



PROYECTO DE GRADO

Presentado ante la ilustre UNIVERSIDAD DE LOS ANDES como requisito parcial para obtener el Título de INGENIERO DE SISTEMAS

MÓDULO RECONFIGURABLE DE RECONOCIMIENTO PARA LA INTERACCIÓN HUMANO-ROBOT

Por

Br. Nerio Morán

Tutor: Prof. Jesús Pérez

Febrero 2019

©2019 Universidad de Los Andes, Mérida, Venezuela

PROYECTO DE GRADO
CALIFICACIÓN FINAL

Título del Proyecto de Grado: "Módulo Reconfigurable de Reconocimiento para la Interacción Humano-Robot"

Bachiller: NERIO JOSÉ MORÁN PÁEZ

C.I: 23.556.790

1) Calificación del (de la) Profesor(a) Tutor(a)	20	(20%)
2) Calificación del manuscrito final:	20	(40%)
3) Calificación de la defensa oral:	20	(40%)
Calificación final:	20	(puntos)

Los suscritos miembros del Jurado asignan como calificación final del proyecto de Grado la nota de:

20 (puntos)

Veinte (puntos)

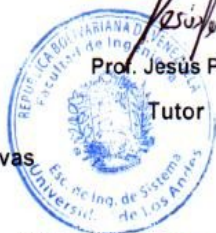
Numeros

Letras

Observaciones: Los miembros del jurado calificador sugerimos que el presente trabajo sea accesible a la comunidad universitaria en general, a través de la publicación de un artículo científico, donde se destaquen los principales resultados de este trabajo. En consecuencia le otorgamos MENCION PUBLICACION.

~~Prof. Rafael Rivas~~

~~Jurado~~



Prof. Jesús Pérez

Tutor

Prof. Junior Altamiranda

Jurado

Mérida, 08 de Febrero del 2019.

Módulo Reconfigurable de Reconocimiento para la Interacción Humano-Robot

Br. Nerio Morán

Proyecto de Grado — Sistemas Computacionales, 229 páginas
Escuela de Ingeniería de Sistemas, Universidad de Los Andes, 2019

Resumen: La interacción humano-robot tiene como propósito facilitar el intercambio de información entre personas y robots. Recientemente, se ha notado un interés por adaptar las interacciones de los robots según la persona con quien interactúa. En este sentido, las tareas de reconocimiento requeridas por los robots en un momento determinado, también varían según la persona. En general, la capacidad de adaptarse a diferentes personas requiere de una gran cantidad de tareas de reconocimiento, lo cual es un inconveniente para el limitado almacenamiento y procesamiento interno de los robots. Por lo tanto, en este Proyecto de Grado, se implementa un Módulo Reconfigurable de Reconocimiento orientado a la gestión de tareas y paquetes de reconocimiento de manera local y remota, que además ofrece la capacidad de solicitar servicios de reconocimiento a un robot en particular. El caso de estudio para verificar el funcionamiento del Módulo Reconfigurable incluye la implementación de dos tareas de reconocimiento para determinar quién es la persona y qué emoción tiene. Estas tareas de reconocimiento se implementan con algoritmos de aprendizaje de máquina, por lo tanto, debido a la ausencia de bases de datos en español, este Proyecto de Grado incluye la construcción de conjuntos de datos de entrenamiento y validación, y la determinación de los algoritmos de aprendizaje apropiados para realizar estas tareas de reconocimiento.

Palabras clave: Interacción humano-robot, reconfiguración, aprendizaje de máquina, reconocimiento.

Este trabajo fue procesado en L^AT_EX.

Índice general

Índice de Tablas	XII
Índice de Figuras	XVII
Introducción	XXII
Agradecimientos	XXIV
1. Contextualización	1
1.1. Antecedentes	2
1.1.1. Módulos de reconocimiento	2
1.1.2. Reconocimiento de personas mediante la voz	7
1.1.3. Reconocimiento de emociones mediante la voz	10
1.1.4. Bases de datos orientadas al reconocimiento de personas mediante la voz	12
1.1.5. Bases de datos orientadas al reconocimiento de emociones mediante la voz	14
1.2. Planteamiento del problema	17
1.3. Justificación	18
1.4. Objetivos	19
1.4.1. Objetivo general	19
1.4.2. Objetivos específicos	19
1.5. Metodología	20
1.5.1. Fase de diagnóstico	20
1.5.2. Fase de diseño	20

