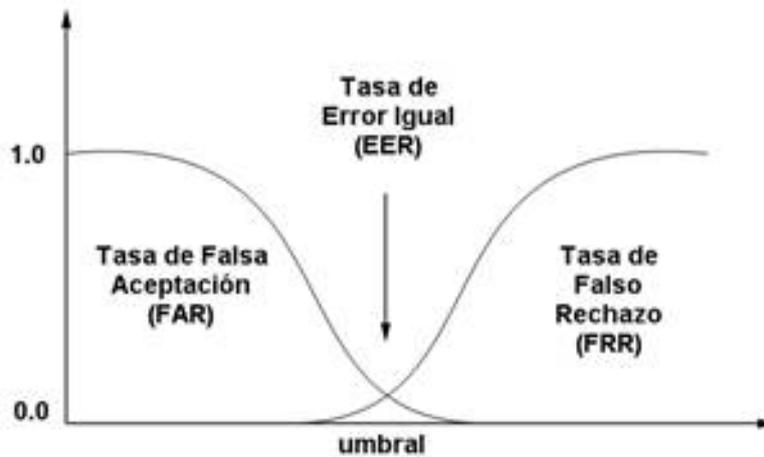


Figura 4.1. Representación gráfica de las medidas de FRR, FAR y EER

Para cada firma evaluada este clasificador devuelve dos valores de similitud respecto a las dos clases posibles. Para valorar FRR se compara la similitud con la clase correcta y se valora si es mayor o no que el umbral. En el caso de que sea mayor al umbral, se clasifica como positiva. Para el caso FAR, las muestras que se conocen negativas se evalúan respecto al umbral. El EER ofrece un punto donde los errores medidos por FAR y FRR son iguales.

4.2. Experimentos para el estudio del clasificador

En este apartado se explican cada uno de los experimentos que se realizarán. Este proyecto consta de un total de 7 experimentos, de los cuales los 5 primeros experimentos tratan de ajustar el clasificador y determinar cuál o cuáles son los conjuntos de señales que ofrecen un menor EER y, por tanto, que mejor caracterizan el trazo de cada individuo. Estos cinco primeros experimentos se realizan con la base de datos inicial de 39 usuarios. Los experimentos 6 y 7 consisten en la simulación final sobre 70 y 100 usuarios, respectivamente, haciendo uso de los conjuntos seleccionados.

En la tabla 4.1 se muestra una breve descripción o título de cada uno de los experimentos.

Experimento 1	Análisis en detalle con el clasificador
Experimento 2	Ajuste del clasificador y selección del <i>kernel</i>
Experimento 3	Estudio de los datos añadiendo nuevos parámetros estadísticos
Experimento 4	Estudio de la influencia de la tercera dimensión con respecto al sistema plano
Experimento 5	Fusión de los conjuntos
Experimento 6	Simulación del sistema con una base de datos ampliada a 70 usuarios
Experimento 7	Simulación con la base de datos final

Tabla 4.1. Breve descripción de cada experimento

A continuación se explica más detalladamente cada experimento:

Experimento 1. Análisis en detalle con el clasificador.

Este primer experimento consiste en obtener unos primeros resultados con los que se puedan extraer conclusiones sobre qué subconjuntos de señales presentan un EER más bajo.

Se pasan por el clasificador los diferentes conjuntos de señales divididos en segmentos, denominados subconjuntos, y se obtienen los valores de EER para cada subconjunto. En concreto, se analizan los diferentes conjuntos de los 4 planos con los que trabajamos: planos 2D (XY, XZ, YZ) y plano 3D (plano XYZ). Este experimento se lleva a cabo usando unos valores predeterminados del clasificador. En el capítulo anterior se explica cada una de las variables del clasificador (el *kernel*, *gamma* y *sig2*). En este caso, los valores predeterminados son:

Capítulo 4. Metodología

- *Kernel*: RBF
- *Gamma* = 100
- *sig2*=10

A partir de los resultados que se obtengan se decide con qué conjuntos de señales se seguirá trabajando para ajustar el clasificador.

Experimento 2. Ajuste del clasificador y selección del *kernel*.

Una vez obtenidos los primeros resultados se ajustará el clasificador, como ya se ha comentado, usando como referencia el subconjunto que haya presentado el EER más bajo en el experimento anterior. Este ajuste se lleva a cabo dando valores a *gamma* y *sig2* mediante el método de prueba y error. Se deja fijo un valor y se varía el otro hasta obtener el resultado óptimo de éste. Luego, se fija el valor que antes variaba y se itera ahora el otro. Así se hará sucesivamente hasta obtener el EER más bajo posible.

Este ajuste se realiza, por un lado, para un *kernel RBF* y, por otro lado, para un *kernel* polinómico:

Experimento 2.1. Kernel RBF.

Se procede como se ha explicado. Primero se fija el valor de *gamma* y se varía el de *sig2*. Cuando se alcanza el valor de EER más bajo para ese valor de *gamma*, se fija el valor de *sig2* obtenido y se empieza a variar el valor de *gamma*, así sucesivamente. De esta forma se obtienen los valores de estos dos parámetros que ofrecen el mínimo EER posible para el conjunto estudiado.

Experimento 2.2. Kernel polinómico.

Se procede de igual manera que para el *kernel RBF*, con la diferencia de que ahora se trabaja con 3 valores: Uno para el parámetro *gamma* y dos para el parámetro *sig2*, ya para este *kernel* es del tipo $sig2=[a,b]$. En este caso se dejan fijo dos valores y se varía un valor.

Experimento 3. Estudio de los datos añadiendo nuevos parámetros estadísticos.

En este experimento se añaden otros parámetros estadísticos, a parte de la media, desviación típica y la correlación que ya se usan para caracterizar las señales. Estos parámetros que se añaden son la entropía de Shannon, la curtosis y la asimetría (*skewness*). La finalidad de este experimento es comprobar si añadiendo uno o varios de estos parámetros se mejoran los resultados obtenidos en los experimentos previos.

Hasta ahora, como se explica en el capítulo 3, de cada firma se extraen los diferentes conjuntos de señales con los que se trabaja y, a su vez, de estos conjuntos se obtienen subconjuntos, que no es más que dividir éstos en segmentos. Cada conjunto se divide de 2 a 30 segmentos, es decir, se obtienen 29 subconjuntos de cada conjunto. Ahora, en cada segmento se haya la media y desviación típica de cada señal y la correlación que hay entre las señales.

Por tanto, en este experimento, a la media, desviación típica y correlación, se añaden la entropía de Shannon, la curtosis y la asimetría o *skewness*.

Los subconjuntos que se seleccionan son los que mejores resultados presenten para diferentes números de muestras de entrenamiento. Evidentemente, los mejores resultados serán los que se obtienen cuando se simula con 5 muestras de entrenamiento. Por lo que también se seleccionarán algunos subconjuntos que mejor resultado presenten simulando con 1 muestra de entrenamiento.

Se añaden los mejores resultados cuando se simula con 1 muestra de entrenamiento porque interesa mejorar estos resultados de EER. Esto se debe a que sería realmente interesante reducir al máximo el EER usando 1 muestra porque significaría que solo con 1 firma en la base de datos bastaría para realizar una buena clasificación. Además, sería lógico pensar que, si con 1 muestra los resultados son buenos, a medida que se aumentan las muestras se obtienen mejores resultados.

Las simulaciones del experimento se realizan utilizando 5 y 1 muestra de entrenamiento. No se diseña el uso de 2, 3 ó 4 muestras porque realmente no

aportan información para este experimento y, además, se reducen tiempos de simulación.

Experimento 4. Estudio de la influencia de la tercera dimensión con respecto al sistema plano.

Este experimento se diseña para saber cuánta información aporta la tercera dimensión a la hora de clasificar las firmas. Para ello se comparan los resultados de los conjuntos que mejor desempeño ofrecen del plano 3D (XYZ) con los de los diferentes planos 2D (XY, XZ, YZ). Esta comparación se realiza para las diferentes muestras de entrenamiento, de 5 muestras a 1 muestra. De esta forma se puede obtener las conclusiones sobre si la tercera dimensión aporta información y en qué medida.

Experimento 5. Fusión de los conjuntos.

Una vez llegado a este punto, se realiza un último experimento para intentar reducir el EER y seleccionar definitivamente los subconjuntos a usar para las simulaciones finales. Para ello se seleccionan los subconjuntos que mejores resultados presentan en los experimentos previos y se fusionan, obteniendo así el subconjunto de señales final.

Estas fusiones se realizan por el método de prueba y error. Se seleccionan los subconjuntos con un EER más bajo y se fusionan de dos en dos, de tres en tres, y así sucesivamente hasta que se encuentre la fusión con el EER más bajo.

Esto se realiza, por un lado, para el *kernel RBF*, y por otro lado para el *kernel* polinómico. Por lo tanto, para cada *kernel* se puede obtener un subconjunto final diferente.

Es importante indicar el hecho de que un subconjunto tenga el valor de EER más bajo cuando se simula con 5 muestras de entrenamiento no asegura que, cuando se baje el número de muestras de entrenamiento, este subconjunto siga teniendo el mejor valor de EER. Por ejemplo, si se simula con 5 muestras de entrenamiento y se obtiene que el subconjunto de 4 segmentos conformado por todas las señales del plano XY

es el que presenta el valor de EER más bajo con respecto al resto de subconjuntos del mismo conjunto, no significa que cuando se simule con 1 muestra de entrenamiento este siga siendo el subconjunto con el valor de EER más bajo.

Por lo tanto, es necesario decidir si se seleccionan los subconjuntos que mejores resultados presenten para 5, 4, 3, 2 ó 1 muestra de entrenamiento.

Como se comenta en el experimento 3, se priorizan los resultados de los experimentos con 1 muestra de entrenamiento.

Experimento 6. Simulación del sistema con una base de datos ampliada a 70 usuarios.

Se realiza una simulación por cada uno de los dos tipos de *kernel* con los subconjuntos de señales finales que se hayan seleccionado, usando la base de datos ampliada de 70 usuarios antes de realizar la simulación con la base de datos final de 100 usuarios. De esta forma se obtiene información de cómo pueden variar los resultados a medida que se aumenta el número de usuarios.

En los experimentos anteriores, para simular se seleccionan, de las 10 firmas por usuario, las primeras como muestras de entrenamiento. Por ejemplo, si son 4 muestras las que se usan para entrenar el clasificador, se cogen las 4 primeras firmas del usuario. Sin embargo, en estos dos últimos experimentos que se realizan, se seleccionan de manera aleatoria las firmas que se van a usar como muestras de entrenamiento, es decir, ya no serían las 4 primeras firmas, sino 4 cualesquiera de las 10 firmas del usuario. Además, cada simulación se repite 100 veces, con diferentes firmas para muestras de entrenamiento cada vez, para obtener unas conclusiones fiables.

Se realizan simulaciones para 5, 4, 3, 2 y 1 muestra de entrenamiento, tanto para el *kernel* polinómico como para el *kernel RBF*.

Experimento 7. Simulación con la base de datos final.

Finalmente, procediendo como en el experimento anterior, se realiza la simulación con la base de datos final de 100 usuarios.

Capítulo 4. Metodología

Realizados los 7 experimentos de los que consta este proyecto, se podrán obtener las conclusiones definitivas.

Capítulo 5. Resultados

El estudio realizado en el Trabajo de Fin de Máster del que parte este proyecto consiste en agrupar las diferentes señales, obtenidas a través del sensor *Leap Motion* y guardadas en la base de datos, y dividir las en 5, 10 ó 15 segmentos. En este proyecto, como bien se ha explicado en los capítulos anteriores, se realiza un análisis más exhaustivo, dividiendo, para cada agrupación de señales, las mismas en grupos de 2 a 30 segmentos, obteniendo así lo que en este proyecto se denomina un subconjunto. Por tanto, el conjunto formado por las señales X e Y de la posición del dedo índice y, divididas cada una de ellas en 5 segmentos, se llamará “subconjunto de 5 segmentos de las señales X e Y del dedo índice”.

En este capítulo se muestran los resultados de los experimentos previamente explicados en el capítulo 4. Estos resultados son representados en gráficas principalmente, haciendo especial énfasis en los valores más relevantes. En el Anexo A se muestran todos los resultados, en tablas, de manera más detallada.

Como se comentó en el capítulo anterior, los cinco primeros experimentos se realizan usando la base de datos inicial de 39 usuarios. En el experimento 6 se utiliza la base de datos correspondiente a la primera ampliación, es decir, la base de datos ampliada a 70 firmantes. Por último, el experimento 7 se simula con la base de datos final, que consta de 100 usuarios.

Experimento 1. Análisis en detalle con el clasificador.

Como se indica previamente, se agrupan los conjuntos de señales en función al plano al que refieren, es decir, se agrupan en función de si caracterizan a alguno de los 4 planos siguientes: planos 2D (XY, XZ, YZ) o plano 3D (XYZ), analizando cada uno por separado. Estas simulaciones se hacen para 5 muestras de entrenamiento.

Una vez realizada la simulación y observando los resultados obtenidos, se concluye que los mejores resultados corresponden a los conjuntos que están formados por las

agrupaciones de todas las señales para cada plano. Es decir, para el plano XY se alcanzan mejores resultados cuando se agrupan todas las señales del plano [Todas XY] (posición del dedo índice del plano XY, posición de la palma de la mano del plano XY, etc.) que si se agrupan por separado (solo posición del dedo índice del plano XY, o solo posición de la palma de la mano del plano XY, etc.).

No se mostrarán todos los resultados porque consistiría en añadir muchas tablas que realmente no aportan demasiada información en este primer experimento, ya que es una primera toma de contacto con el clasificador y éste no está ajustado. Hay que tener en cuenta que para cada plano hay 7 conjuntos. A su vez, cada conjunto está dividido de 2 a 30 segmentos, lo que hace un total de 29 subconjuntos por cada conjunto. Como se tienen 4 planos diferentes, se cuenta finalmente con 812 resultados.

Debido al gran número de resultados y a que lo realmente importante de este experimento es destacar los conjuntos que mejores resultados presentan, se muestran en la tabla 5.1 los resultados más destacados.

Conjunto de señales	segmentos	EER(%)
Todas XY [2D]	5	0,5769
Todas XY [2D]	4	0,6312
Todas XZ [2D]	5	0,9430
Todas XZ [2D]	4	1,0260
Todas XYZ [3D]	2	1,0507
Todas YZ [2D]	3	1,5380
Todas YZ [2D]	4	1,5380

Tabla 5.1. Mejores resultados con el clasificador no ajustado, con 5 muestras usando *kernel RBF*.

Como se observa en la tabla 5.1, el subconjunto que mejor resultado presenta es el de **5 segmentos de todas las señales del plano XY** con un EER de **0,5769%**. Por lo tanto, se

procede a realizar el ajuste de los parámetros del clasificador usando este subconjunto de señales.

Observando estos primeros resultados, se podría continuar, a partir de aquí, realizando las siguientes simulaciones del resto de experimentos usando estos conjuntos conformados por todas las señales referentes a los planos de estudio. Aun así, aunque los mejores resultados se den con estos conjuntos de señales, **no se debe descartar ningún otro conjunto hasta que no se tenga el clasificador ajustado.**

Experimento 2. Ajuste del clasificador y selección del *kernel*.

El siguiente paso es ajustar el clasificador. En este punto el proyecto sigue dos líneas paralelas. Por un lado, se ajusta el clasificador para un *kernel* RBF y, por otro lado, para un *kernel* polinómico.

Para el ajuste se irán variando los parámetros *gamma* y *sig2* del clasificador. Para acortar el tiempo de simulación, se selecciona el subconjunto que mejor resultado ofrece, de acuerdo al experimento 1, y se ajustan estos parámetros hasta obtener el menor resultado de EER.

En este caso, el subconjunto con mejor resultado de EER es el **subconjunto de 5 segmentos conformado por todas las señales del plano XY**, con un EER de **0,5769%**, como se observa en la tabla 5.1.

A continuación, se muestran los resultados del ajuste para un *kernel* RBF y para un *kernel* polinómico usando el subconjunto mencionado. Cabe mencionar que se usa este subconjunto, que es el que mejores resultados presenta por ahora, para ajustar los parámetros del clasificador. Una vez ajustado el clasificador, se volverá a hacer uso del resto de conjuntos de señales para la posterior selección o rechazo de los diferentes de los mismos.

Experimento 2.1. *Kernel* polinómico

Después de probar con distintos valores, el ajuste óptimo de los parámetros para el *kernel* polinómico es:

Capítulo 5. Resultados

- $\gamma=300$
- $\text{sig}2=[7;2]$

Con estos valores, se consigue un **EER de 0,0405%** para la agrupación de señales seleccionada para el ajuste. Es una mejora considerable, dado que si se compara con el resultado inicial de 0,5769% se puede ver que se disminuye el EER en casi su totalidad.

Para comprobar exactamente cuánto ha disminuido el EER, tenemos:

$$\frac{0,5769}{0,0405} = 14,2\hat{4}$$

por lo tanto, el EER se reduce, siendo ahora 14 veces menor.

Ahora se simula de nuevo para todos los conjuntos de señales de cada plano, para 5 y 1 muestra de entrenamiento, ahorrando así tiempo de simulación, dado que en este punto los resultados para 2, 3 ó 4 muestras no aportan demasiada información.

Para no cargar este apartado con demasiados números, los resultados obtenidos se muestran en gráficas. Además, solo se mostrarán los resultados más importantes. Al igual que en el experimento 1, resulta que los conjuntos que mejores resultados muestran son los conjuntos de todas las señales de cada plano, como se puede observar en las figuras [5.1 – 5.8].

En cada gráfica se representa un conjunto de señales, que en este caso corresponden a cada conjunto de señales de cada plano, es decir, todas las señales del plano XY, todas las señales del plano XZ, todas las señales del plano YZ y todas las señales del plano XYZ. En el eje de abscisas se tienen los segmentos en los que se divide el conjunto y en el eje de ordenadas se muestran los valores obtenidos de EER (%). Se representan en total 8 gráficas, 4 para las simulaciones con 5 muestras de entrenamiento y otras 4 para aquellas realizadas con 1 muestra de entrenamiento.

Los resultados para 5 muestras de entrenamiento se representan en las figuras [5.1-5.4].

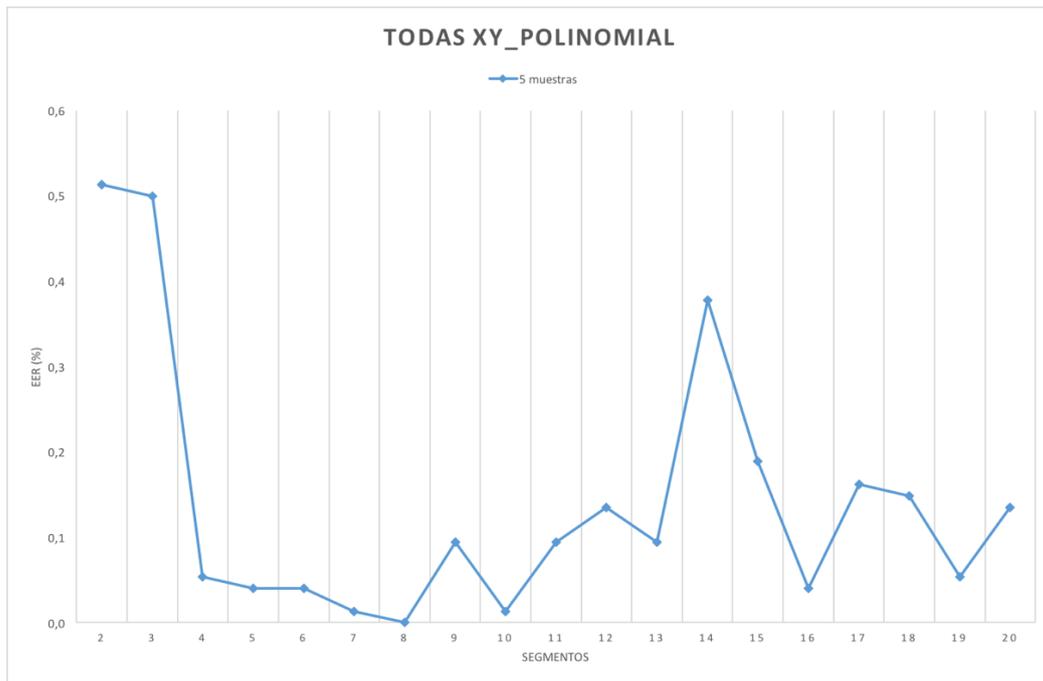


Figura 5.1. Kernel polinómico para todas las señales XY con 5 muestras

Como se observa en la figura 5.1, para el subconjunto de 8 segmentos de todas las señales del plano XY se consigue un valor de EER del **0%**.

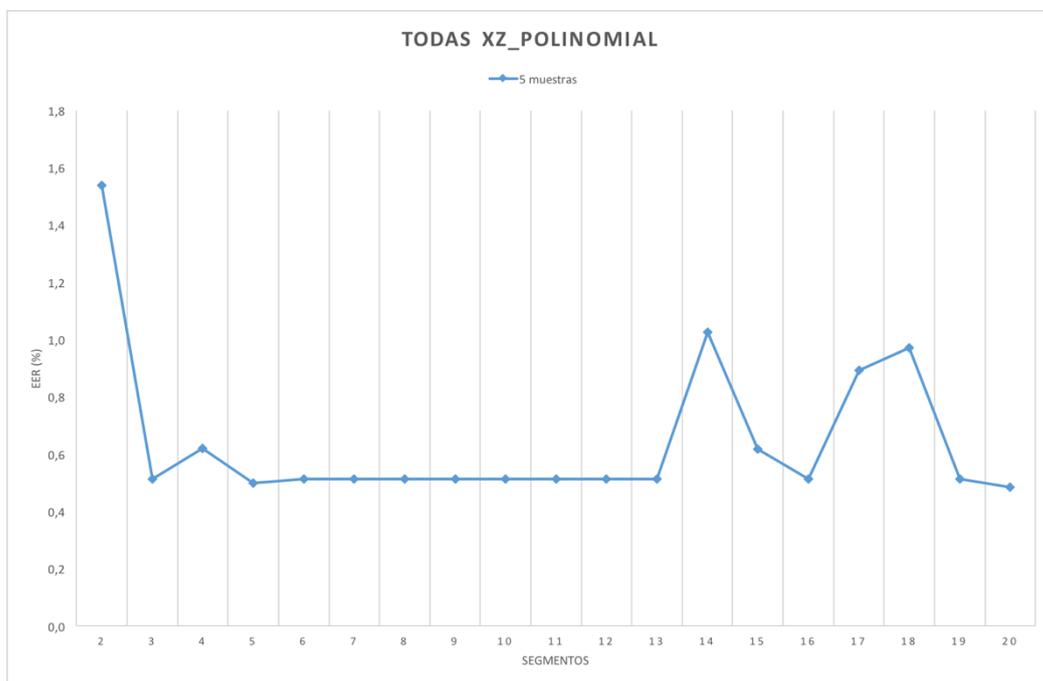


Figura 5.2. Kernel polinómico para todas las señales XZ con 5 muestras

Para el subconjunto de 20 segmentos de todas las señales del plano XZ se consigue un valor de EER del **0.48583%**.

Capítulo 5. Resultados

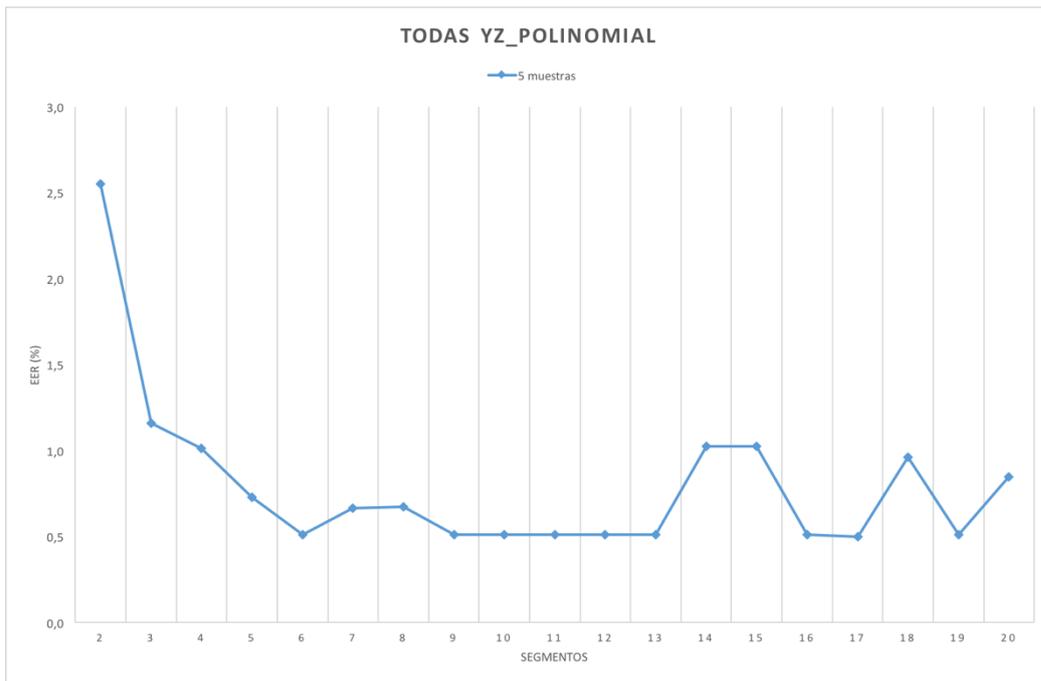


Figura 5.3. *Kernel* polinómico para todas las señales YZ con 5 muestras

Para el subconjunto de 17 segmentos de todas las señales del plano YZ se consigue un valor de EER del **0.49933%**.

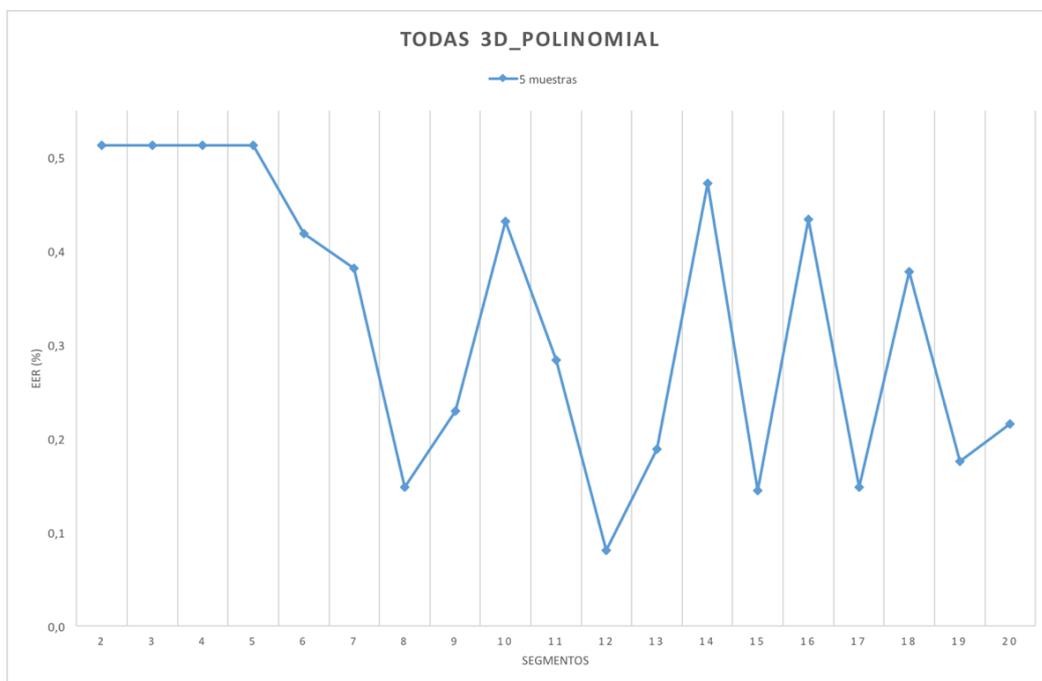


Figura 5.4. *Kernel* polinómico todas las señales XYZ con 5 muestras

Para el subconjunto de 12 segmentos de todas las señales del plano XYZ se consigue un valor de EER del **0.0810%**.

Cabe destacar que, el valor más alto de EER ($\approx 0,5\%$) y, por tanto el peor, para el conjunto de todas las señales del plano XY (figura 5.1), coincide prácticamente con el valor más bajo y, por tanto el mejor, para los conjuntos de todas las señales de los planos XZ e YZ (figura 5.2 y figura 5.3 respectivamente).

Para 1 muestra se obtienen los resultados mostrados en las figuras [5.5-5.8].

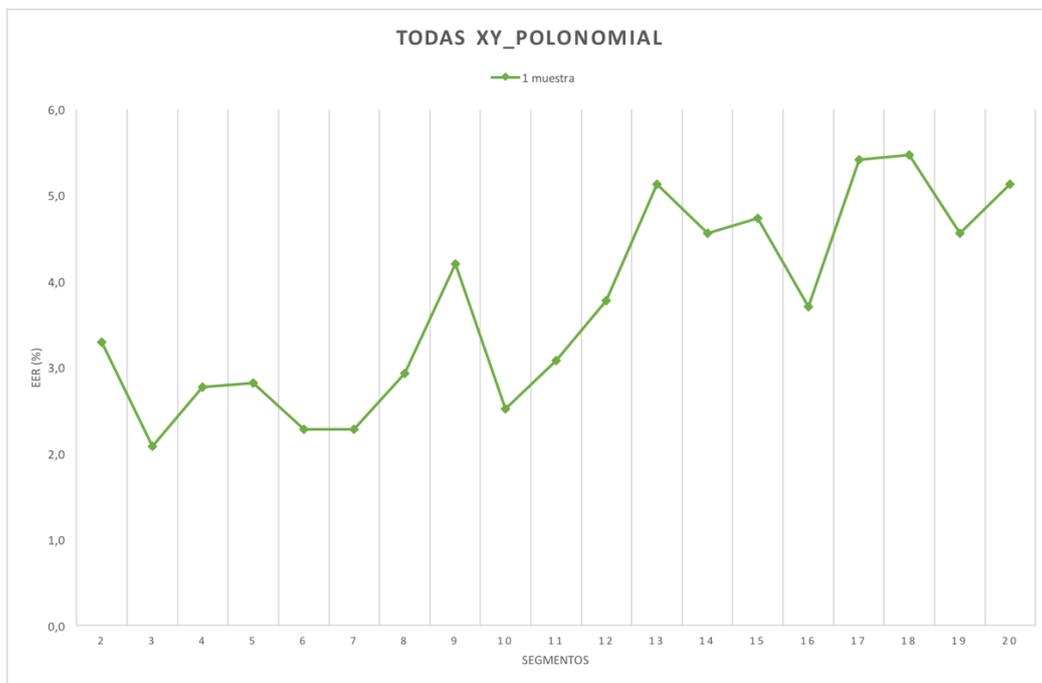


Figura 5.5. *Kernel* polinómico para todas las señales XY con 1 muestra

Para el subconjunto de 3 segmentos de todas las señales del plano XY se consigue un valor de EER del **2.0869%**, como se puede ver en la figura 5.5.

Capítulo 5. Resultados

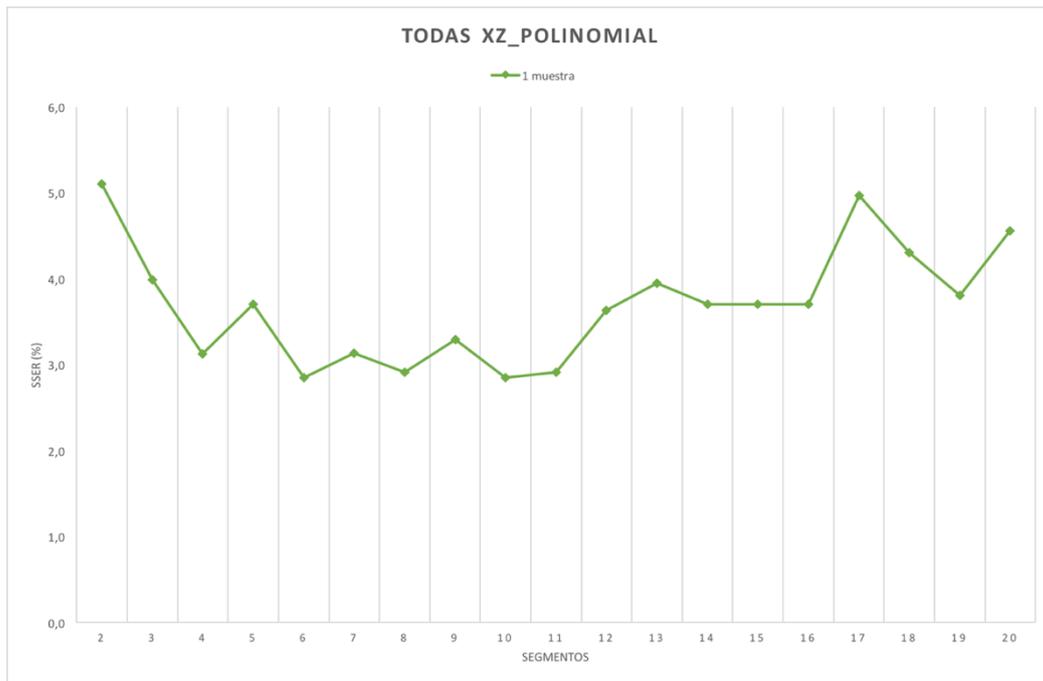


Figura 5.6. *Kernel* polinómico para todas las señales XZ con 1 muestra

Para los subconjuntos de 6 y de 10 segmentos de todas las señales del plano XZ se consigue un valor de EER del **2.8490%**.

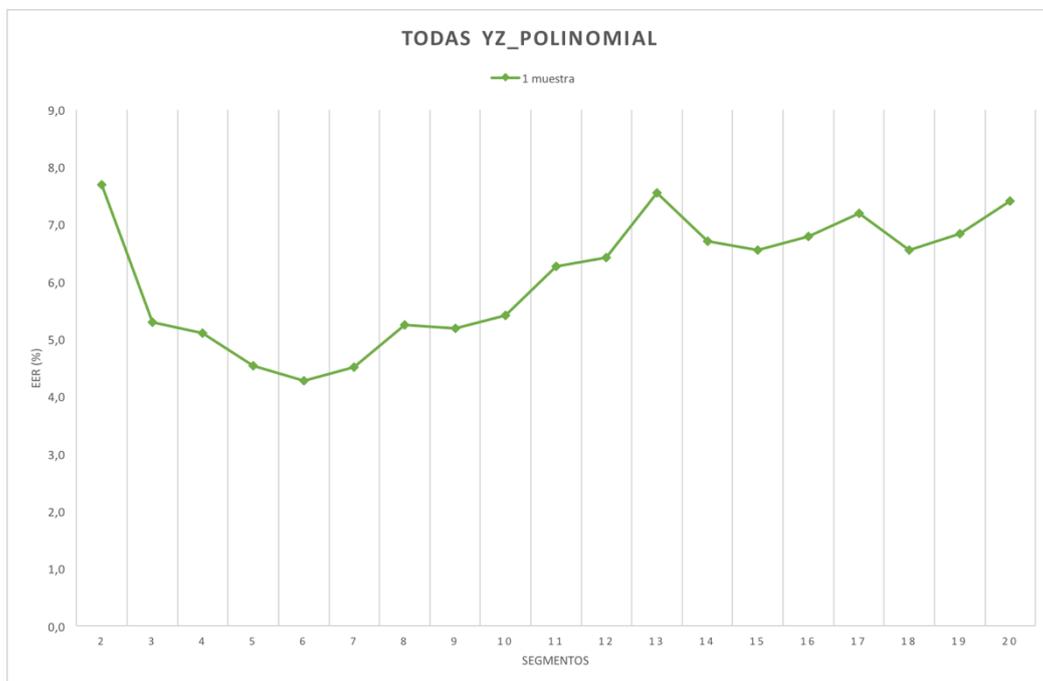


Figura 5.7. *Kernel* polinómico para todas las señales YZ con 1 muestra

Para el subconjunto de 6 segmentos de todas las señales del plano YZ se consigue un valor de EER del **4.2735%**.

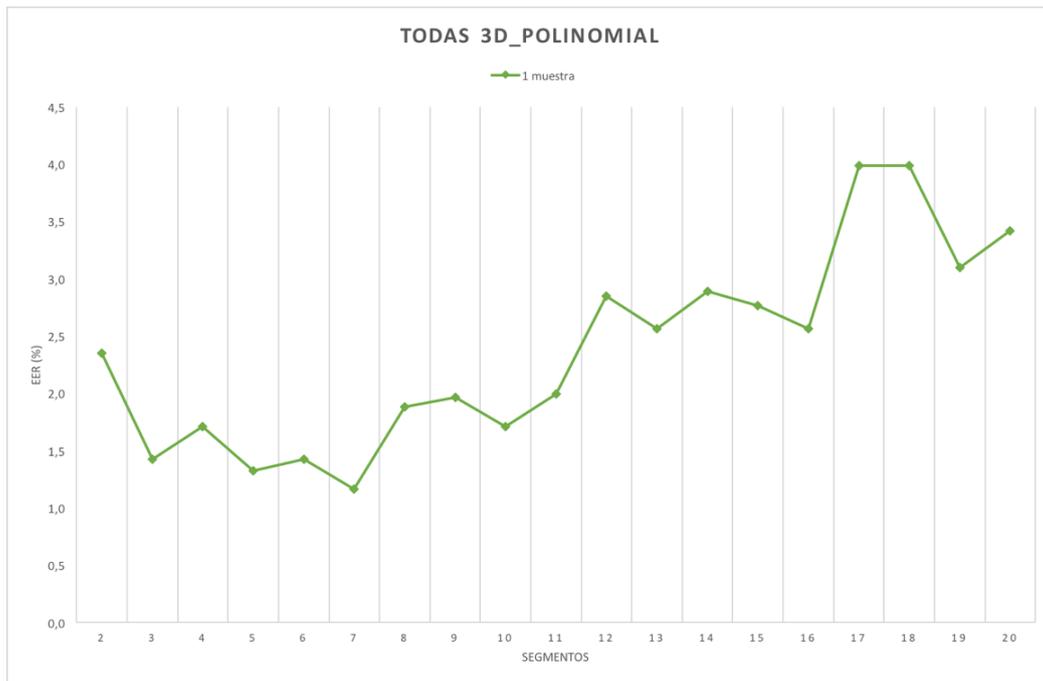


Figura 5.8. Kernel polinómico para todas las señales XYZ con 1 muestra

Para el subconjunto de 7 segmentos de todas las señales del plano XYZ se consigue un valor de EER del **1.1674%**.

Se pueden ver los resultados de una manera más detallada en las tablas del Anexo A.

Si se comparan los conjuntos, se tiene que, para 5 muestras de entrenamiento el plano que mejores resultados ofrece es el plano XY. En cambio, al simular con 1 muestra de entrenamiento, el plano con un EER más bajo es el plano 3D (plano XYZ), es decir, el conjunto de todas las señales.

Si se analizan, dentro de cada conjunto, los diferentes subconjuntos se puede decir que los mejores resultados se obtienen, en general, para los subconjuntos de entre 4 y 10 segmentos.

Otro punto importante es destacar como el plano YZ da los peores resultados de EER. Para el caso de 5 muestras, comparando las gráficas obtenidas, se aprecia menos, en cambio, para 1 muestra es mucho más considerable la diferencia.

La conclusión que se obtiene al comparar los resultados de este experimento es que los planos donde aparece la coordenada X dan mejores resultados (el plano YZ da los peores

Capítulo 5. Resultados

resultados con diferencia). Además, los resultados del plano XZ, en comparación con los de los planos XY y XYZ, no se consideran buenos.

Como consecuencia, a partir de este experimento **se descartan los planos XZ e YZ**. A partir del próximo experimento **se trabaja**, entonces, con el **plano XY y con el plano XYZ**.

Experimento 2.2. *Kernel* RBF

El ajuste óptimo para los parámetros del clasificador usando el *kernel* RBF es:

- $\gamma=400$
- $\sigma^2=150$

Con estos valores, se logra tener un **EER de 0%** para la agrupación de señales seleccionada en el experimento 1 para el ajuste.

Ahora la simulación se realiza, de acuerdo a la conclusión del experimento 2.1, sobre todas las señales del plano XY y sobre todas las señales del plano tridimensional (XYZ), obteniendo los siguientes resultados para 5 muestras:

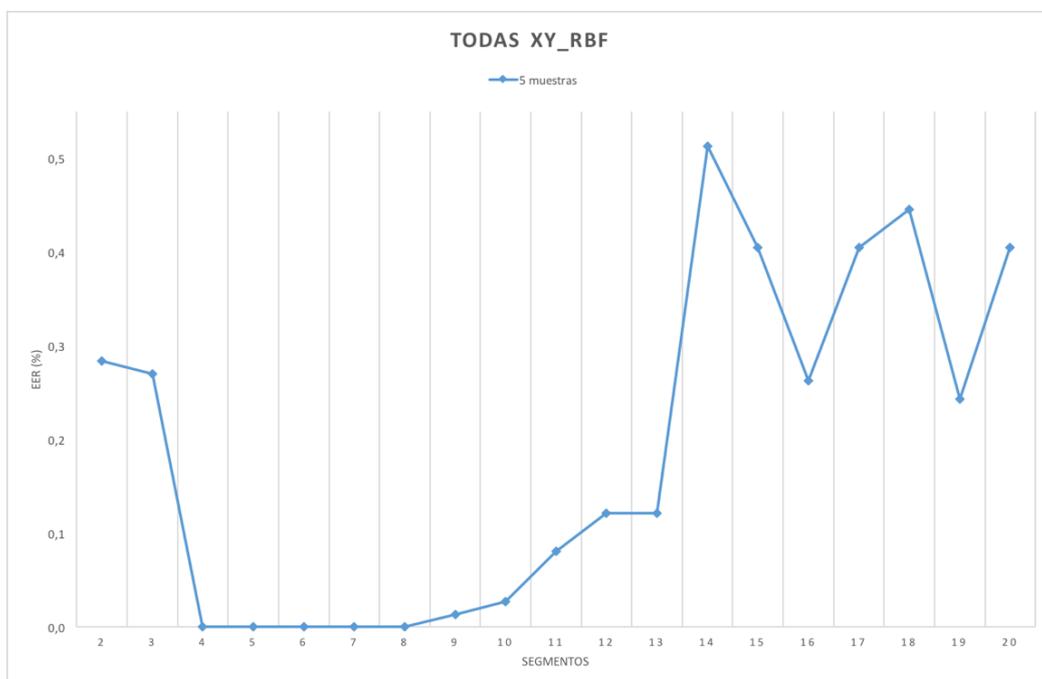


Figura 5.9. *Kernel* RBF para todas las señales XY con 5 muestras

Para los subconjuntos de 4, 5, 6, 7 y 8 segmentos de todas las señales del plano XY se consigue un valor de EER del **0%**.