1.5.3.	Fase de implementación	20
1.5.4.	Fase de pruebas	21
1.6.	Alcance	21
2.	Marco Teórico	23
2.1.	Sistema operativo robótico (ROS)	24
2.1.1.	Distribuciones de ROS	25
2.1.2.	Beneficios de ROS	26
2.1.3.	Niveles conceptuales de ROS	27
2.1.3.1.	Nivel de sistema de archivos	27
2.1.3.2.	Nivel del grafo computacional	28
2.1.3.3.	Nivel de la comunidad	30
2.2.	Aprendizaje de máquina	30
2.2.1.	Máquina de vectores de soporte	31
2.2.1.1.	Núcleos	33
2.2.2.	Modelo de mezclas Gaussianas	34
2.2.2.1.	Distribución Gaussiana	34
2.2.2.2.	Modelo de mezclas Gaussianas (GMM) y el algoritmo K-means	36
2.2.2.3.	Clasificación mediante el modelo de mezclas Gaussianas	38
2.2.2.4.	Algoritmo Esperanza-Maximización	39
2.2.3.	Árboles de decisión	41
2.2.3.1.	Decisiones binarias	42
2.2.4.	Aprendizaje combinado	44
2.2.4.1.	Clasificación basada en votación	45
2.2.4.2.	Bosques aleatorios	46
2.2.4.3.	Aumento del gradiente	46
2.3.	Procesamiento de señales de audio	47
2.3.0.1.	Parámetros de las señales de audio	48
2.3.0.2.	Procesamiento a corto plazo de las señales de audio . .	49
2.3.1.	Características del audio	52

3. Análisis de requisitos	56
3.1. Descripción general del Módulo Reconfigurable de Reconocimiento	57
3.1.1. Funciones del Módulo Reconfigurable de Reconocimiento	58
3.1.1.1. Gestor local	58
3.1.1.2. Gestor remoto	59
3.1.1.3. Servicios de reconocimiento	60
3.1.2. Interfaces	61
3.1.3. Requisitos funcionales	61
3.1.3.1. Gestión de tareas de reconocimiento local y remota . .	62
3.1.3.2. Gestión de paquetes de reconocimiento local y remoto	65
3.1.3.3. Servicios de reconocimiento local y remoto	68
3.1.4. Requisitos no funcionales	70
3.1.4.1. Requisitos de ejecución	70
3.1.4.2. Restricciones de diseño	70
3.1.4.3. Otros requisitos	71
4. Diseño	75
4.1. Diseño del Módulo Reconfigurable de Reconocimiento	76
4.1.1. Casos de uso del Módulo Reconfigurable de Reconocimiento . .	77
4.1.2. Implementación del Módulo Reconfigurable de Reconocimiento .	78
4.1.2.1. Implementación del gestor local	78
4.1.2.2. Implementación del gestor remoto	79
4.1.2.3. Implementación de los servicios de reconocimiento . . .	79
4.2. Diseño de la base de datos para el reconocimiento de emociones	84
4.2.1. Bases de datos en condiciones controladas y actuadas	85
4.2.2. Bases de datos en condiciones controladas y semi-natural	89
4.3. Diseño de la base de datos para el reconocimiento de personas	91
4.3.1. Construcción de la base de datos	92
4.3.2. Diseño y proceso de grabación	92
4.3.3. Diseño del conjunto de frases y palabras de entrenamiento y prueba	93
4.4. Diseño del reconocedor de emociones mediante la voz	102
4.4.1. Selección del algoritmo	103

4.4.2.	Implementación	103
4.4.3.	Optimización	105
4.5.	Diseño del reconocedor de personas mediante la voz	108
4.5.1.	Selección del algoritmo	109
4.5.2.	Implementación	109
4.5.3.	Optimización	112
5.	Pruebas	113
5.1.	Pruebas sobre el Módulo Reconfigurable de Reconocimiento	114
5.1.1.	Planificación de pruebas	114
5.1.1.1.	Estrategias utilizadas en las pruebas	114
5.1.1.2.	Límites y suposiciones	115
5.1.2.	Especificación del diseño de las pruebas	116
5.1.2.1.	Pruebas del gestor local	116
5.1.2.2.	Pruebas del gestor remoto	116
5.1.2.3.	Pruebas sobre los servicios de reconocimiento	117
5.1.2.4.	Medición de memoria y procesamiento	117
5.1.2.5.	Identificación de casos de prueba	117
5.1.3.	Especificación de los casos de prueba	119
5.1.4.	Medición de memoria y procesamiento de uno de los gestores	136
5.1.5.	Medición de memoria y procesamiento de una de las tareas de reconocimiento	137
5.2.	Pruebas sobre el reconocedor de emociones	138
5.2.1.	Prueba 1: validación cruzada	140
5.2.1.1.	Comparativa de los algoritmos: Prueba estadística	141
5.2.2.	Prueba 2: pruebas de retención	142
5.2.2.1.	Resultados: algoritmo SVM	143
5.2.2.2.	Resultados: algoritmo RF	144
5.2.2.3.	Resultados: algoritmo GB	145
5.2.2.4.	Resultados: algoritmo EL	146
5.2.3.	Prueba 3: Cálculo de métricas	147
5.2.3.1.	Precisión, sensibilidad y puntaje F1	147