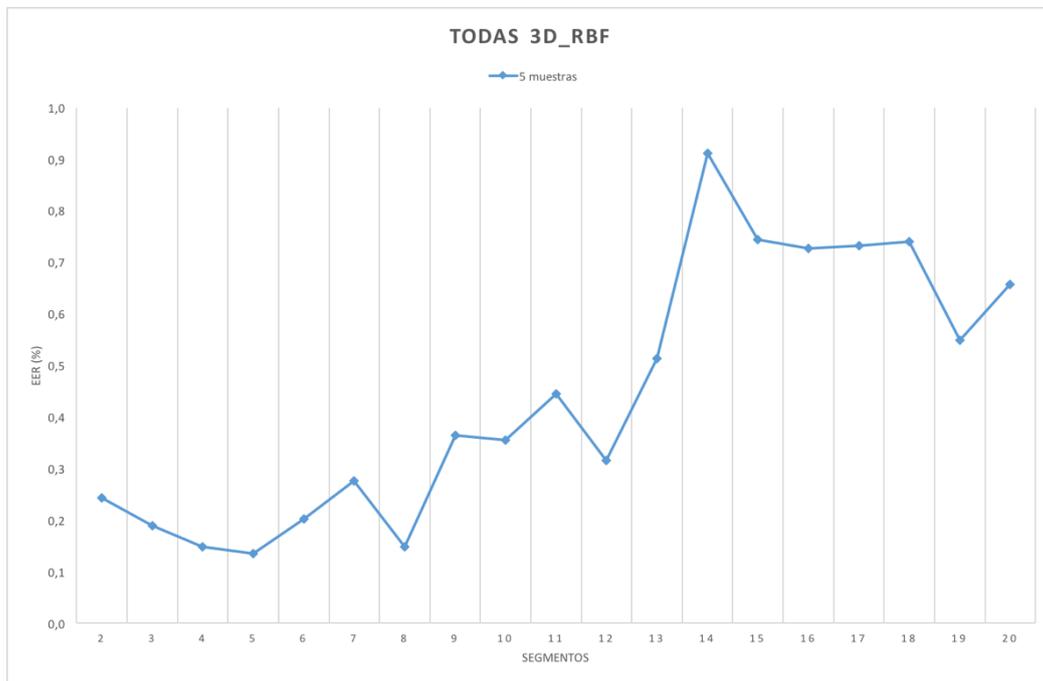


Figura 5.10. *Kernel* RBF para todas las señales XYZ con 5 muestras

Para el subconjunto de 5 segmentos de todas las señales del plano XYZ se consigue un valor de EER del **0.1350%**.

Y para 1 muestra se obtiene lo siguiente:

Capítulo 5. Resultados

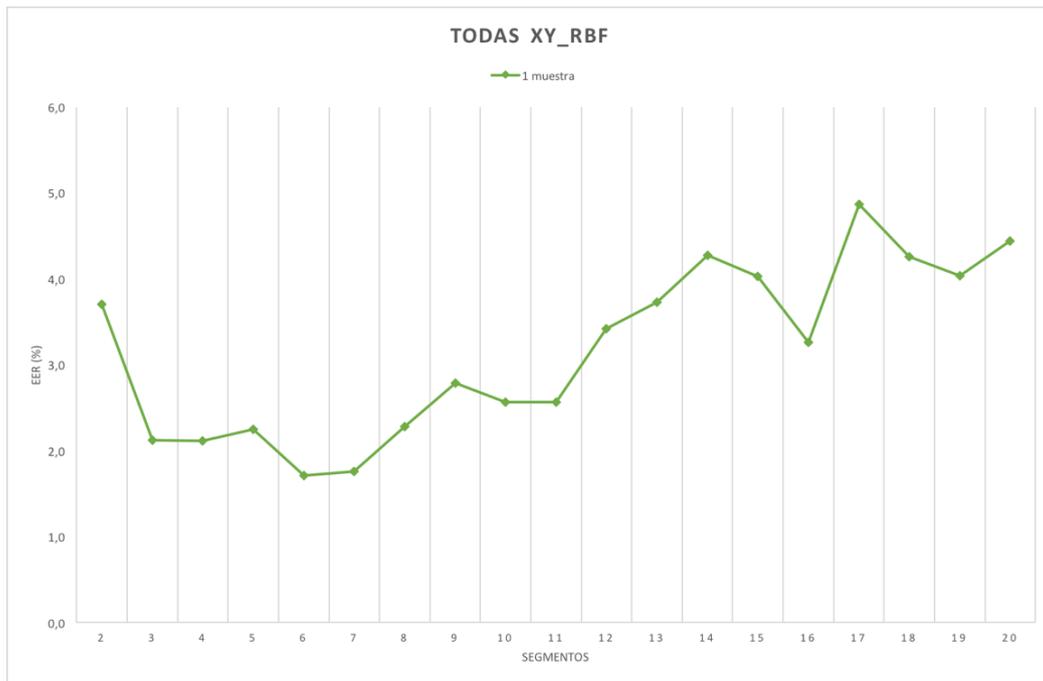


Figura 5.11. *Kernel* RBF para todas las señales XY con 1 muestra

Para el subconjunto de 6 segmentos de todas las señales del plano XY se consigue un valor de EER del **1.7094%**.

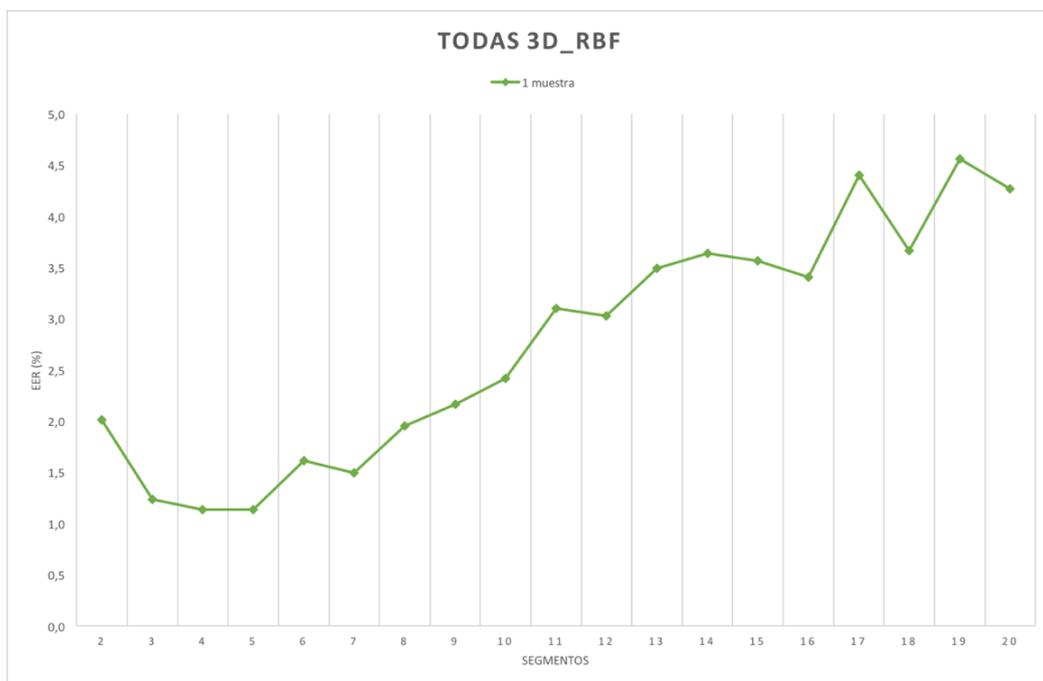


Figura 5.12. *Kernel* RBF para todas las señales XYZ con 1 muestra

Para los subconjuntos de 4 y 5 segmentos de todas las señales del plano XYZ se consigue un valor de EER del **1.1396%**.

Se pueden ver los resultados de una manera más detallada en tablas en el Anexo A.

Ocurre lo mismo que en el caso del *kernel* RBF, con 5 muestras de entrenamiento el EER es más bajo para plano XY pero, con 1 muestra de entrenamiento, es más bajo para el plano XYZ.

Con estos resultados podemos decir que **la tercera dimensión** (la coordenada Z que corresponde con la profundidad) **sí aporta información cuando hay pocas muestras de entrenamiento.**

Experimento 3. Estudio de los datos añadiendo nuevos parámetros estadísticos.

Para realizar este experimento se usan los subconjuntos que mejores resultados ofrecen de acuerdo al experimento 2. Se seleccionan los 7 subconjuntos que mejor resultado presentan, que coinciden, como es de esperar, con resultados obtenidos cuando se simula con 5 muestras de entrenamiento.

Como ya se ha comentado, es interesante obtener un EER muy bajo usando 1 muestra de entrenamiento, por lo que también se seleccionan los dos mejores resultados obtenidos al usar 1 muestra.

En primer lugar, se realiza el experimento usando el *kernel* polinómico.

Los subconjuntos que mejor resultado se obtienen para 5 muestras son, en este orden:

- Subconjunto de **8 segmentos** de **todas** las señales **XY**.
- Subconjunto de **12 segmentos** de **todas** las señales **XYZ**.
- Subconjunto de **7 segmentos** de las señales de **posición XYZ del índice**.
- Subconjunto de **11 segmentos** de las señales de **posición XY del índice**.
- Subconjunto de **22 segmentos** de las señales de **posición XYZ de la palma de la mano**.
- Subconjunto de **6 segmentos** de las señales de la **velocidad XYZ del índice**.
- Subconjunto de **22 segmentos** de las señales de **posición XY de la palma de la mano**.

Y los mejores resultados para 1 muestra:

- Subconjunto de **7 segmentos** de **todas** las señales **XYZ**.
- Subconjunto de **3 segmentos** de **todas** las señales **XY**.

Se realizan simulaciones usando estos subconjuntos para 5 muestras y 1 muestra de entrenamiento.

A cada uno de estos 9 subconjuntos se le añaden, como se ha explicado en el capítulo anterior, estos tres parámetros estadísticos: entropía de Shannon, skewness (asimetría) y curtosis.

En las figura 5.13 y figura 5.14 se representa la variación del EER con respecto al valor inicial. Los valores están normalizados, es decir, se toma el valor de EER inicial de cada subconjunto como 0. En el eje de abscisas se indican los parámetros que se añaden (curtosis, skewness, curtosis + skewness, etc.).

Por ejemplo, si un subconjunto presenta un EER inicial de 0,21% y, al añadir la entropía se obtiene un valor de 0,16%, entonces se indica que ha mejorado su EER en un 0,05% y en la gráfica este valor se representa en -0,05%. En cambio, si se añade la curtosis y el valor de EER ahora es de 0,30%, el valor de EER ha empeorado un 0,09% con respecto al valor inicial y, entonces, se representa en la gráfica en 0,09%.

En definitiva, los valores por debajo del umbral (línea roja) son valores de EER mejorados, mientras que los valores que están por encima del umbral son valores más altos de EER y, por tanto, peores. Lo que interesa es que hayan más valores por debajo del umbral.

Por otro lado, los valores del eje de abscisas en este orden corresponden a añadir:

- Skewness.
- Entropía de Shannon.
- Curtosis.
- Skewness y entropía de Shannon.
- Skewness y curtosis.

- Curtosis y entropía de Shannon
- Skewness, curtosis y entropía de Shannon (Todas).

En la gráfica de la figura 5.13 se representan estos subconjuntos estudiados para 5 muestras de entrenamiento, mientras que en la gráfica de la figura 5.14 se representan para 1 muestra de entrenamiento.

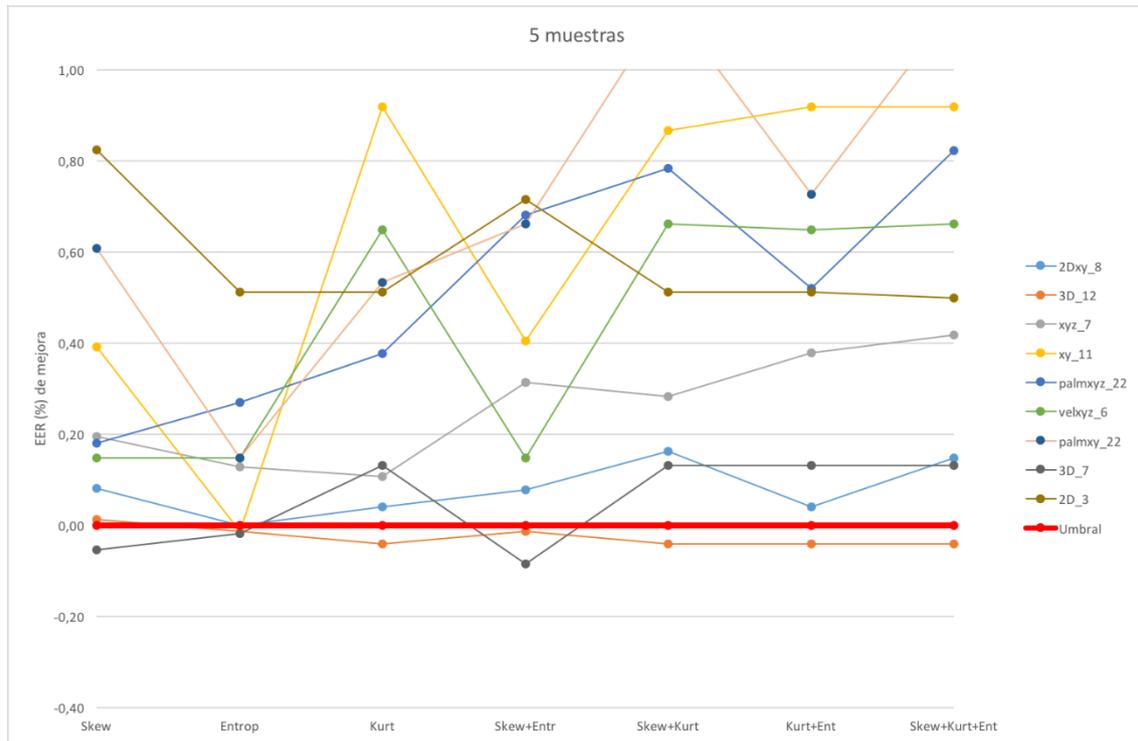


Figura 5.13. Variación del EER al añadir nuevos parámetros estadísticos, usando 5 muestras.

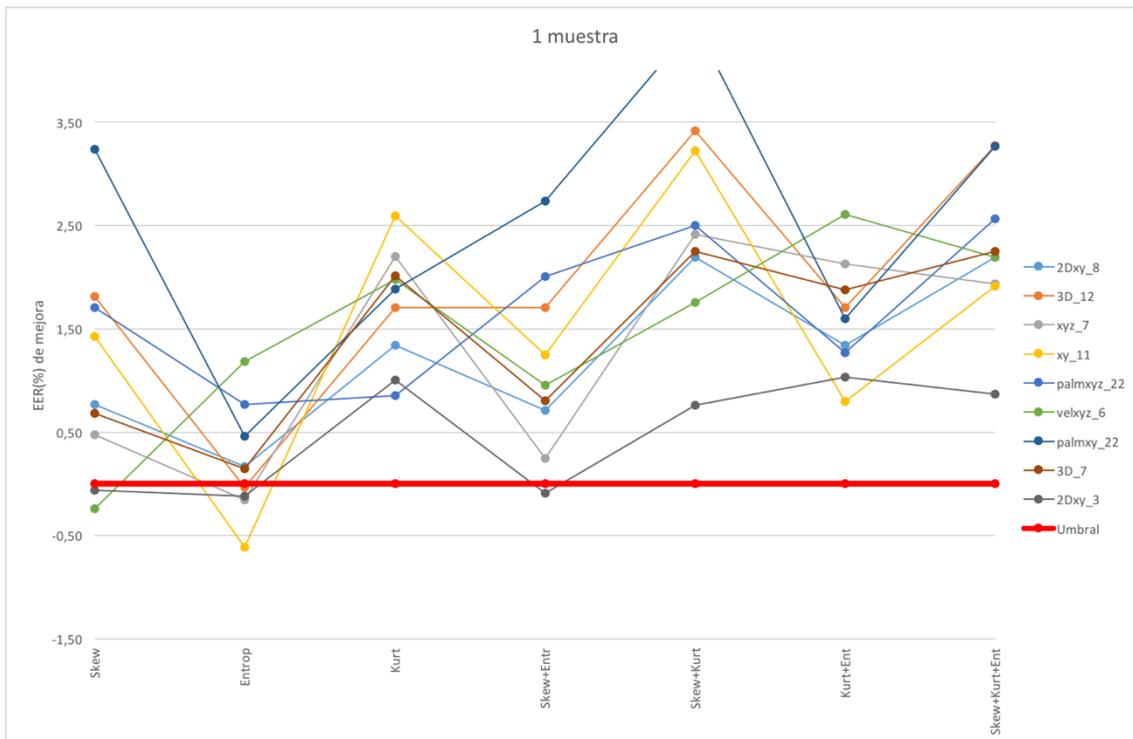


Figura 5.14. Variación del EER al añadir nuevos parámetros estadísticos usando 1 muestra.

Se pueden ver los resultados más detallados en tablas en el Anexo A.

Como se puede observar, si se cuentan los 18 casos (9 para 5 muestras y 9 para 1 muestra), la curtosis mejora los resultados en 1 de 18 casos (5,56 %); el skewness lo hace en 3 de 18 casos (16,67 %); y la entropía de Shannon mejor los resultados en 7 de 18 casos (38,89 %).

Por lo general, no se mejoran los resultados. Por lo tanto, **se descarta añadir cualquiera de estos parámetros estadísticos**. Así mismo, también se descarta repetir el experimento usando el *kernel* RBF.

Experimento 4. Estudio de la influencia de la tercera dimensión con respecto al sistema plano.

Como se indica en el experimento 2, se observa que para 5 muestras de entrenamiento el plano XY aporta mayor información que el plano XYZ, es decir, la tercera dimensión no solo no añade información, sino que además empeora los resultados. Sin embargo, los resultados para **1 muestra** de entrenamiento, dicen todo lo contrario, por lo que en este caso la tercera dimensión **sí que aporta información útil**.

Este experimento consiste en determinar a partir de cuántas muestras es importante añadir la tercera dimensión, es decir, cuándo la tercera dimensión empieza a aportar información discriminante.

Para ello, se generan varias gráficas, 5 para el *kernel* RBF y 5 para el *kernel* polinómico, una por cada conjunto de muestras. En cada gráfica se representan los resultados de EER de cada subconjunto del plano XY (azul) y del plano XYZ (rojo).

Para el *kernel* RBF se generan las siguientes figuras:

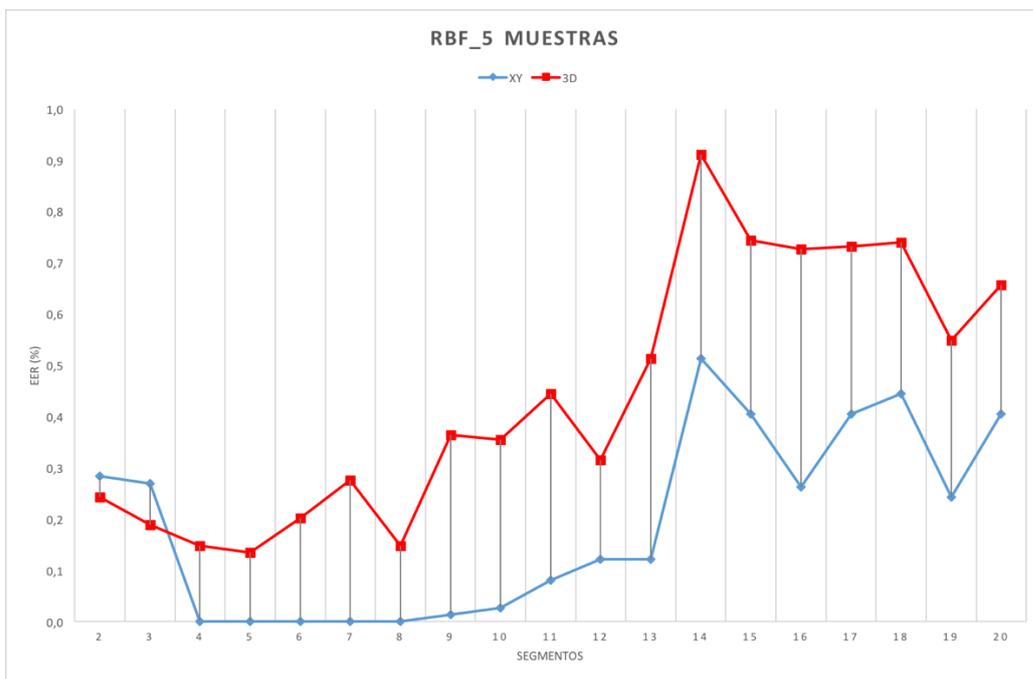


Figura 5.15. Comparación de los planos XY y XYZ, usando *kernel* RBF y 5 muestras de entrenamiento.

Capítulo 5. Resultados

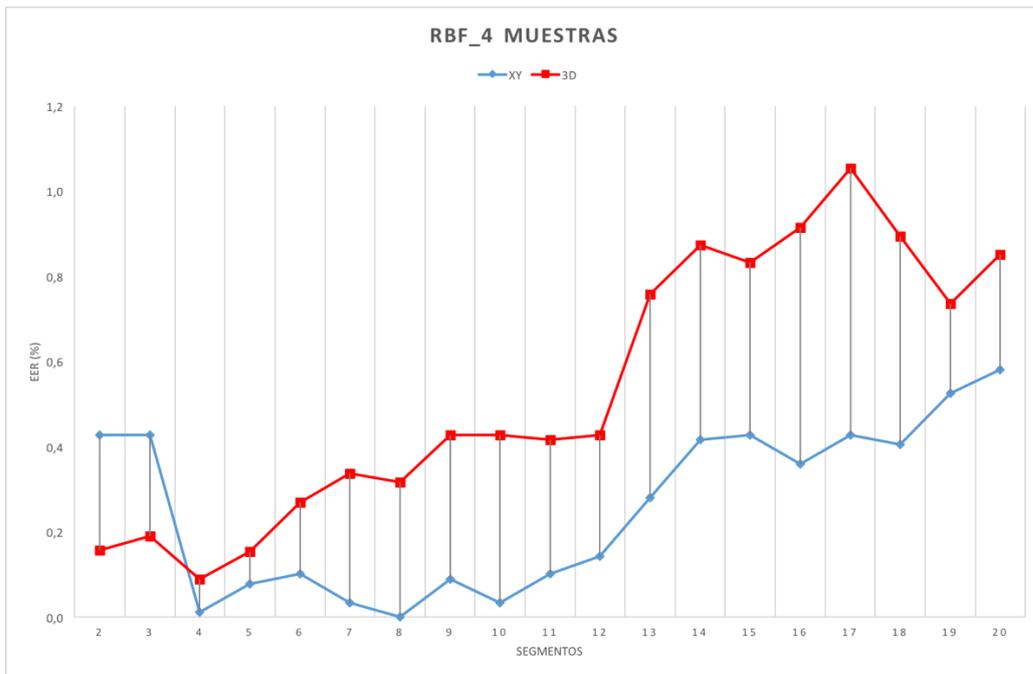


Figura 5.16. Comparación de los planos XY y XYZ, usando *kernel* RBF y 4 muestras de entrenamiento.



Figura 5.17. Comparación de los planos XY y XYZ, usando *kernel* RBF y 3 muestras de entrenamiento.

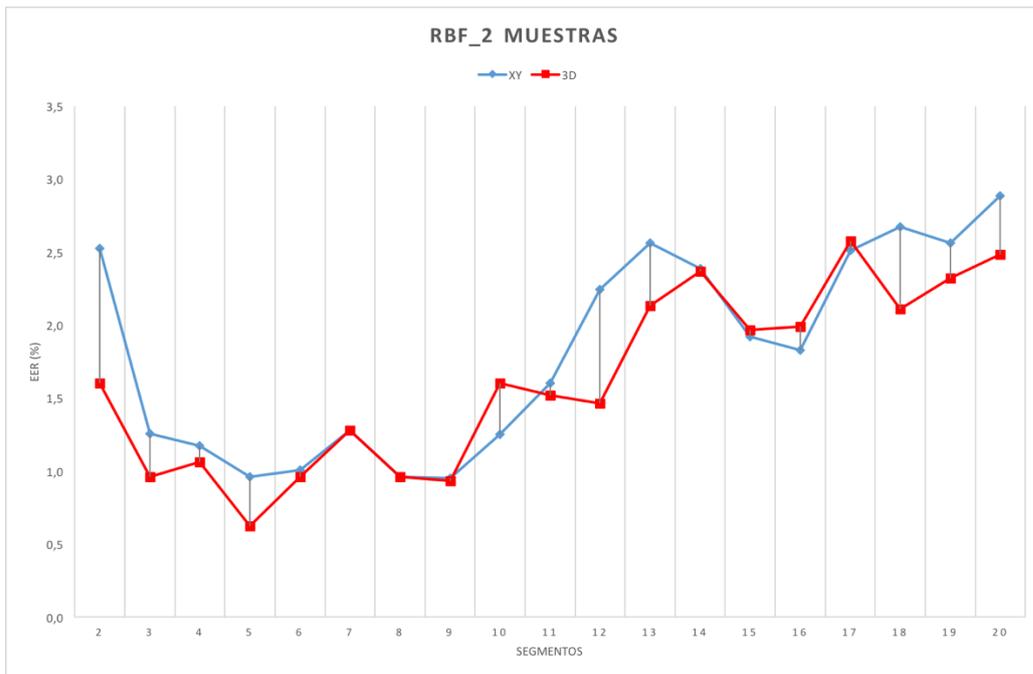


Figura 5.18. Comparación de los planos XY y XYZ, usando *kernel* RBF y 2 muestras de entrenamiento.



Figura 5.19. Comparación de los planos XY y XYZ, usando *kernel* RBF y 1 muestra de entrenamiento.

Como se indica previamente, para 5 muestras, claramente, el añadir la tercera dimensión se empeoran los resultados. Con 4 muestras se observa que las diferencias entre los resultados son muy parecidos al caso de 5 muestras. Con 3 muestras se puede decir que son más parecidos los resultados, pero parece que todavía no hay una mejora clara de los mismos. Con 2 muestras ya se puede decir que en muchos subconjuntos hay

Capítulo 5. Resultados

mejora de resultados. Y para 1 muestra, como se comenta anteriormente, se obtiene una mejoría de resultados al añadir la tercera dimensión en la mayoría de subconjuntos.

A continuación, se muestran las gráficas para el **kernel polinómico**:

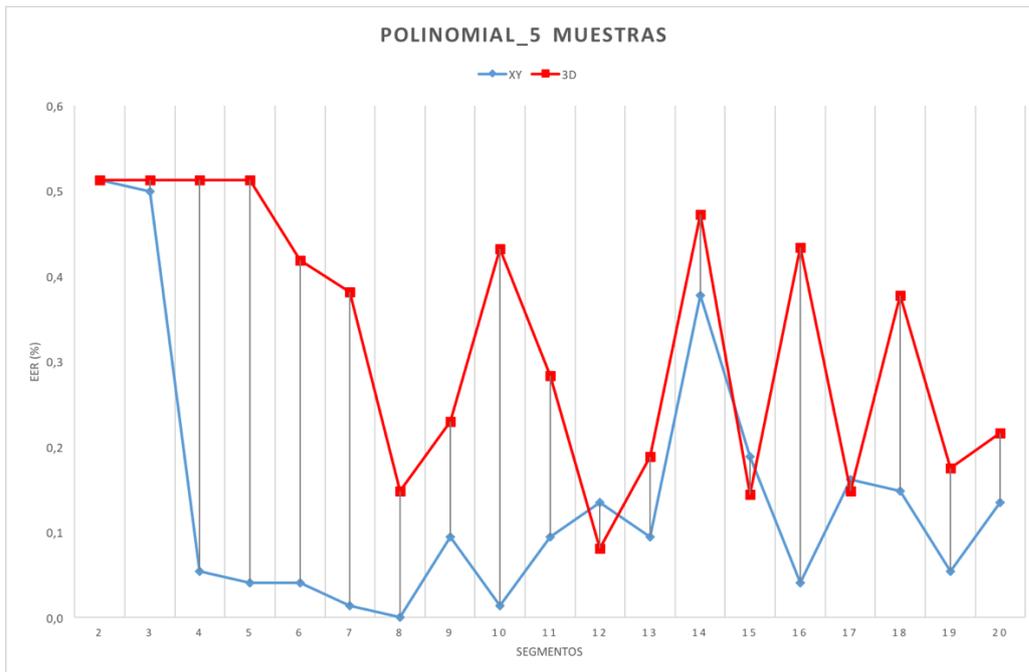


Figura 5.20. Comparación de los planos XY y XYZ, usando *kernel* polinómico y 5 muestras de entrenamiento.

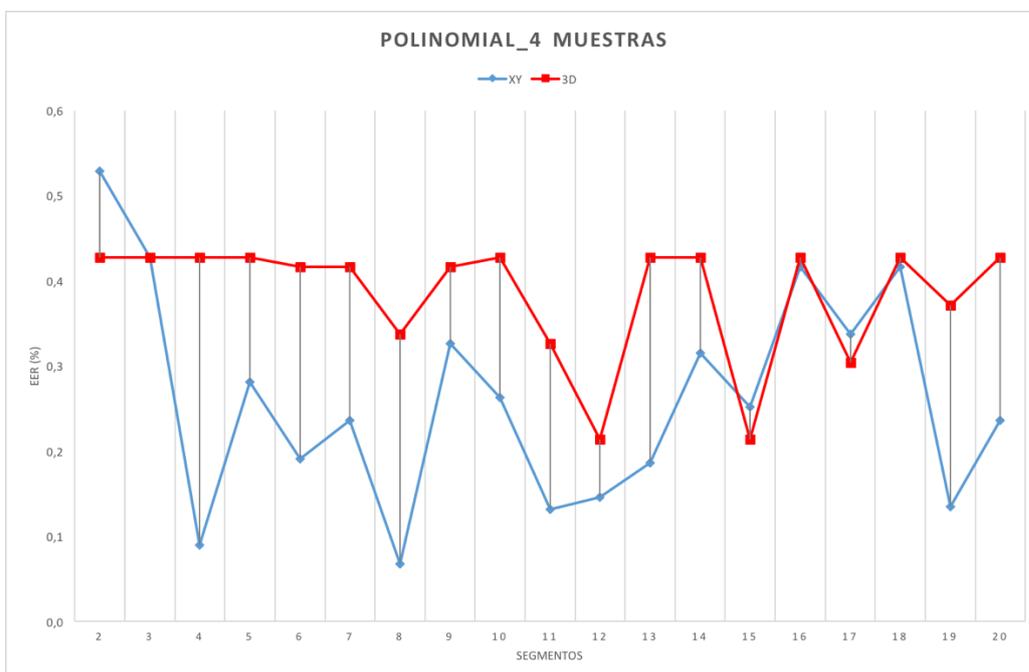


Figura 5.21. Comparación de los planos XY y XYZ, usando *kernel* polinómico y 4 muestras de entrenamiento.

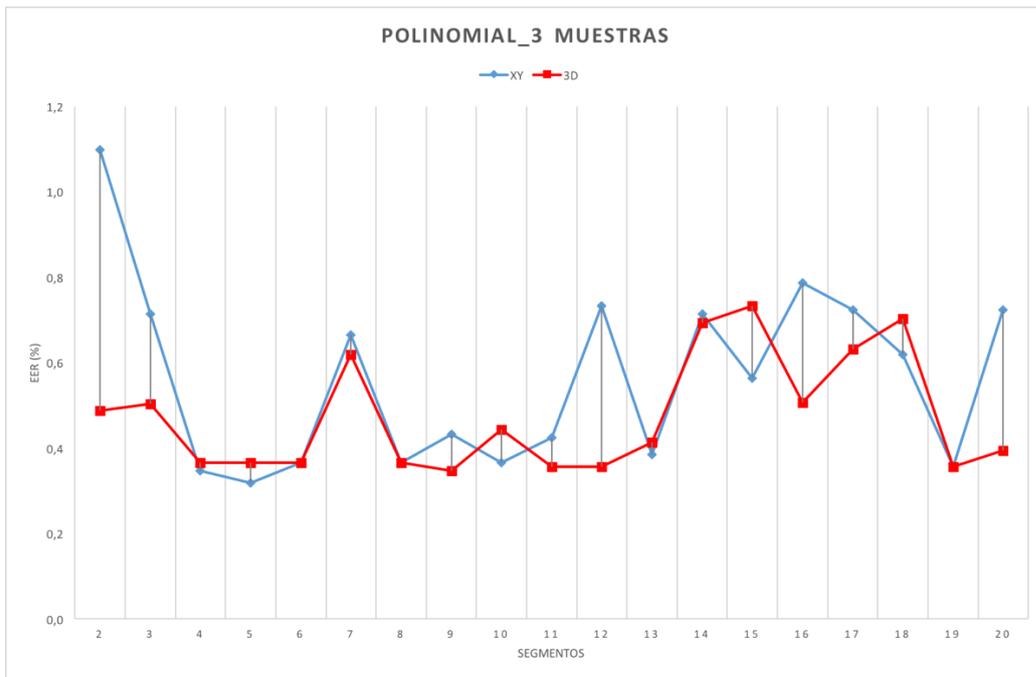


Figura 5.22. Comparación de los planos XY y XYZ, usando *kernel* polinómico y 3 muestras de entrenamiento.

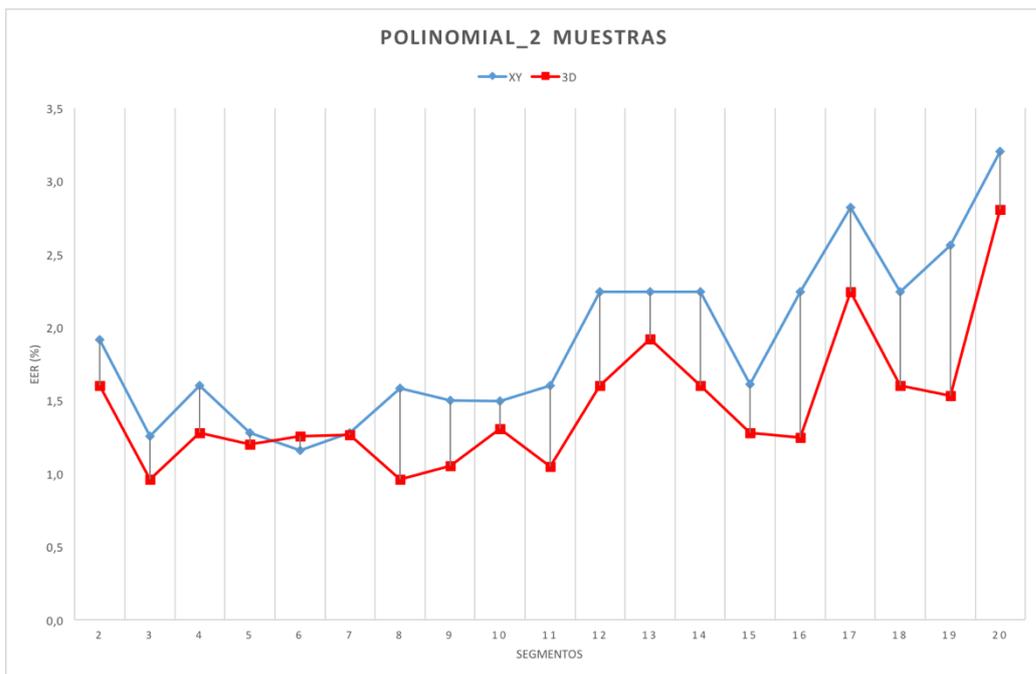


Figura 5.23. Comparación de los planos XY y XYZ, usando *kernel* polinómico y 2 muestras de entrenamiento.

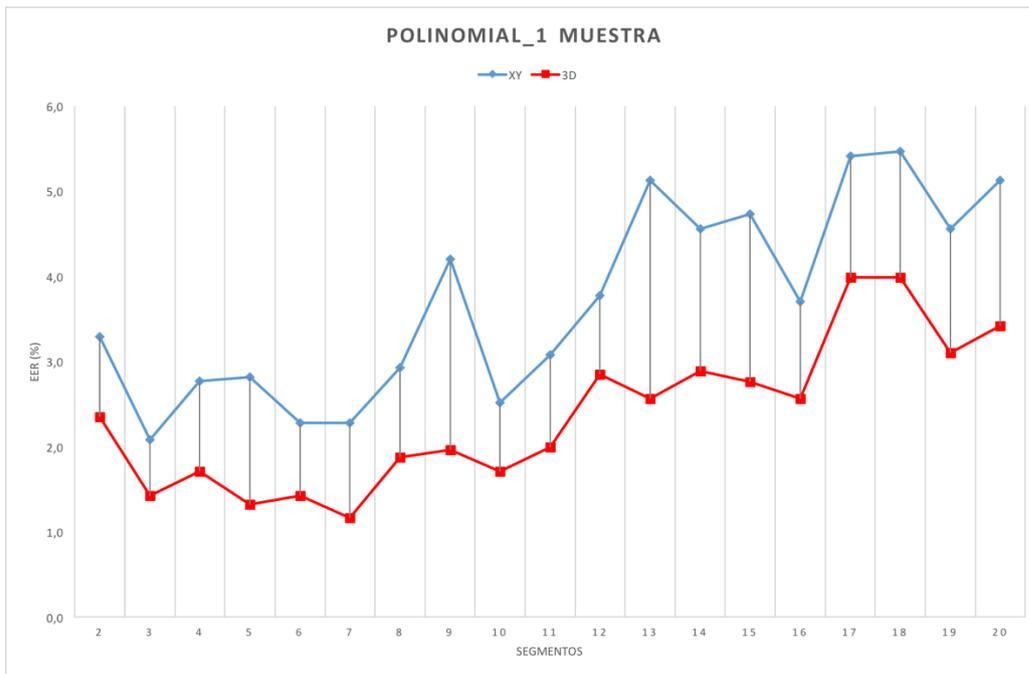


Figura 5.24. Comparación de los planos XY y XYZ, usando *kernel* polinómico y 1 muestra de entrenamiento.

En este caso, pasa lo mismo que para el *kernel* RBF para 5 y 4 muestras de entrenamiento. Sin embargo, para 3 muestras de entrenamiento se nota una mejoría con respecto al caso anterior. Se puede decir, que a partir de 3 muestras ya están muy a la par los resultados y en muchos casos se mejora algo. Para 2 muestras ya hay una clara mejoría y para 1 muestra se mejoran los resultados de todos los subconjuntos de una manera muy clara.

Como conclusión de este experimento se puede deducir que la tercera dimensión empieza a aportar información útil cuando se simula con 2 muestras de entrenamiento, aunque con 3 muestras los resultados son muy parecidos. Además, cabe indicar que este aporte de información es más evidente para el *kernel* polinómico que para el *kernel* RBF.

Experimento 5. Fusión de los conjuntos.

Con 1 muestra de entrenamiento el conjunto de señales que mejores resultados presenta es el del conjunto de todas las señales, es decir, el del plano XYZ. Por lo tanto, las fusiones se harán con el conjunto de señales del plano XYZ. Por un lado, se hará para el *kernel* polinómico y, por el otro, para el *kernel* RBF.

Después de fusionar los diferentes subconjuntos de distintas maneras se llega a que, para el **kernel polinómico**, la fusión que mejor resultado da es la que está compuesta por los **subconjuntos de todas las señales dividido en 5 segmentos y el mismo subconjunto dividido en 6 segmentos**. Esta fusión baja el valor del EER de un 1,1674% hasta un **1,1096%**.

En el caso del **kernel RBF**, la fusión se realiza para los **subconjuntos de todas las señales en 3 segmentos y el mismo dividido en 4 segmentos**. En este caso el valor de EER que se obtiene con esta fusión disminuye el EER de un 1,1396% a un **0,7890%**.

Experimento 6. Simulación del sistema con una base de datos ampliada a 70 usuarios.

Una vez realizados los experimentos anteriores, con los que se ha logrado bajar el valor de EER hasta un 0,7890% para 1 muestra de entrenamiento se procederá a realizar la simulación con la base de datos ampliada a 70 usuarios antes de realizar la simulación final con la base de datos final de 100 usuarios.

Como se establece en el experimento anterior, para el **kernel polinómico** se simula la fusión del subconjunto de todas las señales dividido en 5 segmentos con el subconjunto de todas las señales dividido en 6 segmentos. Por el otro lado, para el **kernel RBF** se simula la fusión del subconjunto de todas las señales dividido en 3 segmentos con el subconjunto de todas las señales dividido en 4 segmentos.

A continuación se muestra la tabla 5.1 con los resultados del experimento. Cada tabla (separando las tablas por el número de muestras de entrenamientos usadas en la simulación) muestra, en este orden, la **media** de todos los resultados y los valores **mínimo** y **máximo** que se han obtenido.

En el Anexo A se muestra una tabla más completa de los resultados. A los valores ya mencionados también se añaden la **moda** (el valor que más se repite); la **varianza** (es la media aritmética del cuadrado de las desviaciones respecto a la media de una distribución estadística), la **desviación típica** (es una medida del grado de dispersión de

Capítulo 5. Resultados

las observaciones alrededor de su valor medio) y la **mediana** (el valor que se situaría justo en medio si se colocaran todos los resultados en orden de menor a mayor).

Muestras	Kernel	Mean	Min	Max
5	RBF	0,0395	0,0000	0,2857
	Polinómico	0,1046	0,0000	0,3139
4	RBF	0,1020	0,0000	0,2381
	Polinómico	0,1611	0,0035	0,4762
3	RBF	0,1760	0,0059	0,6122
	Polinómico	0,2916	0,0204	0,8163
2	RBF	0,3765	0,0981	0,8929
	Polinómico	0,5483	0,1734	1,4101
1	RBF	1,0277	0,5327	1,9048
	Polinómico	1,4048	0,5467	2,5248

Tabla 5.1. Resultados de la simulación hecha con la BBDD de 70 usuarios

En las figuras 5.25 y 5.26 se puede ver la distribución de los resultados en gráficos *boxplots* (diagrama de cajas). Estos gráficos pueden verse individualmente y de una manera más detallada en el Anexo B.