5.2.3.2.	Curvas ROC	150
5.2.4.	Análisis de los resultados	153
5.3.	Pruebas sobre el reconocedor de personas	155
5.3.1.	Pruebas de identificación de personas	156
5.3.1.1.	Prueba: P1-A	157
5.3.1.2.	Prueba: P1-B	158
5.3.1.3.	Prueba: P2-A	159
5.3.1.4.	Prueba: P2-B	160
5.3.2.	Pruebas de verificación de personas	162
5.3.2.1.	Prueba: P3-A y P3-B	163
5.3.2.2.	Prueba: P4-A y P4-B	163
5.3.3.	Análisis de los resultados	164
6.	Conclusiones y recomendaciones	166
6.1.	Conclusiones	167
6.2.	Aportes	169
6.3.	Recomendaciones	170
6.4.	Trabajos Futuros	171
A.	Documento de Especificación de Requisitos	172
A.1.	Introducción	172
A.1.1.	Objetivo	172
A.1.2.	Alcance	172
A.1.3.	Definiciones, acrónimos y abreviaciones	173
A.1.4.	Referencias	173
A.1.5.	Visión general	173
A.2.	Descripción general	174
A.2.1.	Perspectiva del producto	174
A.2.2.	Funciones del producto	175
A.2.3.	Características de los usuarios	177
A.2.4.	Limitaciones generales	177
A.2.5.	Supuestos y dependencias	178

A.3.5.2.3.	CU-08: Eliminar un paquete de reconocimiento en el gestor local	192
A.3.5.2.4.	CU-09: Eliminar una lista de paquetes de reconocimiento en el gestor local	193
A.3.5.2.5.	CU-10: Solicitar la dirección de un paquete . .	194
A.3.5.2.6.	Diagrama de secuencia CU-06: Instalar un paquete de reconocimiento en el gestor local .	195
A.3.5.2.7.	Diagrama de secuencia CU-07: Instalar una lista de paquetes de reconocimiento en el gestor local	196
A.3.5.2.8.	Diagrama de secuencia CU-08: Eliminar un paquete de reconocimiento del gestor local . .	197
A.3.5.2.9.	Diagrama de secuencia CU-09: Eliminar una lista de paquetes de reconocimiento del gestor remoto	197
A.3.5.3.	Gestión de tareas de reconocimiento remoto	198
A.3.5.3.1.	CU-11: Activar tarea de reconocimiento remoto	199
A.3.5.3.2.	CU-12: Activar lista de tareas de reconocimiento remoto	200
A.3.5.3.3.	CU-13: Desactivar tarea de reconocimiento remoto	201
A.3.5.3.4.	CU-14: Desactivar lista de tareas de reconocimiento remoto	202
A.3.5.3.5.	CU-15: Obtener lista de tareas de reconocimiento remoto	203
A.3.5.3.6.	Diagrama de secuencia CU-11: Activar tarea de reconocimiento remoto	204
A.3.5.3.7.	Diagrama de secuencia CU-12: Activar lista de tareas de reconocimiento remoto	204
A.3.5.3.8.	Diagrama de secuencia CU-13: Desactivar tarea de reconocimiento remoto	205