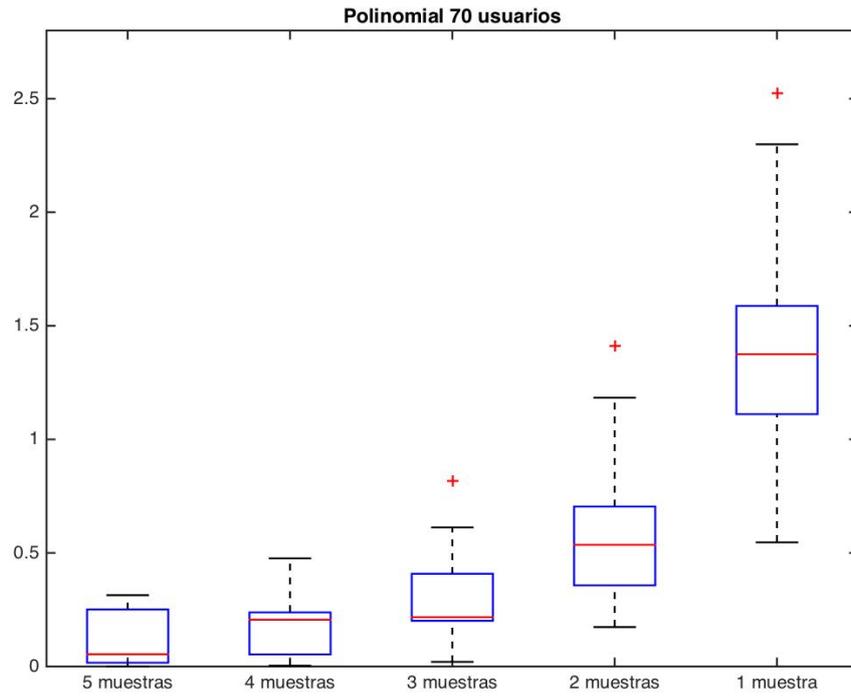


Figura 5.25. Conjunto de boxplots de resultados para 70 usuarios, usando *kernel* polinómico

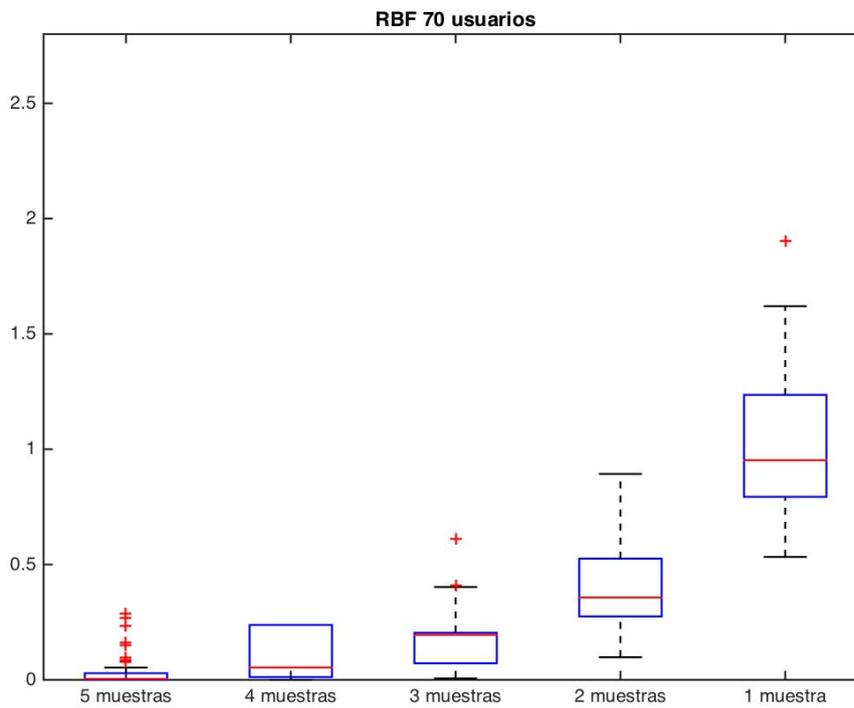


Figura 5.26. Conjunto de boxplots de resultados para 70 usuarios, usando *kernel* RBF

Experimento 7. Simulación del sistema con la base de datos final de 100 usuarios.

De la misma manera que en el experimento anterior se procede a realizar este último experimento. En este caso se usa la base de datos final de 100 usuarios. Obteniendo así los siguientes resultados mostrados en la tabla 5.2.

Muestras	Kernel	Mean	Min	Max
5	RBF	0,0751	0,0000	0,4000
	Polinómico	0,1207	0,0000	0,4000
4	RBF	0,1318	0,0000	0,5000
	Polinómico	0,1716	0,0034	0,5939
3	RBF	0,2218	0,0216	0,5714
	Polinómico	0,2897	0,0750	0,5714
2	RBF	0,4209	0,1212	0,7500
	Polinómico	0,5187	0,0850	1,0863
1	RBF	1,1017	0,6667	1,6667
	Polinómico	1,4185	0,8620	2,7778

Tabla 5.2. Resultados de la simulación hecha con la BBDD de 100 usuarios

De igual manera que en el experimento anterior, en las figuras 5.27 y 5.28 se representa la distribución de los resultados en diagramas de cajas (*boxplots*). Estos gráficos pueden verse individualmente y de una manera más detallada en el Anexo B.

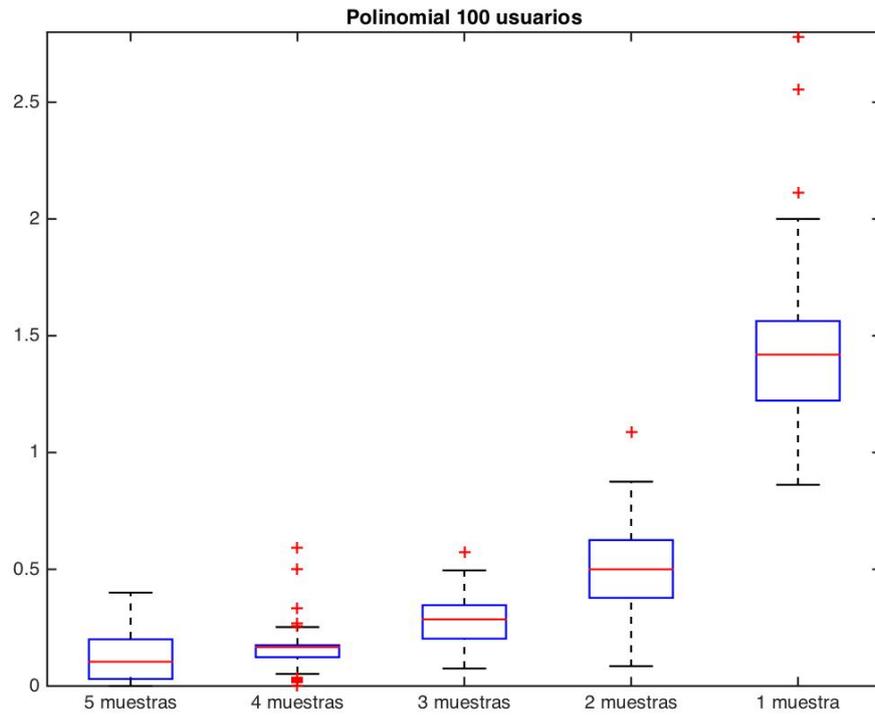


Figura 5.27. Conjunto de boxplots de resultados para 100 usuarios, usando *kernel* polinómico

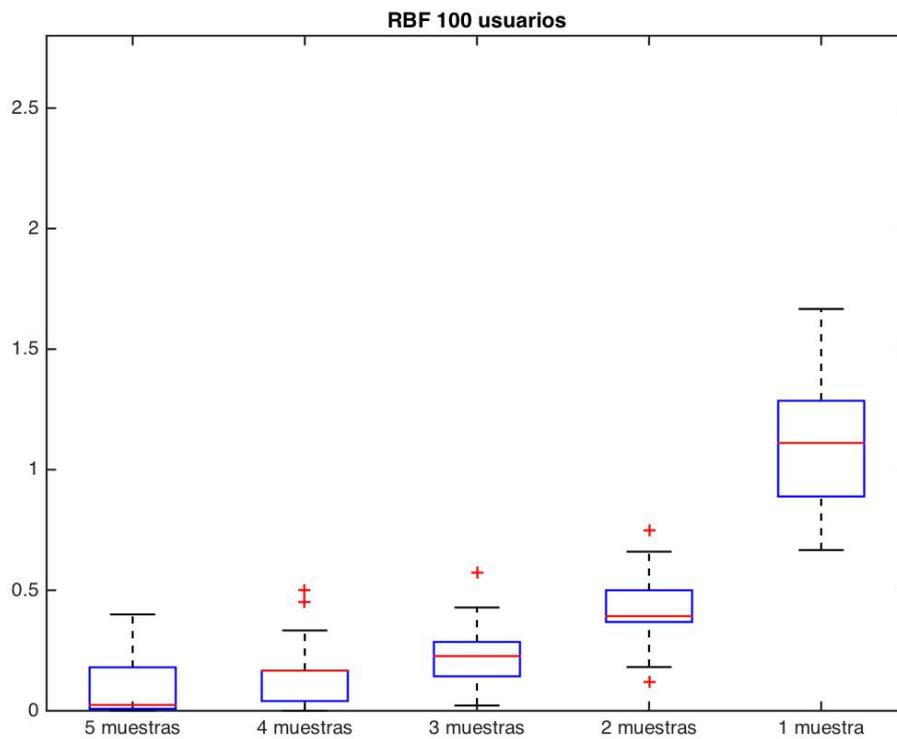


Figura 5.28. Conjunto de boxplots de resultados para 100 usuarios, usando *Kernel* RBF

Capítulo 5. Resultados

Para corroborar que la fusión aporta más información y, por lo tanto, se obtienen mejores resultados de EER se realiza la misma simulación para los mejores resultados que se dieron para 5 y para 1 muestra de manera individual, tanto para el *kernel* polinómico como para el *kernel* RBF. Es decir, se realizan 100 simulaciones seleccionando de manera aleatoria las muestras de entrenamiento para los subconjuntos siguientes:

- Para el *kernel* RBF:
 - Los mejores para 1 muestras son los subconjuntos de 4 y 5 segmentos para todas las señales (3D).
 - Los mejores para 5 muestras son los subconjuntos de 4, 5, 6, 7 y 8 segmentos para todas las señales del plano XY (2D).
- Para el *kernel* polinómico:
 - Los mejores para 1 muestras son los subconjuntos de 5 y 7 segmentos para todas las señales (3D).
 - Los mejores para 5 muestras son los subconjuntos de 7, 8 y 10 segmentos para todas las señales del plano XY (2D).

Como se indica, se hace la simulación de cada subconjunto por separado y se comparan los resultados con la simulación de la fusión, para así comprobar que realmente se han conseguido mejoras en el EER.

Hay que recordar, que en los experimentos previos a la hora de seleccionar cada fusión, se escogieron los subconjuntos que mejores resultados dan para 1 muestra de entrenamiento. Por lo tanto, hay otros subconjuntos que dan mejores resultados para 5 muestras que las fusiones que se realizaron. Todo esto para las primeras simulaciones que no se seleccionaban de manera aleatoria las muestras de entrenamiento. Entonces, podría ser lógico pensar que al realizar la simulación aleatoria sobre los subconjuntos que daban mejores resultados para 5 muestras también darán mejores resultados ahora para 5 muestras.

En cada tabla de las que se proporciona a continuación se muestran los resultados de la fusión, el subconjunto 2D que mejores valores dio y el subconjunto 3D que mejor valor

dio. La tabla 5.3 muestra los resultados para un *kernel* RBF y, por su parte, la tabla 6.4 muestra los resultados para un *kernel* polinómico. En cada una de las tablas se muestran los valores mínimos, máximos y los valores de la media de las 100 simulaciones de cada subconjunto.

RBF				
Muestras	Subconjunto	Mean	Min	Max
5	XY_4 seg	0.6657	0.2961	1.2000
	XY_5 seg	0.7530	0.2000	1.2000
	XY_6 seg	0.8744	0.3919	1.4000
	XY_7 seg	0.9313	0.2263	1.6000
	XY_8 seg	1.0261	0.5089	1.8000
	XYZ_4 seg	0.4729	0.0788	0.9879
	XYZ_5 seg	0.4746	0.1919	1.2000
	Fusión	0.0751	0	0.4000
4	XY_4 seg	0.8828	0.3333	1.5212
	XY_5 seg	0.9201	0.3846	1.5000
	XY_6 seg	1.0446	0.5660	1.6667
	XY_7 seg	1.1594	0.4725	2.0772
	XY_8 seg	1.2177	0.6667	1.8475
	XYZ_4 seg	0.5957	0.1667	1.0000
	XYZ_5 seg	0.5987	0.2163	1.1667
	Fusión	0.1318	0	0.5000
3	XY_4 seg	1.1503	0.5685	1.7143
	XY_5 seg	1.1740	0.7143	1.9562
	XY_6 seg	1.3727	0.6831	2.0449

	XY_7 seg	1.4298	0.6949	2.1429
	XY_8 seg	1.5566	0.9963	2.2948
	XYZ_4 seg	0.7799	0.4299	1.2857
	XYZ_5 seg	0.8328	0.3703	1.4286
	Fusión	0.2218	0.0216	0.5714
2	XY_4 seg	1.7220	1.0000	2.4624
	XY_5 seg	1.7362	1.2274	2.6250
	XY_6 seg	1.9191	1.3604	2.6250
	XY_7 seg	2.1618	1.3409	3.0438
	XY_8 seg	2.3740	1.6250	3.3750
	XYZ_4 seg	1.1836	0.7500	1.9295
	XYZ_5 seg	1.3106	0.8594	2.0000
	Fusión	0.4209	0.1212	0.7500
1	XY_4 seg	4.0013	2.5232	5.5556
	XY_5 seg	3.4730	2.4065	4.7481
	XY_6 seg	3.9997	2.8852	5.0000
	XY_7 seg	4.2534	2.7490	6.0000
	XY_8 seg	4.7498	3.8761	6.6883
	XYZ_4 seg	2.6406	1.8889	3.7778
	XYZ_5 seg	2.8369	1.8889	4.0000
	Fusión	1.1017	0.6667	1.6667

Tabla 5.3. Comparación de resultados para *kernel* RBF.

Polinómico				
Muestras	Subconjunto	Mean	Min	Max
5	XY_7 seg	1.0630	0.4000	1.7466
	XY_8 seg	1.0445	0.5706	1.6000
	XY_10 seg	1.3247	0.5899	2.0000
	XYZ_5 seg	0.4920	0.1071	0.8000
	XYZ_7 seg	0.6944	0.1939	1.4000
	Fusión	0.1207	0	0.4000
4	XY_7 seg	1.2481	0.5147	2.0000
	XY_8 seg	1.2942	0.5000	1.8899
	XY_10 seg	1.5587	1.0000	2.3000
	XYZ_5 seg	0.6371	0.2417	1.1667
	XYZ_7 seg	0.8072	0.2508	1.3276
	Fusión	0.1716	0.0034	0.5939
3	XY_7 seg	1.6009	0.9794	2.4286
	XY_8 seg	1.6874	1.0000	2.7143
	XY_10 seg	2.0966	1.1429	3.0000
	XYZ_5 seg	0.8429	0.4586	1.5714
	XYZ_7 seg	1.0768	0.5714	1.5714
	Fusión	0.2897	0.0750	0.5714
2	XY_7 seg	2.2118	1.4798	3.2500
	XY_8 seg	2.4373	1.6250	3.6250
	XY_10 seg	2.8773	1.8526	4.0000
	XYZ_5 seg	1.3355	0.8750	2.2500

	XYZ_7 seg	1.5764	0.9798	2.5026
	Fusión	0.5187	0.0850	1.0863
1	XY_7 seg	4.6371	3.4444	5.6755
	XY_8 seg	5.0046	3.7933	6.55562
	XY_10 seg	5.6987	4.0000	7.9559
	XYZ_5 seg	3.0782	2.0000	4.2629
	XYZ_7 seg	3.4179	2.3333	5.3333
	Fusión	1.4185	0.8620	2.7778

Tabla 5.4. Comparación de resultados para *kernel* RBF.

Se puede observar que, para los dos casos de *kernel*, los resultados obtenidos del subconjunto fusionado es considerablemente mejor frente a cualquier otro subconjunto y para cualquier número de muestras de entrenamiento.

Además, cabe destacar que con 100 usuarios la tercera dimensión aporta mucha información con respecto al plano 2D. Hay que resaltar que, incluso para 5 muestras de entrenamiento donde se había visto que con 39 usuarios la tercera dimensión empeoraba los resultados, ahora, sin embargo, los resultados son claramente mejores.

Entonces, se concluye que **que, según la línea de trabajo seguida, se puede confirmar la hipótesis sobre que la tercera dimensión aporta información discriminante.**

Capítulo 6. Conclusiones y líneas futuras

En este capítulo se presentan las conclusiones a las que se ha llegado en el desarrollo de este proyecto, así como las líneas futuras de trabajo que surgen tras la realización del mismo.

6.1. Conclusiones

En primer lugar, hay que destacar que se han logrado los objetivos principales del proyecto. Se ha conseguido desarrollar un sistema de clasificación con un porcentaje de error más bajo que el logrado en los trabajos nombrados en la sección de antecedentes.

En cuanto a qué conjuntos son los que aportan mayor información se extraen dos conclusiones. La primera es que, para cada plano, los conjuntos que mejores resultados ofrecen son los constituidos por **todas las señales de dicho plano**. La segunda, partiendo de la conclusión anterior, es que los dos conjuntos que aportan más información son los correspondientes al **plano XY (2D) y al plano XYZ (3D)**. Estos dos conjuntos ofrecen resultados mucho mejores que los dos planos restantes (plano XZ e YZ).

A priori, el plano XY ofrece mejores resultados que el plano XYZ, es decir, la coordenada Z empeora los resultados. Sin embargo, a medida que se disminuyen las muestras de entrenamiento, la tercera dimensión va aportando mayor información. **A partir de 2 muestras de entrenamiento es el plano XYZ el que ofrece mejores resultados y, por lo tanto, la tercera dimensión (coordenada Z) sí aporta información.**

También se demostró que la **fusión de dos subconjuntos de datos genera mejores resultados** que cada subconjunto por separado. Los resultados son considerablemente mejores.

Por otro lado, **se descartó añadir** tres parámetros estadísticos que se pensó que podrían aportar información. Éstos son la **entropía de Shannon, la curtosis y la asimetría o**

skewness. Este rechazo se deba a que al añadirlos a la parametrización de los conjuntos de señales, los **resultados empeoraban**.

Por último, cabe mencionar que todas estas conclusiones mencionadas hasta el momento fueron obtenidas usando la base de datos inicial de 39 usuarios. Pero, al realizar la simulación con la **base de datos final de 100 usuarios**, se extrae la conclusión más importante de este proyecto. En este caso, **la tercera dimensión (plano XYZ) sí aporta información y, por lo tanto, da mejores resultados que el plano XY sin importar el número de muestras de entrenamiento usadas**. Además, se sigue cumpliendo que a medida que disminuye el número de muestras de entrenamiento la diferencia entre los resultados de los dos planos (XY y XYZ) es mayor.

Por lo tanto, se puede decir que **mientras mayor sea el número de usuarios en nuestra base de datos y menor sea el número de muestras de entrenamiento, el plano XYZ (3D) aporta cada vez mayor información que cualquier plano 2D**.

Por último, si se observan los resultados para la base de datos de 100 usuarios, se ve que los mejores resultados se obtienen cuando se simula con 5 muestras de entrenamiento usando un *kernel* RBF. Sin embargo, se podría reducir el número de muestras de entrenamiento hasta 2 y el resultado máximo de EER no sobrepasa de 0,75%. Por lo tanto, se puede concluir que el sistema que mejores resultados ofrece para 100 usuarios sería el constituido por un **clasificador LS-SVM con las siguientes características**:

- *Kernel*: RBF
- $\gamma = 400$
- $\sigma^2 = 150$
- Número de muestras de entrenamiento = 2
- Conjunto resultante de la fusión entre: **Conjunto de todas las señales dividido en 3 segmentos y el mismo dividido en 4 segmentos**

6.2. Comparación de resultados

En la tabla 6.1 se muestra el valor de EER obtenido en los diferentes trabajos nombrados en el capítulo de antecedentes y el obtenido en este proyecto para así realizar una comparativa.

TRABAJO	EER(%)	BASE DE DATOS (USUARIOS)
[18] Bayes	1,81	96
[18] DTW	2,12	96
[20] DP-matching	5	22
[21] uWave System	2	25
[34] DTW	0,6	50
[5] SVM	0,36	39
[5] DTW	0,51	39
*[Este proyecto] LS-SVM con <i>kernel</i> RBF.	0,075	100
*[Este proyecto] LS-SVM con <i>kernel</i> polinómico.	0,12	100

Tabla 6.1. Comparativa de rendimiento entre diferentes trabajos

*Valor medio de las 100 simulaciones realizadas de manera aleatoria y usando 5 muestras de entrenamiento.

Se puede observar cómo se han obtenido resultados bastante mejores que los obtenidos en otros trabajos. Si sólo se compara el mejor resultado obtenido en este proyecto con el mejor resultado obtenido en el trabajo en el que se usó una base de datos de 96 usuarios, que es la más parecida a la de este proyecto, se puede observar que el resultado obtenido es 24 veces mejor.

6.3. Líneas futuras

En cuanto a las líneas futuras que se pueden seguir a partir del desarrollo de este trabajo se plantean las siguientes:

- Realizar experimentos con diferentes cantidades de usuarios en la base de datos para ver realmente la evolución de los resultados a medida que aumenta o disminuye la base de datos.
- Ampliar la base de datos con falsificaciones de firmas para los diferentes usuarios.
- Realizar un estudio similar al propuesto en este proyecto pero basado en técnicas off-line.
- Implementar un sistema de clasificación que integre las técnicas off-line propuestas en el punto anterior con las técnicas on-line usadas en este proyecto.

Anexo A. Tablas de resultados

A continuación se muestran los resultados de los experimentos en tablas, obteniendo así una visión más detallada.

A.1. Resultados del experimento 2

Aquí se pueden ver todos los resultados exactos obtenidos para los conjuntos de todas las señales de los planos XY y XYZ.