A.3.5.3.9.	Diagrama de secuencia CU-14: Desactivar lista de tareas de reconocimiento remoto	205
A.3.5.3.10.	Diagrama de secuencia CU-15: Obtener lista de tareas de reconocimiento remoto	206
A.3.5.4.	Gestión de paquetes de reconocimiento remoto	207
A.3.5.4.1.	CU-16: Instalar un paquete de reconocimiento en el gestor remoto	208
A.3.5.4.2.	CU-17: Eliminar un paquete de reconocimiento en el gestor remoto	209
A.3.5.4.3.	Diagrama de secuencia CU-16: Instalar un paquete de reconocimiento remoto	210
A.3.5.4.4.	Diagrama de secuencia CU-17: Eliminar un paquete de reconocimiento del gestor remoto .	211
A.3.5.5.	Servicios de reconocimiento	212
A.3.5.5.1.	CU-18: Solicitar servicio de reconocimiento local	213
A.3.5.5.2.	CU-19: Solicitar un servicio de reconocimiento remoto	214
A.3.5.5.3.	Diagrama de secuencia CU-18: Solicitar servicio de reconocimiento local	215
A.3.5.5.4.	Diagrama de secuencia CU-19: Solicitar servicio de reconocimiento remoto	216
A.3.6.	Requisitos de ejecución	217
A.3.7.	Restricciones de diseño	217
A.3.8.	Otros Requisitos	218

Índice de Tablas

2.1. Distribuciones de ROS desde el 31 de diciembre del 2012	25
3.1. Formato de la tabla de descripción de casos de uso	62
4.1. Casos de usos relacionados con el gestor local	77
4.2. Casos de usos relacionados con el gestor remoto	77
4.3. Casos de usos relacionados con los servicios de reconocimiento	78
4.4. Conjunto de declaraciones de cada emoción	86
4.5. Escala utilizada para la validación de las muestras de audio	88
4.6. Frases y palabras del conjunto de entrenamiento basadas en los fonemas del sistema vocálico del español	94
4.7. Frases y palabras de los fonemas del sistema vocálico del español: vocales nasalizadas	94
4.8. Frases y palabras de los fonemas del sistema consonántico del español: realizaciones de los fonemas /p/ y /b/	94
4.9. Frases y palabras de los fonemas del sistema consonántico del español: neutralización de /p/ y /b/, realizaciones del archifonema /B/	94
4.10. Frases y palabras de los fonemas del sistema consonántico del español: Elisión de la d	94
4.11. Frases y palabras de los fonemas del sistema consonántico del español: realizaciones de los fonemas /t/ y /d/	95
4.12. Frases y palabras de los fonemas del sistema consonántico del español: neutralización de /t/ y /d/, realizaciones del archifonema /D/	95
4.13. Frases y palabras de los fonemas del sistema consonántico del español: realizaciones de los fonemas /k/ y /g/	95

4.14. Frases y palabras de los fonemas del sistema consonántico del español: neutralización de /k/ y /g/, realizaciones del archifonema /G/	95
4.15. Frases y palabras de los fonemas del sistema consonántico del español: neutralización de las oposiciones /p ~ b/, /t ~ d/ y /k ~ g/, realizaciones de los archifonemas /B/ /D/ /G/	95
4.16. Frases y palabras de los fonemas del sistema consonántico del español: realizaciones de los fonema /tʃ/, /f/ y /s/	96
4.17. Frases y palabras de los fonemas del sistema consonántico del español: tipos de /s/	96
4.18. Frases y palabras de los fonemas del sistema consonántico del español: la grafía X, archifonema /Gs/	96
4.19. Frases y palabras de los fonemas del sistema consonántico del español: realizaciones del fonema /j/	96
4.20. Frases y palabras de los fonemas del sistema consonántico del español: realizaciones del fonema /h/	96
4.21. Frases y palabras de los fonemas del sistema consonántico del español: realizaciones del fonema /m/, /n/ y /ɲ	97
4.22. Frases y palabras de los fonemas del sistema consonántico del español: Asimilación general	97
4.23. Frases y palabras de los fonemas del sistema consonántico del español: Asimilación parcial	98
4.24. Frases y palabras de los fonemas del sistema consonántico del español: realizaciones del fonema /l/	98
4.25. Frases y palabras de los fonemas del sistema consonántico del español: realizaciones de los fonemas /r/ y /r/, casos especiales	98
4.26. Frases y palabras del conjunto de prueba: vocales orales	99
4.27. Frases y palabras del conjunto de prueba: semi-consonantes	99
4.28. Frases y palabras del conjunto de prueba: semi-vocales	99
4.29. Frases y palabras del conjunto de prueba: consonantes oclusivas	99
4.30. Frases y palabras del conjunto de prueba: fricativas	100
4.31. Frases y palabras del conjunto de prueba: aproximantes	100