Segmentos	EER(%)		
2	0,2834	11	0,0810
3	0,2699	12	0,1215
4	0	13	0,1215
5	0	14	0,5128
6	0	15	0,4049
7	0	16	0,2630
8	0	17	0,4049
9	0,0135	18	0,4453
10	0,0270	19	0,2429
		20	0,4049

Tabla A.1. Resultados del conjunto de todas señales del plano XY para 5 muestras entrenamiento con *kernel* RBF

Anexo A. Tablas de resultados

SEGMENTOS	EER (%)		
2	0,2429	11	0,4453
3	0,1889	12	0,3156
4	0,1484	13	0,5128
5	0,1350	14	0,9117
6	0,2024	15	0,7436
7	0,2761	16	0,7270
8	0,1484	17	0,7326
9	0,3644	18	0,7394
10	0,3550	19	0,5486
		20	0,6576

Tabla A.2. Resultados del conjunto de todas señales del plano XYZ para 5 muestras entrenamiento con *kernel* RBF

Segmentos	EER(%)		
2	3,7037	11	2,5641
3	2,1232	12	3,4188
4	2,1154	13	3,7311
5	2,2492	14	4,2735
6	1,7094	15	4,0266
7	1,7593	16	3,2634
8	2,2792	17	4,8661
9	2,7857	18	4,2551
10	2,5641	19	4,0354
		20	4,4377

Tabla A.3. Resultados del conjunto de todas señales del plano XY para 1 muestra entrenamiento con *kernel* RBF

Segmentos	EER(%)		
2	2,0151	11	3,1009
3	1,2369	12	3,0303
4	1,1396	13	3,4952
5	1,1396	14	3,6377
6	1,6166	15	3,5652
7	1,4940	16	3,4047
8	1,9563	17	4,3996
9	2,1652	18	3,6618
10	2,4171	19	4,5599
		20	4,2710

Tabla A.4. Resultados del conjunto de todas señales del plano XYZ con 1 muestra entrenamiento para *kernel* RBF

Segmentos	EER(%)		
2	0,5128	11	0,0945
3	0,4993	12	0,1350
4	0,0540	13	0,0945
5	0,0405	14	0,3779
6	0,0405	15	0,1889
7	0,0135	16	0,0405
8	0	17	0,1619
9	0,0945	18	0,1484
10	0,0135	19	0,0540
		20	0,1350

Tabla A.5. Resultados del conjunto de todas señales del plano XY con 5 muestras entrenamiento para *kernel* polinómico

Anexo A. Tablas de resultados

Segmentos	EER(%)		
2	0,5128	11	0,2834
3	0,5128	12	0,0810
4	0,5128	13	0,1889
5	0,5128	14	0,4723
6	0,4184	15	0,1446
7	0,3813	16	0,4339
8	0,1484	17	0,1484
9	0,2294	18	0,3779
10	0,4318	19	0,1754
		20	0,2159

Tabla A.6. Resultados del conjunto de todas señales del plano XYZ con 5 muestras entrenamiento para *kernel* polinómico

Segmentos	EER(%)		
2	3,2946	11	3,0821
3	2,0869	12	3,7764
4	2,7695	13	5,1282
5	2,8219	14	4,5584
6	2,2792	15	4,7350
7	2,2792	16	3,7037
8	2,9313	17	5,4131
9	4,2039	18	5,4691
10	2,5191	19	4,5584
		20	5,1282

Tabla A.7. Resultados del conjunto de todas señales del plano XY con 1 muestra entrenamiento para *kernel* polinómico

Segmentos	EER(%)		
		11	1,9943
2	2,3504	12	2,8490
3	1,4245	13	2,5641
4	1,7094	14	2,8906
5	1,3272	15	2,7661
6	1,4245	16	2,5641
7	1,1674	17	3,9886
8	1,8803	18	3,9886
9	1,9643	19	3,0997
10	1,7094	20	3,4188

Tabla A.8. Resultados del conjunto de todas señales del plano XYZ con 1 muestra entrenamiento para *kernel* polinómico

A.2. Resultados del experimento 3

A continuación se presentan las tablas de resultados obtenidas para la simulación del experimento 3. La columna "EER" corresponde al valor de EER (%) obtenido antes de añadir cualquiera de las nuevas variables aleatorias. Las celdas coloreadas de azul, indican los casos en que se ha mejorado ese valor de EER.

Anexo A. Tablas de resultados

Subconj señales	EER	Skew	Entr	kurt	Skew + Entr	Skew + Kurt	Kurt + Ent	Skew + Kurt + Ent
<i>Todas xy 8 seg</i>	0	0,08097	0	0,04049	0,07890	0,16194	0,04049	0,14845
<i>Todas 3D 12 seg</i>	0,08097	0,09447	0,06748	0,04049	0,06748	0,04049	0,04049	0,04049
<i>Pos índice xyz 7 seg</i>	0,09447	0,28928	0,22354	0,20243	0,40763	0,37787	0,47337	0,51282
<i>Pos índice xy 11 seg</i>	0,10796	0,49933	0,09447	1,02564	0,51282	0,97436	1,02564	1,02564
<i>Pos palma xyz 22 seg</i>	0,24291	0,42308	0,51282	0,62078	0,92308	1,02564	0,76266	1,06509
<i>Vel índice xyz 6 seg</i>	0,36437	0,51282	0,51282	1,01215	0,51282	1,02564	1,01215	1,02564
<i>Pos palma xy 22 seg</i>	0,36437	0,97166	0,51282	0,89744	1,02564	1,51147	1,09139	1,49798
<i>Todas 3D 7 seg</i>	0,38133	0,32873	0,36437	0,51282	0,2969	0,51282	0,51282	0,51282
<i>Todas xy 3 seg</i>	0,49933	0,82321	0,51282	0,51282	0,71525	0,51282	0,51282	0,49933

Tabla A.9. Variación del EER al añadir nuevos parámetros estadísticos para 5 muestras.

Subconj señales	EER	Skew	Entr	kurt	Skew + Entr	Skew + Kurt	Kurt + Ent	Skew + Kurt + Ent
<i>Todas xy 8 seg</i>	2,93131	3,70370	3,09642	4,27350	3,64309	5,12821	4,27350	5,12821
<i>Todas 3D 12 seg</i>	2,84900	4,66046	2,81811	4,55840	4,55840	6,26781	4,55840	6,12368
<i>Pos índice xyz 7 seg</i>	2,42877	2,90598	2,27170	4,63115	2,67674	4,84330	4,55840	4,36626
<i>Pos índice xy 11 seg</i>	4,27350	5,69801	3,66127	6,86998	5,52318	7,50055	5,06823	6,18534
<i>Pos palma xyz 22 seg</i>	5,41311	7,12251	6,18328	6,26781	7,41928	7,91741	6,68385	7,97721
<i>Vel índice xyz 6 seg</i>	5,08322	4,84330	6,26781	7,06253	6,03591	6,83761	7,69231	7,27368
<i>Pos palma xy 22 seg</i>	6,37587	9,61323	6,83761	8,26211	9,11681	10,82621	7,97721	9,64482
<i>Todas 3D 7 seg</i>	1,16740	1,85185	1,31187	3,18519	1,97531	3,41880	3,04393	3,41880
<i>Todas xy 3 seg</i>	2,08689	2,02429	1,97181	3,09320	1,99430	2,84900	3,11891	2,95811

Tabla A.10. Variación del EER al añadir nuevos parámetros estadísticos para 1 muestra.

A.3. Resultados del experimento 4

Se ha seleccionado el rango de 3 a 10 segmentos, que se considera que es el rango que muestra la mayor información. Se obvian las simulaciones para 5 y 1 muestra de entrenamiento, ya que han sido expuestas anteriormente.

Segmentos	EER(%)
3	0,4274
4	0,0900
5	0,2812
6	0,1912
7	0,2362
8	0,0675
9	0,3261
10	0,2630

Tabla A.11. Resultados del conjunto del plano XY para 4 muestras entrenamiento usando *kernel* polinómico

Segmentos	EER(%)
3	0,4274
4	0,4274
5	0,4274
6	0,4161
7	0,4161
8	0,3374
9	0,4161
10	0,4274

Tabla A.12. Resultados del conjunto del plano XYZ para 4 muestras entrenamiento usando *kernel* polinómico

Anexo A. Tablas de resultados

Segmentos	EER(%)
3	0,7133
4	0,3470
5	0,3181
6	0,3663
7	0,6651
8	0,3663
9	0,4320
10	0,3663

Tabla A.13. Resultados del conjunto del plano XY para 3 muestras entrenamiento usando *kernel* polinómico

Segmentos	EER(%)
3	0,5037
4	0,3663
5	0,3663
6	0,3663
7	0,6199
8	0,3663
9	0,3470
10	0,4434

Tabla A.14. Resultados del conjunto del plano XYZ para 3 muestras entrenamiento usando *kernel* polinómico

Segmentos	EER(%)
3	1,2567
4	1,6026
5	1,2821
6	1,1588
7	1,2821
8	1,5857
9	1,5009
10	1,4957

Tabla A.15. Resultados del conjunto del plano XY para 2 muestras entrenamiento usando *kernel* polinómico

Segmentos	EER(%)
3	0,9615
4	1,2821
5	1,2019
6	1,2567
7	1,2652
8	0,9615
9	1,0519
10	1,3074

Tabla A.16. Resultados del conjunto del plano XYZ para 2 muestras entrenamiento usando *kernel* polinómico

Anexo A. Tablas de resultados

Segmentos	EER(%)
3	0,4274
4	0,0112
5	0,0787
6	0,1012
7	0,0337
8	0
9	0,0900
10	0,0337

Tabla A.17. Resultados del conjunto del plano XY para 4 muestras entrenamiento usando *kernel* RBF

Segmentos	EER(%)
3	0,1912
4	0,0900
5	0,1534
6	0,2699
7	0,3374
8	0,3178
9	0,4274
10	0,4274

Tabla A.18. Resultados del conjunto del plano XYZ para 4 muestras entrenamiento usando *kernel* RBF

Segmentos	EER(%)
3	0,5013
4	0,1157
5	0,0771
6	0,3567
7	0,3567
8	0,3567
9	0,3663
10	0,3567

Tabla A.19. Resultados del conjunto del plano XY para 3 muestras entrenamiento usando *kernel* RBF

Segmentos	EER(%)
3	0,2313
4	0,1832
5	0,2217
6	0,3567
7	0,5205
8	0,3663
9	0,5769
10	0,4434

Tabla A.20. Resultados del conjunto del plano XYZ para 3 muestras entrenamiento usando *kernel* RBF

Anexo A. Tablas de resultados

Segmentos	EER(%)
3	1,2567
4	1,1724
5	0,9615
6	1,0084
7	1,2821
8	0,9615
9	0,9531
10	1,2508

Tabla A.21. Resultados del conjunto del plano XY para 2 muestras entrenamiento usando *kernel* RBF

Segmentos	EER(%)
3	0,9615
4	1,0628
5	0,6242
6	0,9615
7	1,2821
8	0,9615
9	0,9362
10	1,6026

Tabla A.22. Resultados del conjunto del plano XYZ para 2 muestras entrenamiento con *kernel* RBF

A.4. Resultados del experimento 6 y del experimento 7

Por último, se muestran las tablas completas correspondientes a los dos últimos experimentos. Se añaden, a los resultados ya mostrados, los valores de la moda (*Mode*), la varianza (*Var*), la desviación típica (*Std*) y la mediana (*Median*).

5 muestras							
Kernel	Mean	min	max	Mode	Var	Std	Median
RBF	0,0395	0,0000	0,2857	0,0000	0,0065	0,0805	0,0041
Polinómico	0,1046	0,0000	0,3139	0,0000	0,0124	0,1115	0,0538
4 muestras							
Kernel	Mean	min	max	Mode	Var	Std	Median
RBF	0,1020	0,0000	0,2381	0,2381	0,0097	0,0987	0,0535
Polinómico	0,1611	0,0035	0,4762	0,2381	0,0111	0,1053	0,2059
3 muestras							
Kernel	Mean	min	max	Mode	Var	Std	Median
RBF	0,1760	0,0059	0,6122	0,2041	0,0148	0,1218	0,1952
Polinómico	0,2916	0,0204	0,8163	0,2041	0,0191	0,1383	0,2170
2 muestras							
Kernel	Mean	min	max	Mode	Var	Std	Median
RBF	0,3765	0,0981	0,8929	0,3571	0,0219	0,1479	0,3571
Polinómico	0,5483	0,1734	1,4101	0,5357	0,0466	0,2158	0,5357
1 muestras							
Kernel	Mean	min	max	Mode	Var	Std	Median
RBF	1,0277	0,5327	1,9048	0,7937	0,0648	0,2545	0,9524
Polinómico	1,4048	0,5467	2,5248	1,1111	0,1319	0,3631	1,3750

Tabla A.21. Resultados del experimento 6, para 70 usuarios

Anexo A. Tablas de resultados

5 muestras							
Kernel	Mean	min	max	Mode	Var	Std	Median
RBF	0,0751	0,0000	0,4000	0,0000	0,0081	0,0897	0,0241
Polinómico	0,1207	0,0000	0,4000	0,2000	0,0098	0,0991	0,1040
4 muestras							
Kernel	Mean	min	max	Mode	Var	Std	Median
RBF	0,1318	0,0000	0,5000	0,1667	0,0107	0,1037	0,1667
Polinómico	0,1716	0,0034	0,5939	0,1667	0,0111	0,1052	0,1667
3 muestras							
Kernel	Mean	min	max	Mode	Var	Std	Median
RBF	0,2218	0,0216	0,5714	0,1429	0,0099	0,0996	0,2267
Polinómico	0,2897	0,0750	0,5714	0,1429	0,0133	0,1151	0,2857
2 muestras							
Kernel	Mean	min	max	Mode	Var	Std	Median
RBF	0,4209	0,1212	0,7500	0,2500	0,0162	0,1274	0,3928
Polinómico	0,5187	0,0850	1,0863	0,5000	0,0276	0,1660	0,5000
1 muestras							
Kernel	Mean	min	max	Mode	Var	Std	Median
RBF	1,1017	0,6667	1,6667	0,8889	0,0524	0,2290	1,1111
Polinómico	1,4185	0,8620	2,7778	1,4444	0,1076	0,3280	1,4192

Tabla A.22. Resultados del experimento 7, para 100 usuarios

Anexo B. Boxplots

En este anexo se muestran los resultados finales (experimentos 6 y 7) mediante *boxplots* o diagramas de cajas. Se trata de gráficos que están basados en cuartiles y en los que se visualiza la distribución de los resultados obtenidos. En la figura B.1 se muestra un diagrama boxplot explicativo.

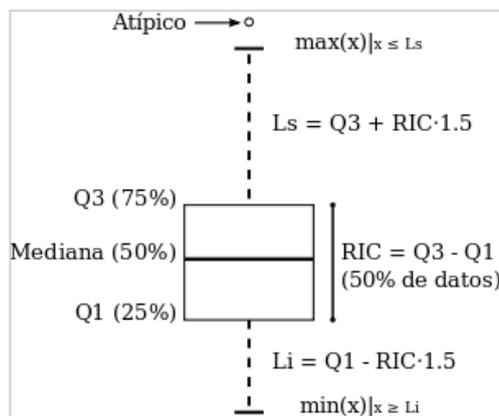


Figura B.1. Explicación del boxplot

En primer lugar, el rango de figuras [B.2 – B.6] muestran los resultados para la base de datos de 70 usuarios usando un *kernel* polinómico.

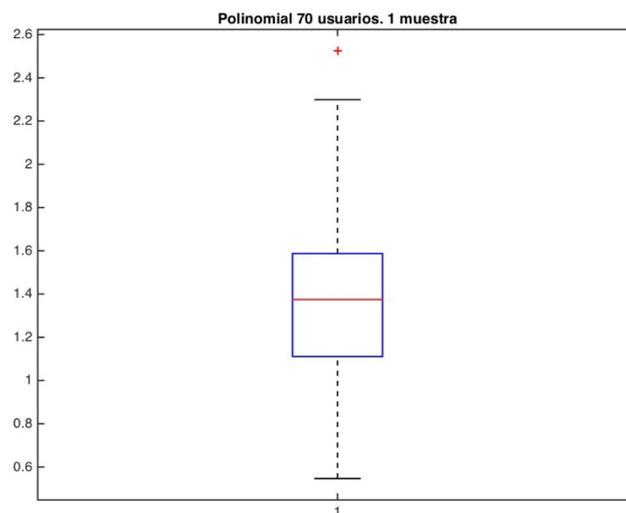


Figura B.2. Boxplot de resultados para 70 usuarios con 1 muestra entrenamiento usando *kernel* polinómico

Anexo B. Boxplots

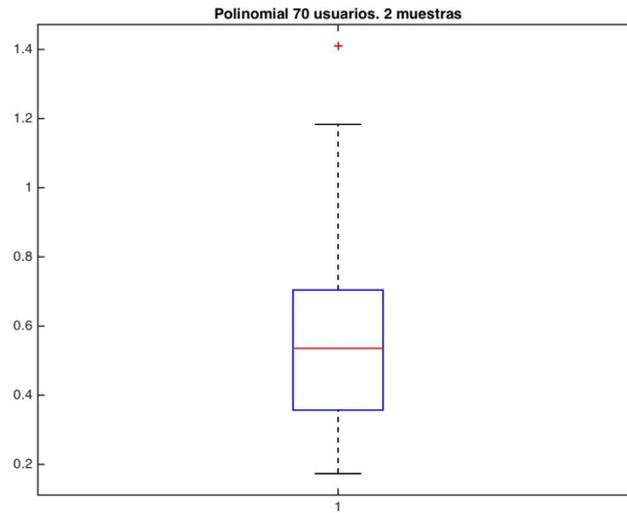


Figura B.3. Boxplot de resultados para 70 usuarios con 2 muestras entrenamiento usando *kernel* polinómico

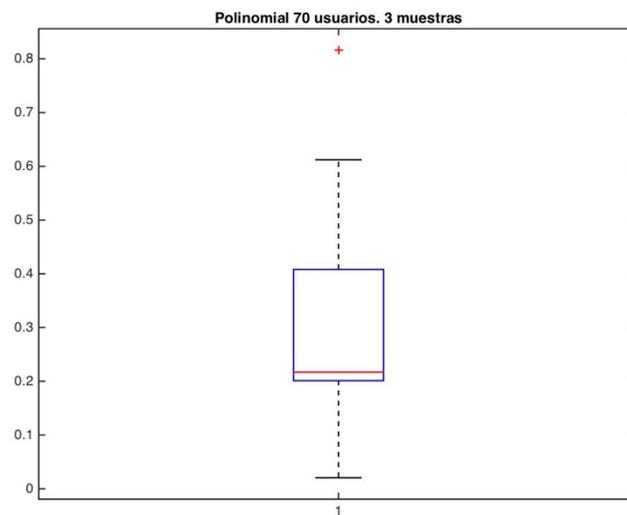


Figura B.4. Boxplot de resultados para 70 usuarios con 3 muestras entrenamiento usando *kernel* polinómico

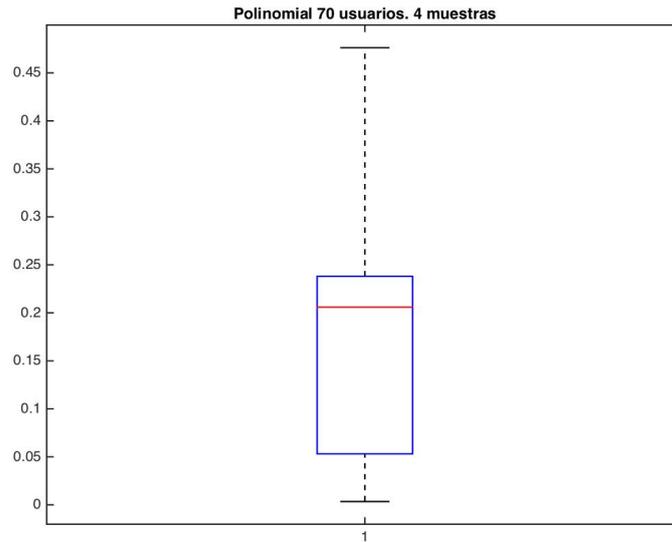


Figura B.5. Boxplot de resultados para 70 usuarios con 4 muestras entrenamiento usando *kernel* polinómico

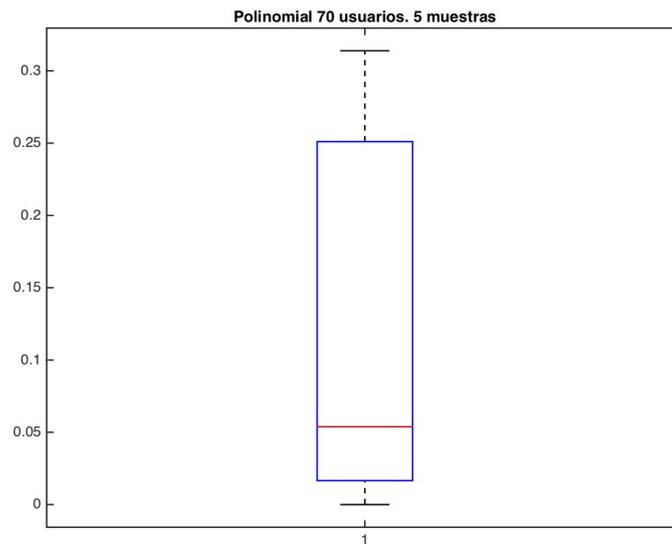


Figura B.6. Boxplot de resultados para 70 usuarios con 5 muestras entrenamiento usando *kernel* polinómico

En el rango de figuras [B.7 – B.11] se tienen los resultados para la base de datos de 70 usuarios usando un *kernel* polinómico.