4.32. Frases y palabras del conjunto de prueba: africadas	100
4.33. Frases y palabras del conjunto de prueba: consonantes nasales	100
4.34. Frases y palabras del conjunto de prueba: alófonos nasales ante consonante	100
4.35. Frases y palabras del conjunto de prueba: consonantes líquidas laterales	101
4.36. Frases y palabras del conjunto de prueba: consonantes líquidas laterales	101
4.37. Frases y palabras del conjunto de prueba: hiatos	101
5.1. Casos de pruebas unitarias del gestor local	117
5.2. Casos de pruebas de integración del gestor local	118
5.3. Casos de pruebas unitarias del gestor remoto	118
5.4. Casos de pruebas de integración del gestor remoto	118
5.5. Casos de pruebas de integración de los servicios de reconocimiento . . .	118
5.6. Casos de pruebas sobre el sistema	118
5.7. Caso de prueba: P-GL-01	119
5.8. Caso de prueba: P-GL-02	119
5.9. Caso de prueba: P-GL-03	120
5.10. Caso de prueba: P-GL-04	120
5.11. Caso de prueba: P-GL-05	121
5.12. Caso de prueba: P-GL-06	121
5.13. Caso de prueba: P-GL-07	122
5.14. Caso de prueba: P-GL-08	122
5.15. Caso de prueba: P-GL-09	123
5.16. Caso de prueba: P-GL-10	123
5.17. Caso de prueba: P-GR-01	124
5.18. Caso de prueba: P-GR-02	124
5.19. Caso de prueba: P-GR-03	125
5.20. Caso de prueba: P-GR-04	125
5.21. Caso de prueba: P-GR-05	126
5.22. Caso de prueba: P-GR-06	126
5.23. Caso de prueba: P-GR-07	127
5.24. Caso de prueba: P-SR-01	127
5.25. Caso de prueba: P-SR-02	128

5.26. Caso de prueba: P-SIS-01	129
5.27. Caso de prueba: P-SIS-02	131
5.28. Caso de prueba: P-SIS-03	133
5.29. Exactitud de cada uno de los segmentos de prueba de la validación cruzada, $K = 10$	140
5.30. Promedio y desviación estándar de la exactitud de cada uno de los segmentos de prueba de la validación cruzada, $K = 10$	141
5.31. Pruebas de hipótesis de Wilcoxon: Valores de las pruebas (pv) y conclusiones	142
5.32. Exactitud o tasa de reconocimiento de cada uno de los modelos.	147
5.33. Métricas obtenidas utilizando el algoritmo SVM	147
5.34. Métricas obtenidas utilizando el algoritmo RF	148
5.35. Métricas obtenidas utilizando el algoritmo GB	148
5.36. Métricas obtenidas utilizando la técnica basada en aprendizaje combinado (EL)	149
5.37. Macro promedio y micro promedio del área bajo la curva ROC	152
5.38. Tasa de correcta identificación, utilizando 16 componentes con sesión única y sesión múltiple para los usuarios 01_M, 02_F, 03_M, 04_M, 05_F	161
5.39. Tasa de correcta identificación, utilizando 32 componentes con sesión única y sesión múltiple para los usuarios 01_M, 02_F, 03_M, 04_M, 05_F	161
5.40. Tasas (FAR), (FRR) y (SAR) de los usuarios, utilizando 16 componentes sesión única y sesión múltiple para los usuarios 01_M, 02_F, 03_M, 04_M, 05_F.	163
5.41. Tasas (FAR), (FRR) y (SAR) de los usuarios, utilizando 32 componentes sesión única y sesión múltiple para los usuarios 01_M, 02_F, 03_M, 04_M, 05_F.	163
A.1. Descripción del Caso de Uso CU-01	181
A.2. Descripción del Caso de Uso CU-02	182
A.3. Descripción del Caso de Uso CU-03	183
A.4. Descripción del Caso de Uso CU-04	184
A.5. Descripción del Caso de Uso CU-05	185

A.6. Descripción del Caso de Uso CU-06	190
A.7. Descripción del Caso de Uso CU-07	191
A.8. Descripción del Caso de Uso CU-08	192
A.9. Descripción del Caso de Uso CU-09	193
A.10.Descripción del Caso de Uso CU-10	194
A.11.Descripción del Caso de Uso CU-11	199
A.12.Descripción del Caso de Uso CU-12	200
A.13.Descripción del Caso de Uso CU-13	201
A.14.Descripción del Caso de Uso CU-14	202
A.15.Descripción del Caso de Uso CU-15	203
A.16.Descripción del Caso de Uso CU-16	208
A.17.Descripción del Caso de Uso CU-17	209
A.18.Descripción del Caso de Uso CU-18	213
A.19.Descripción del Caso de Uso CU-19	214