Anexo B. Boxplots

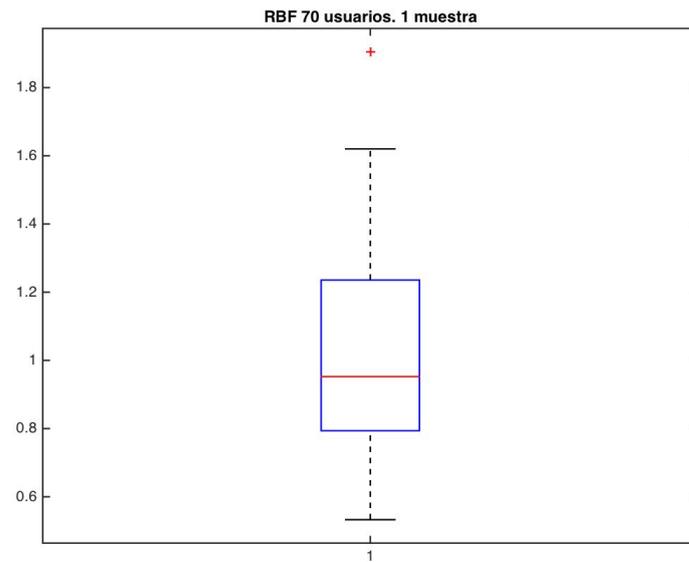


Figura B.7. Boxplot de resultados para 70 usuarios con 1 muestra entrenamiento usando *kernel* RBF

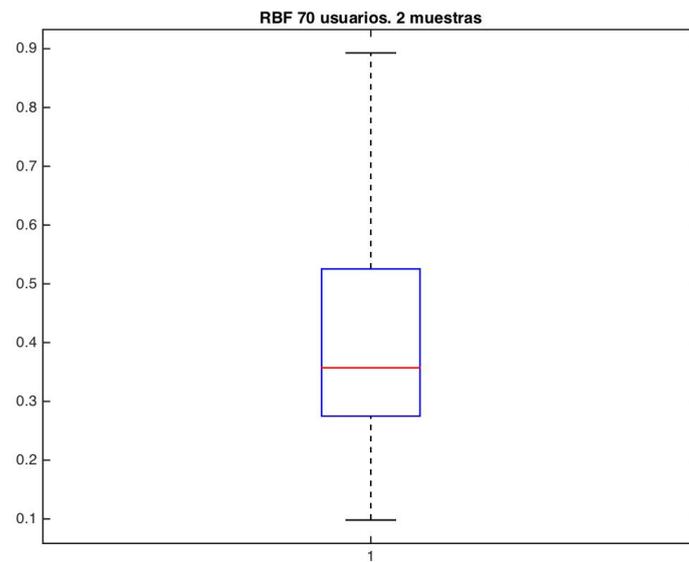


Figura B.8. Boxplot de resultados para 70 usuarios con 2 muestras entrenamiento usando *kernel* RBF

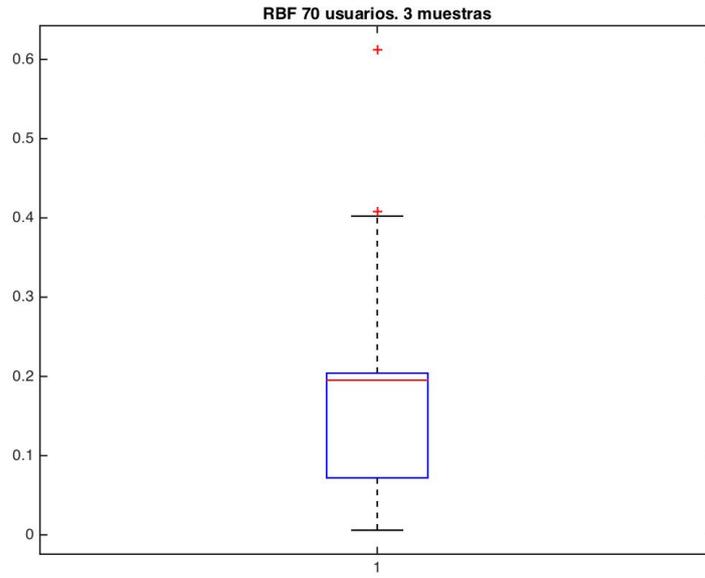


Figura B.9. Boxplot de resultados para 70 usuarios con 2 muestras entrenamiento usando *kernel* RBF

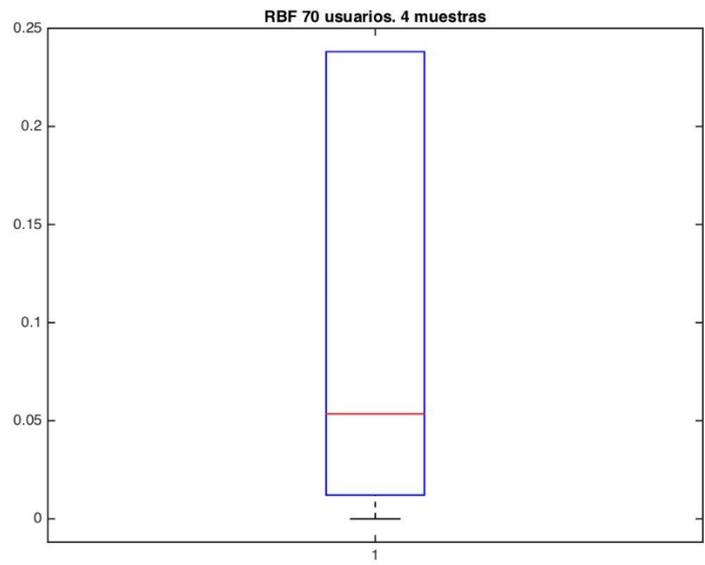


Figura B.10. Boxplot de resultados para 70 usuarios con 4 muestras entrenamiento usando *kernel* RBF

Anexo B. Boxplots

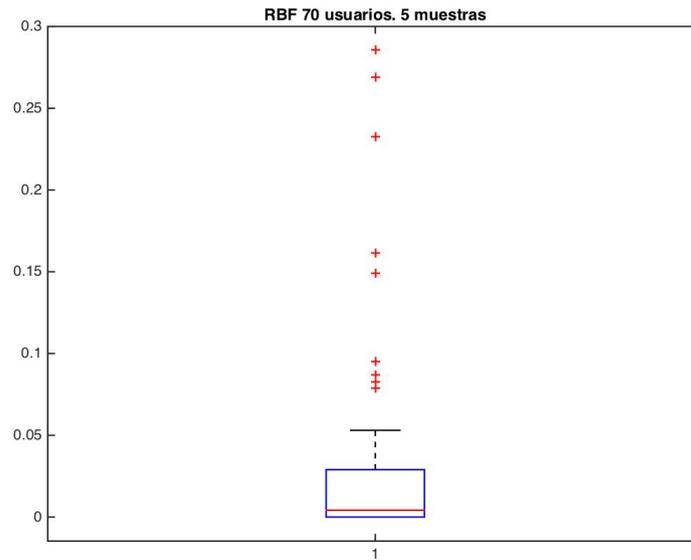


Figura B.11. Boxplot de resultados para 70 usuarios con 5 muestras entrenamiento usando *kernel* RBF

A continuación, se muestran los resultados para la base de datos de 100 usuarios usando *kernel* polinómico en las figuras del rango [B.12 – B.16].

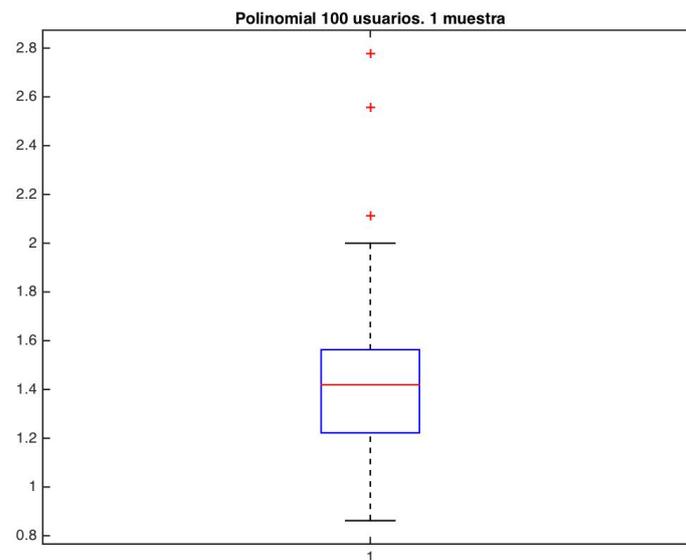


Figura B.12. Boxplot de resultados para 100 usuarios con 1 muestra entrenamiento usando *kernel* polinómico

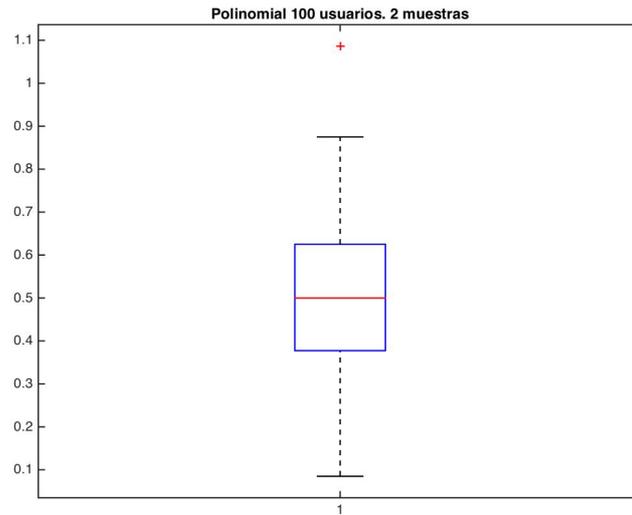


Figura B.13. Boxplot de resultados para 100 usuarios con 2 muestras entrenamiento usando *kernel* polinómico

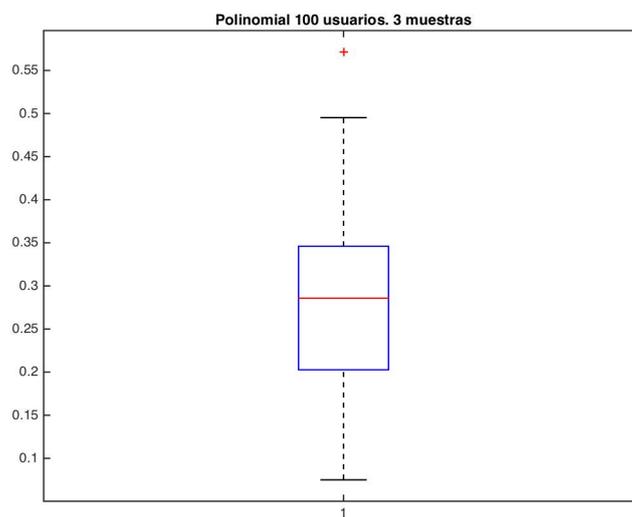


Figura B.14. Boxplot de resultados para 100 usuarios con 3 muestras entrenamiento usando *kernel* polinómico

Anexo B. Boxplots

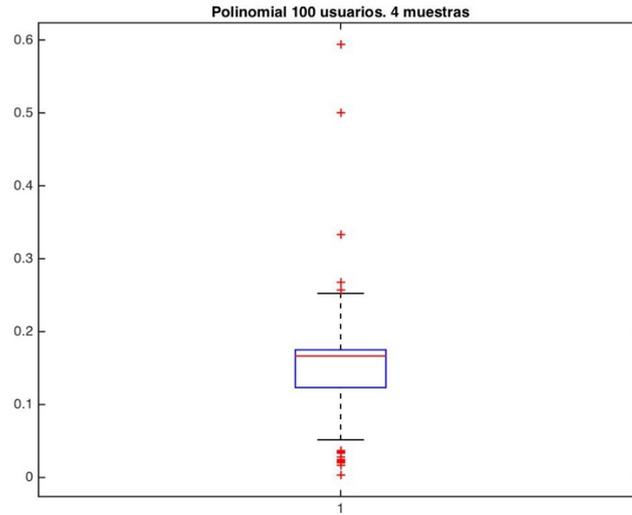


Figura B.15. Boxplot de resultados para 100 usuarios con 4 muestras entrenamiento usando *kernel* polinómico

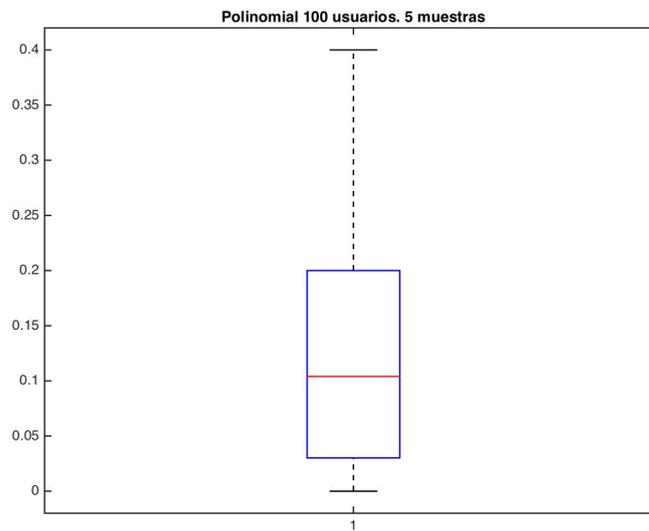


Figura B.16. Boxplot de resultados para 100 usuarios con 5 muestras entrenamiento usando *kernel* polinómico

Se procede igual para la misma base de datos de 100 usuarios usando un *kernel* RBF y se visualizan en el rango de figuras [B.17 – B.21]

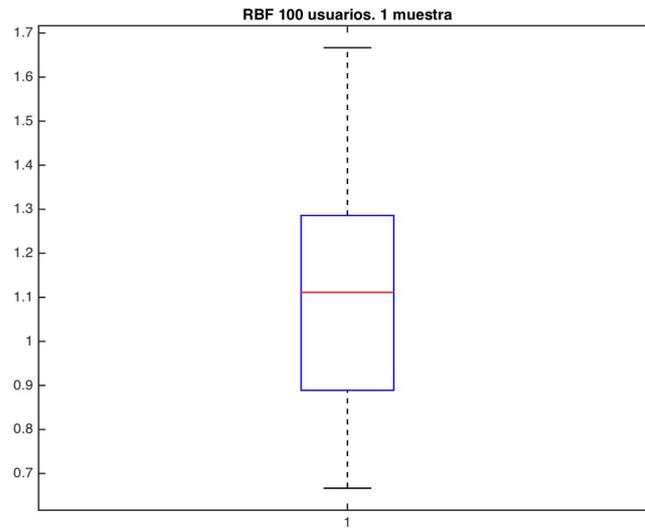


Figura B.17. Boxplot de resultados para 100 usuarios con 1 muestra entrenamiento usando *kernel* RBF

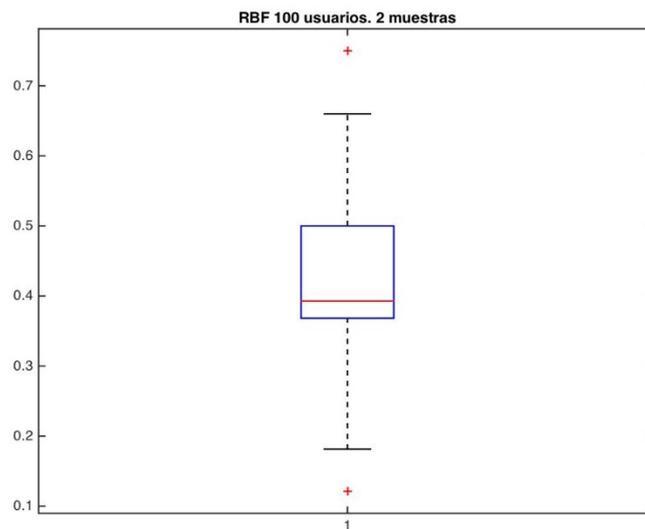


Figura B.18. Boxplot de resultados para 100 usuarios con 2 muestras entrenamiento usando *kernel* RBF

Anexo B. Boxplots

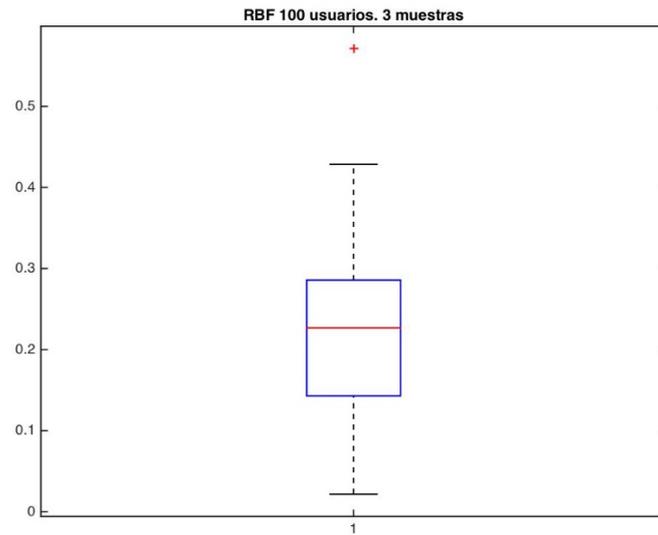


Figura B.19. Boxplot de resultados para 100 usuarios con 3 muestras entrenamiento usando *kernel* RBF

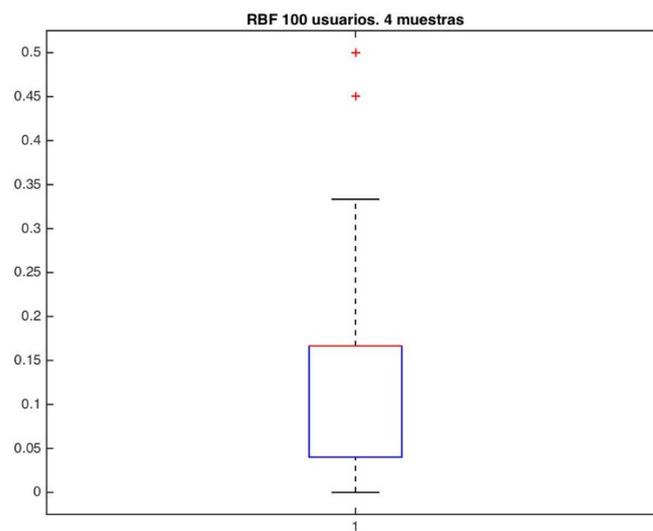


Figura B.20. Boxplot de resultados para 100 usuarios con muestras entrenamiento usando *kernel* RBF

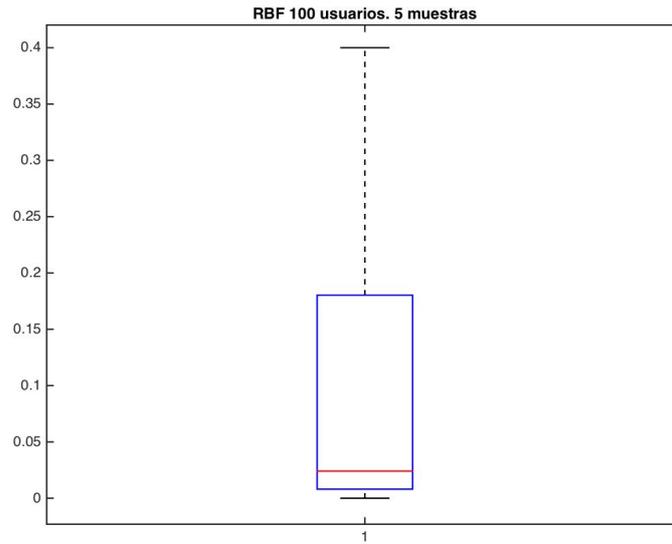


Figura B.21. Boxplot de resultados para 100 usuarios con 5 muestras entrenamiento usando kernel RBF

Anexo B. Boxplots

Bibliografía

- [1] Araña y I. A. García, «SISTEMA BIOMÉTRICO DE RECONOCIMIENTO DE FIRMAS MANUSCRITAS OFF-LINE MEDIANTE EL ANÁLISIS DEL CONTORNO,» Las Palmas de Gran Canaria, 2015.
- [2] A. Morales, «Estrategias para la Identificación de Personas mediante Biometría de la Mano sin Contacto,» Las Palmas de Gran Canaria, 2011.
- [3] P. R. Anand, G. Bajpai y V. Bhaskar, «3D Signature for Efficient Authentication in Multimodal Biometric Security Systems,» 2010.
- [4] J. Guerra-Casanova, «Propuesta, Implementación y Evaluación de la Biometría de la Firma en el Aire como Sistema de Verificación en Teléfonos Móviles,» Madrid, 2014.
- [5] E. Guerra, «Estudio de la biometría del trazo aéreo de la firma para la identificación personal,» Las Palmas de Gran Canaria, 2015.
- [6] «Secutatis Information Security,» 2017. [En línea]. Available: http://www.secutatis.com/?page_id=78#. [Último acceso: 2017].
- [7] R. Plamondon y S. N. Srihari, «On-Line and Off-Line Handwriting Recognition: A Comprehensive Survey,» 2000.
- [8] J. F. Vargas, M. A. Ferrer, C. M. Travieso y J. B. Alonso, «Offline Signature Verification Based on Pseudo-Cepstral Coefficients,» 2009.
- [9] S. J. Elliott y A. R. Hunt, «The Challenge of Forgeries and Perception of Dynamic Signature Verification,» 2006.
- [10] M. Ammar, Y. Yoshida y T. Fukumura, «A New Effective Approach for Automatic Off-Line Verification of Signatures by Using Pressure Features,» 1986.
- [11] Leap Motion, «LEAP MOTION. Developer Portal,» [En línea]. Available: <https://developer.leapmotion.com/>. [Último acceso: Enero 2016].

Bibliografía

- [12] M. Katagiri y T. Sugimura, «Personal Authentication by Free Space Signing with Video Capture,» 2002.
- [13] A. Takeuchi, Y. Manabe y K. Sugawara, «Multimodal Soft Biometric Verification by Hand Shape and Handwriting Motion in the Air,» 2013.
- [14] K. Yasuda, D. Muramatsu, S. Shirato y T. Matsumoto, «Visual-based online signature verification using features extracted from video,» 2010.
- [15] R. E. Haskell, D. M. Hanna y K. Van Sickle, «3D Signature Biometrics Using Curvature Moments,» 2006.
- [16] S. Hangai, S. Yamanaka y T. Hamamoto, «On-line Signature Verification Based on Altitude and Direction of Pen Movement,» 2000.
- [17] E. Sesa i Noguerras, «An Online Writer Recognition System Based on In-Air and On-Surface Trajectories,» 2012.
- [18] G. Bailador, C. Sanchez-Avila, J. Guerra-Casanova y A. De Santos Sierra, «Analysis of pattern recognition techniques for in-air signature biometrics,» 2011.
- [19] J. Guerra Casanova, C. Sanchez Avila, A. De Santos Sierra y C. Jimenez Suarez, «ES 2 338 092 A1,» 2010.
- [20] F. Okumura, A. Kubota, Y. Hatori, K. Matsuo, M. Hashimoto y A. Koike, «A Study on Biometric Authentication based on Arm Sweep Action with Acceleration Sensor,» 2006.
- [21] J. Liu, L. Zhong, J. Wickramasuriya y V. Vasudevan, «User Evaluation of Lightweight User Authentication with a Single Tri-Axis Accelerometer,» 2009.
- [22] S. Berman, H. Stern, O. Mendels, I. Kliche, J. Minkwitz y J. Stegmann, «EP 2 535 835 A2,» 2012.
- [23] O. Mendels, H. Stern y S. Berman, «User Identification for Home Entertainment Based on Free-Air Hand Motion Signatures,» 2014.
- [24] S. Kratz y T. I. Aumi, «AirAuth : Evaluating In-Air Hand Gestures for Authentication,» 2014.
- [25] E. Guerra, C. M. Travieso y J. B. Alonso, «Evaluation of different mathematical distances to develop the Dynamic Time Warping algorithm applied to gesture recognition,» 2015.

- [26] «eumed.net Enciclopedia Virtual,» [En línea]. Available: www.eumed.net. [Último acceso: Enero 2017].
- [27] «mailxmail.com,» [En línea]. Available: <http://www.mailxmail.com>. [Último acceso: Enero 2017].
- [28] «ITM Institución Universitaria,» [En línea]. Available: <http://itmojs.itm.edu.co>. [Último acceso: Enero 2016].
- [29] «Universidad de Pamplona,» [En línea]. Available: <http://www.unipamplona.edu.co>. [Último acceso: Enero 2016].
- [30] «Pontificia Universidad Católica de Valparaíso,» [En línea]. Available: <http://opac.pucv.cl/>. [Último acceso: Enero 2016].
- [31] «Coursera,» [En línea]. Available: <https://es.coursera.org>. [Último acceso: Marzo 2017].
- [32] J. Suykens y J. Vandewalle, «Least squares support vector machine classifiers, Neural Processing Letters,» 1999.
- [33] «Redes de Ingeniería,» [En línea]. Available: <http://revistas.udistrital.edu.co>. [Último acceso: Abril 2017].
- [34] J.-H. Jeon, B.-S. Oh y K.-A. Toh, «A System for Hand Gesture Based Signature Recognition,» 2012.

Bibliografía

PLANOS Y PROGRAMAS

En este apartado se presentan los algoritmos desarrollados en este proyecto. Primeramente, se presenta un esquema de cómo se distribuyen los algoritmos, de forma que pueda entenderse en qué fase del proyecto se ha utilizado cada uno. Seguidamente, se introducen las cabeceras de cada una de los códigos implementados, explicando sus funciones y sus parámetros básicos.

PP.1. Distribución

Aquí se presenta un pequeño esquema de la distribución de los algoritmos.

Extracción de subconjuntos con sus respectivas características

- feature_extraction.m
 - normalizar.m
- feature_extraction_Mod.m
 - normalizar.m

Fusión de subconjuntos

- Fusion.m

Clasificación

- Main.m
 - lssvm.m
 - trainlssvm.m
 - initlssvm.m
 - prelssvm.m
 - postlssvm.m
 - simlssvm.m
 - initlssvm.m
 - prelssvm.m
 - postlssvm.m

PP.2. Algoritmos para la extracción de subconjuntos con sus respectivas características

```
% Nombre: "feature_extraction.m"
%
% Definición: Divide un conjunto de señales en segmentos y extrae
% las características de éstos. En este caso, se halla
% la media, la desviación típica y la correlación.
%
% Parámetros Entrada: - SEG: Nº Segmentos en los que se quiere dividir el
% conjunto de señales.
% - USERS: Nº de usuarios que existen en la base de
% datos.
%
%
%
% Parámetros Salida: - CARACT: Un array con todos los datos extraídos.
% Lo que se definirá como subconjunto de
% datos.
%

% Nombre: "normalizar.m"
%
% Definición: Normaliza un conjunto de datos. Es decir, cada dato
% tendrá un valor entre 0 y 1.
%
% Parámetros Entrada: - ARRAY: Conjunto de datos que se quiere normalizar
%
%
%
% Parámetros Salida: - ARRAYNORM: Array con los datos normalizados
%
%
```

```

% Nombre:          "feature_extraction_Mod.m"
%
% Definición:      Divide un conjunto de señales en segmentos y extrae
%                  las características de éstos. En este caso, se halla
%                  la media, la desviación típica y la correlación y,
%                  además, se pueden hallar la asimetría, la curtosis
%                  y la entropía de Shannon según se desee.
%
% Parámetros Entrada: - SEG: Nº Segmentos en los que se quiere dividir el
%                    conjunto de señales.
%                    - USERS: Nº de usuarios que existen en la base de
%                      datos.
%                    - SKEW: Si se quisiera hallar la asimetría.
%                    - ENT: Si se quisiera hallar la entropía de Shannon.
%                    - KURT: Si se quisiera hallar la curtosis.
%
% Parámetros Salida: - CARACT: Un array con todos los datos extraídos.
%                    Lo que se definirá como subconjunto de
%                    datos.
%
%

```

PP.3. Algoritmo de fusión de subconjuntos

```

% Nombre:          "Fusion.m"
%
% Definición:      Fusiona 2 o más subconjunto de señales.
%
% Parámetros Entrada: - SUBCONJ: Subconjuntos de señales que se quieren
%                    fusionar.
%
%
% Parámetros Salida: - S_FUSION: Suconjunto de señales fusionado.
%
%

```

PP.4. Algoritmos de clasificación

```

% Nombre:          "Main.m"
%
% Definición:      Realiza la clasificación de los usuarios a partir
%                  de un subconjunto de señales que se especifica y se
%                  muestra la gráfica del EER.
%
%
% Parámetros Entrada: - SUBCONJ: Subconjuntos de señales por el que se
%                       quiere realizar la clasificación
%                       - TNUM: Nº de muestras de entrenamiento con las que
%                           se quiere clasificar.
%                       - KERNEL: Se define el kernel (Polinómico o RBF).
%
%
% Parámetros Salida: - EER: Valor del Equal Error Rate
%                       - UMBRAL: Valor del umbral
%
% Funciones llamadas:
%                   - lssvm.m
%
%
% Nombre:          "lssvm.m"
%
% Definición:      Realiza la clasificación de los usuarios a partir
%                  de un subconjunto de señales que se especifica.
%
%
% Parámetros Entrada: - SUBCONJ: Subconjuntos de señales por el que se
%                       quiere realizar la clasificación
%                       - TNUM: Nº de muestras de entrenamiento con las que
%                           se quiere clasificar.
%                       - KERNEL: Se define el kernel (Polinómico o RBF).
%
%
% Parámetros Salida: - Zt: Predicciones (valores próximos a -1 o 1).
%
%
% Funciones llamadas:
%                   - trainlssvm.m
%                   - simlssvm.m
%
%

```

```

% Nombre:          "trainlssvm.m"
%
% Definición:      Entrenar el lssvm para la posterior clasificación.
%
%
% Parámetros Entrada: - X: Firmas que se usan para el entrenamiento.
%                   - Y: Vector con las etiquetas para las firmas de
%                       entrenamiento. 1 si es el usuario correcto o
%                       -1 si no lo es.
%                   - GAM: Parámetro de regularización.
%                   - SIG2: Parámetro del kernel.
%                   - KERNEL: Se define el kernel (Polinómico o RBF).
%
% Parámetros Salida: - alpha: Matriz con valores de soporte del lssvm.
%                   - b: Vector con la/las condición/es de polarización
%                       del lssvm.
%
%
% Funciones llamadas:
%                   - initlssvm.m
%                   - prelssvm.m
%                   - postlssvm.m
%

```

```

% Nombre:          "initlssvm.m"
%
% Definición:      Iniciar la estructura orientada a objetos que
%                 representa el modelo LS-SVM.
%
%
% Parámetros Entrada: - X: Firmas que se usan para el entrenamiento.
%                   - Y: Vector con las etiquetas para las firmas de
%                       entrenamiento. 1 si es el usuario correcto o
%                       -1 si no lo es.
%                   - GAM: Parámetro de regularización.
%                   - SIG2: Parámetro del kernel.
%                   - KERNEL: Se define el kernel (Polinómico o RBF).
%
% Parámetros Salida: - MODEL: Representación orientada a objetos del
%                   modelo LS-SVM
%
%

```

```

% Nombre:          "prelssvm.m"
%
% Definición:      Preprocesamiento del LS-SVM.
%
%
% Parámetros Entrada: - MODEL: Representación orientada a objetos
%                   preprocesada del modelo LS-SVM.
%
% Parámetros Salida: - MODEL: Representación orientada a objetos del
%                   modelo LS-SVM.
%
%

```

Planos y programas

```
% Nombre: "postlssvm.m"
%
% Definición: Procesamiento posterior del LS-SVM.
%
%
% Parámetros Entrada: - MODEL: Representación orientada a objetos
%                       preprocesada del modelo LS-SVM.
%
% Parámetros Salida: - MODEL: Representación orientada a objetos del
%                     modelo LS-SVM.
%
%
%
%
% Nombre: "simlssvm.m"
%
% Definición: Evalúa el LS-SVM en los puntos dados.
%
%
% Parámetros Entrada: - X: Firmas que se usan para el entrenamiento.
%                     - Y: Vector con las etiquetas para las firmas de
%                           entrenamiento. 1 si es el usuario correcto o
%                           -1 si no lo es.
%                     - GAM: Parámetro de regularización.
%                     - SIG2: Parámetro del kernel.
%                     - KERNEL: Se define el kernel (Polinómico o RBF).
%                     - ALPHA: Matriz con valores de soporte del lssvm.
%                     - b: Vector con la/las condición/es de polarización
%                           del lssvm.
%                     -Xt: Firmas que se usan para test.
%
% Parámetros Salida: - Yt: Matriz con el resultado previsto de los datos
%                     del test.
%                     - Zt: Matriz con variables latentes previstas del
%                           clasificador.
%
% Funciones llamadas:
%   - initlssvm.m
%   - prelssvm.m
%   - postlssvm.m
%
```

PLIEGO DE CONDICIONES

En el presente apartado se presentan los requisitos software y hardware necesarios para la elaboración de este Proyecto Fin de Carrera.

PC.1. Requisitos hardware

Para el presente proyecto, se ha utilizado una serie de herramientas hardware que se indican a continuación:

- Portátil Apple (MacBook Air): NetBeans 8.1, Matlab R2015a, Microsoft Office 2015.
- Pen drive 8 GB para guardar el código realizado y ejecutarlo en el PC propiedad de la ULPGC, así como guardar los resultados obtenidos.
- Leap Motion para la obtención de las firmas para la base de datos.
- Ordenador propiedad de la ULPGC, PC ACER, ordenador de sobremesa en el que está instalado el software Matlab R2014a y en el que se han realizado las simulaciones correspondientes a los dos últimos experimentos, que son los que más tiempo de ejecución necesitan.

Las principales características de este PC son:

- Procesador Intel Core i5 a 1,6 GHz.
- 8 GB de memoria.
- Gráficos: Intel HD Graphics 6000.
- 128 GB de disco duro.

PC.2. Requisitos software

Las herramientas software que fueron necesarias son:

- Sistema operativo del ordenador portátil: OS X El Capitán versión 10.11.6.
- Sistema operativo del PC: Windows 7.
- Microsoft Office 2015 dentro del cuál se han usado:
 - Microsoft Word para redactar la memoria del PFC.
 - Microsoft Excel para crear parte de las gráficas mostradas en el PFC.
 - Microsoft Powerpoint para realizar la presentación del proyecto.
- NetBeans 8.1 para la obtención de las firmas para la base de datos.
- Matlab R2015a en el ordenador portátil. Usado para la realización de los experimentos.
- Matlab R2014a en el PC. Usado para realizar las simulaciones de los últimos experimentos.

PRESUPUESTO

P.1. Declaración jurada

D. Aysse Ortega Pérez, autor del presente Proyecto Fin de Carrera, declara que:

El proyecto Fin de Carrera con título “Estudio y desarrollo de la caracterización de firmas online mediante trazos aéreos”, realizado a petición de la Escuela de Ingeniería de Telecomunicación y Electrónica de la Universidad de Las Palmas de Gran Canaria y en un periodo de diez (40 semanas), tiene un coste de desarrollo total de **56.606,21 CINCUENTA Y SEIS MIL SEIS CIENTOS SEIS EUROS CON VEINTIÚN CÉNTIMOS**, correspondiente a la suma de las cantidades consignadas en los apartados descritos a continuación.

Firmando la presente para que así conste a los efectos oportunos.

Las Palmas de Gran Canaria, 4 de Julio de 2017

El autor del proyecto.

Aysse Ortega Pérez

Presupuesto

P.2. Desglose del presupuesto

Para el cálculo de este presupuesto se ha utilizado la base de los “*Baremos orientativos para el cálculo de honorarios*” del Colegio Oficial de Ingenieros de Telecomunicación. Éste, se divide en varias partes:

- Recursos materiales.
- Trabajo tarifado por tiempo empleado.
- Costes de redacción del proyecto.
- Material fungible.
- Derechos de visado del COIT.
- Gastos de tramitación y envío.
- Aplicación de impuestos.

P.3. Recursos materiales.

En este apartado se incluyen las herramientas hardware y software mencionadas y explicadas en el pliego de condiciones.

Se estipula el coste de amortización para un período de 3 años. Para ello, se usará un sistema de amortización lineal, en el que se supone que el material se deprecia de manera constante a lo largo del tiempo. La cuota de amortización anual se calcula con la ecuación PR.1.

$$Cuota\ anual = \frac{Valor\ de\ Adquisición - Valor\ Residual}{Años\ de\ vida\ útil} \quad (Ec.\ PR.1)$$

donde:

“Valor de Adquisición” : Coste del producto.

“Valor Residual” : Valor teórico que se supone que tendrá el elemento después de su vida útil.

P.4. Recursos software

Los recursos software que se han usado en este proyecto han sido los siguientes:

- Windows 7
- OS X El Capitán
- Microsoft Office 2015
- NetBeans 8.1
- Matlab R2014a
- Matlab R2015a

Teniendo en cuenta que la duración del presente proyecto es de 10 meses e que el cálculo del coste de amortización se establece en un período de 3 años, los costes de amortización se calcularán para el primer año. Estos costes se presentan en la siguiente tabla:

<i>Concepto</i>	<i>Coste total (€)</i>	<i>Valor residual 3 años (€)</i>	<i>Amortización 10 meses (€)</i>
<i>Windows 7</i>	40 €	0 €	11,11 €
<i>OS X El Capitán</i>	0 €	0 €	0 €
<i>MS Office®2015</i>	20 €	0 €	5,56 €
<i>NetBeans 8.1</i>	0 €	0 €	0 €
<i>Matlab R2014a</i>	2.000 €	0 €	555,56 €
<i>Matlab R2015a</i>	2.000 €	0 €	555,56 €
<i>Total amortización del software</i>			1.127,79 €

Tabla PR.1. Costes de los recursos software

Presupuesto

P.5. Recursos hardware

Los recursos hardware empleados en este proyecto han sido:

- Ordenador **portátil** Apple (MacBook Air) con procesador Intel Core i5 a 1,6 GHz. 8 GB de memoria. 120 GB de disco duro.
- Ordenador propiedad de la ULPGC, **PC Acer**, con procesador Intel Core i5 a 1,6 GHz. 8 GB de memoria. 128 GB de disco duro.
- Sensor comercial **Leap Motion**.

<i>Concepto</i>	<i>Coste total (€)</i>	<i>Valor residual 3 años (€)</i>	<i>Amortización 10 meses (€)</i>
<i>Portátil</i>	1.100 €	250 €	236,11 €
<i>PC</i>	800 €	105 €	193,06 €
<i>Leap Motion</i>	80 €	30 €	13,89 €
<i>Total amortización del software</i>			443,06 €

Tabla PR.2. Costes de los recursos hardware

P.6. Trabajo tarifado por tiempo empleado

Se han invertido 10 meses en las tareas de formación, desarrollo y documentación necesarias para la elaboración de este Proyecto de Fin de Carrera. El importe de las horas de trabajo empleadas para la realización del proyecto se calcula siguiendo las recomendaciones del COIT:

$$H = C_t \cdot H_n \cdot 74,88 + C_t \cdot H_e \cdot 96,72 \quad (\text{Ec. PR.2})$$

donde:

H son los honorarios.

H_n son las horas en jornada normal.

H_e son las horas fuera de jornada normal.

C_t es el factor de corrección en función del número de horas trabajadas.

Se estima que para la realización de este proyecto se han empleado 1.600 horas, todas ellas dentro del horario normal (8 horas al día · 5 días a la semana · 4 semanas al mes · 10 meses).

Presupuesto

Según el COIT el valor del coeficiente de corrección (C_t) respecto al tiempo necesario para realizar el proyecto viene reflejado en la siguiente tabla:

<i>HORAS TRABAJADAS</i>	<i>FACTOR DE CORRECCIÓN C_t</i>
<i>Hasta 36 horas</i>	1,0
<i>De 36 a 72 horas</i>	0,90
<i>De 72 a 108 horas</i>	0,80
<i>De 108 a 144 horas</i>	0,70
<i>De 144 a 180 horas</i>	0,65
<i>De 180 a 360 horas</i>	0,60
<i>De 360 a 540 horas</i>	0,55
<i>De 540 a 720 horas</i>	0,50
<i>De 720 a 1080 horas</i>	0,45
<i>Más de 1080 horas</i>	0,40

Tabla PR.3. Factor de corrección del COIT según las horas trabajadas.

Según la tabla, el factor de corrección que se usaría en la ecuación PR.2 es $C_t = 0,40$ puesto que, el tiempo empleado en la realización del presente proyecto superan las 1080 horas. Por lo tanto, se tiene lo siguiente:

$$H = 0,40 \cdot 1600 \cdot 74,88 + 0,40 \cdot 0 \cdot 96,72 = 47.923,20\text{€}$$

El coste de ingeniería, libre de impuestos, asciende a un total de *cuarenta y siete mil novecientos veintitrés* euros con veinte céntimos (47.923,20 €)

P.7. Costes de redacción

El importe de la redacción del proyecto se calcula haciendo uso de la ecuación PR.3, establecida por el COIT y que se muestra a continuación.:

$$R = 0,07 \cdot P \cdot C_h \quad (\text{Ec. PR.3})$$

donde:

P es el presupuesto del proyecto.

C_h es el coeficiente de ponderación en función del presupuesto.

<i>Concepto</i>	<i>Amortización (€)</i>
<i>Recursos Software</i>	1.127,79 €
<i>Recursos Hardware</i>	443,06 €
<i>Trabajo Tarifado por tiempo empleado</i>	47.923,20 €
<i>Total</i>	49.494,05 €

Tabla PR.4. Presupuesto: Software + Hardware + Tiempo empleado.

Hasta el momento, el presupuesto calculado (P) asciende a 49.494,05 € como se detalla en la tabla PR.4.

Según el COIT, el coeficiente de ponderación (C_h) para presupuestos que se encuentran entre los 30.050 € y los 90.150 € queda fijado en 0,9. Entonces, el coste derivado de la redacción es de:

$$R = 0,07 \cdot 49494,05 \cdot 0,9 = 3.118,13 \text{ €}$$

Por tanto, el coste libre de impuestos derivados de la redacción del proyecto es de *tres mil ciento dieciocho euros con trece céntimos (3.118,13 €)*.

Presupuesto

P.8. Material fungible

Además de los recursos software y hardware, también hay que tener en cuenta el coste del material fungible. Este material fungible corresponde con materiales como son los folios, el tóner de la impresora, etc. En la siguiente tabla se muestran los costes correspondientes a estos materiales.

<i>Concepto</i>	<i>Amortización (€)</i>
<i>Tóner impresora</i>	100 €
<i>Folios</i>	10 €
<i>Encuadernación</i>	20 €
<i>Total</i>	130 €

Tabla PR.5. Costes del material fungible.

P.9. Derechos de visado del COIT

Los gastos de visado del COIT se tarifican mediante la siguiente expresión:

$$V = 0,006 \cdot P \cdot C_v \quad (\text{Ec. PR.4})$$

donde:

P es el presupuesto del proyecto.

C_v es el coeficiente reductor en función del presupuesto del proyecto.

El presupuesto calculado hasta el momento, teniendo en cuenta ahora además los costes de redacción y de material fungible, asciende a:

$$P = 49.494,05 + 3.118,13 + 130 = 52.742,18 \text{ €}$$

Como el coeficiente C_v para presupuestos de más de 30.050 € y menos de 60.101 €, viene definido por el COIT con un valor de 0,90, el coste de los derechos de visado del proyecto asciende a la cantidad de:

$$V = 0,006 \cdot 52.742,18 \cdot 0,90 = 284,81 \text{ €}$$

Por lo tanto, el coste de los derechos de visado del proyecto asciende a *doscientos ochenta y cuatro euros con ochenta y un céntimos (284,81 €)*.

P.10. Gastos de tramitación y envío

Los gastos de tramitación y envío son fijos y se estipulan en *seis euros y un céntimo (6,01 €)*.

P.11. Aplicación de impuestos

El coste total del proyecto, antes de aplicarle los correspondientes impuestos, asciende a 52.903 €, a lo que hay que sumarle el 7% de IGIC, con lo que el coste definitivo del proyecto queda desglosado y especificado en la siguiente tabla:

Coste total del proyecto

<i>Coste por tiempo empleado</i>	47.923,20 €
<i>Coste del software</i>	1.127,79 €
<i>Coste del hardware</i>	443,06 €
<i>Coste redacción del proyecto</i>	3.118,13 €
<i>Derechos de visado del COIT</i>	284,81 €
<i>Gastos de tramitación y envío</i>	6,01 €
<i>Total antes de impuestos</i>	52.903 €
<i>IGIC (7%)</i>	3.703,21 €
<i>Total</i>	56.606,21 €

Tabla PR.6. Tabla final de costes con la aplicación de impuestos.

Presupuesto

El presupuesto total, con los impuestos incluidos, asciende a la cantidad de cincuenta y seis mil seiscientos seis euros con veintiún céntimos (56.606,21 €).

Las Palmas de Gran Canaria, 4 de Julio de 2017

El autor del proyecto.

Aysse Ortega Pérez