Índice de Figuras

2.1. Mecanismo de publicación y suscripción de tópicos en ROS	25
2.2. Diagrama del nivel de sistemas de archivos de ROS [45]	28
2.3. Diagrama del grafo computacional de ROS [45]	29
2.4. Elementos del algoritmo máquina de vectores de soporte	33
2.5. Distribución Gaussiana o Normal unidimensional, $\mu = 0$ y $\sigma^2 = 1$	35
2.6. Distribución Gaussiana o Normal bivariada, $\mu = (1, 2)^T$ y $\Sigma =$ $\{(0, 25, 0), (0, 25, 0)\}$	36
2.7. Diagrama del algoritmo K-means: 2 agrupamientos	37
2.8. Diagrama: datos solapados	38
2.9. Estructura del árbol de decisión: clasificación de emociones basado en la energía y los dos Coeficientes de Mel	41
2.10. Estructura del árbol de decisión: característica y límite de un nodo	42
2.11. (A) Señal cuasi-periódica	48
2.12. (B) Señal aperiódica	48
2.13. Clasificación de las señales según sus patrones: aperiódica y cuasi-periódica	48
2.14. Análisis a corto plazo del audio: bloques de marcos	49
2.15. Procesamiento a largo plazo o suprasegmental del audio	51
2.16. Función de distorsión de frecuencias f_w	54
3.1. Diagrama de bloques del gestor local del Módulo Reconfigurable de Reconocimiento.	58
3.2. Diagrama de bloques del gestor remoto del Módulo Reconfigurable de Reconocimiento	59
3.3. Diagrama de bloques del Módulo Reconfigurable de Reconocimiento	60

3.4. Diagrama de caso de uso: Gestión de tareas de reconocimiento local . . .	63
3.5. Diagrama de secuencia CU-01: Activar tarea de reconocimiento local . . .	64
3.6. Diagrama de secuencia CU-03: Desactivar tarea de reconocimiento local . . .	64
3.7. Diagrama de caso de uso: Gestión de paquetes de reconocimiento local . . .	65
3.8. Diagrama de secuencia CU-06: Instalar un paquete de reconocimiento en el gestor local	66
3.9. Diagrama de secuencia CU-08: Eliminar un paquete de reconocimiento en el gestor local	67
3.10. Diagrama de caso de uso: Solicitud de servicios de reconocimiento . . .	68
3.11. Diagrama de secuencia CU-19: Solicitar servicio de reconocimiento remoto. . .	69
3.12. Árbol de directorios de un paquete de reconocimiento.	71
3.13. Paquetes de reconocimiento para los casos de estudio: reconocedor de emociones y reconocimiento de personas	73
3.14. Clase abstracta: Task	74
3.15. Ejemplo del archivo Dependencies.txt , generado mediante el comando “pip freeze”	74
3.16. Ejemplo del archivo ModelSpecs.txt	74
4.1. Diagrama de bloques del Módulo Reconfigurable de Reconocimiento . . .	76
4.2. Servicios de reconocimiento: clase correspondiente a la solicitud.	80
4.3. Servicios de reconocimiento: clase correspondiente a la respuesta.	81
4.4. Servicios de reconocimiento: envío de solicitud de reconocimiento	82
4.5. Servicios de reconocimiento: recepción de respuestas de reconocimiento	83
4.6. Frecuencia de las emociones de la base de datos en condiciones controladas y actuadas	89
4.7. Frecuencia de las emociones de la base de datos en condiciones controladas y semi-naturales	90
4.8. Diagrama de bloques: procesos de entrenamiento y pruebas	104
4.9. Diagrama de bloques: proceso de inscripción e identificación	110
4.10. Diagrama de bloques: sistema de verificación basado en radio de probabilidad [8]	112

5.1. Diagrama de nodos y tópicos de la prueba P-SIS-01: Nodo inicial, gestor local.	130
5.2. Diagrama de nodos y tópicos de la prueba P-SIS-01: Activación de las tareas SpeechEmotionRecognition y SpeechSpeakerIdentification del gestor local.	130
5.3. Diagrama de nodos y tópicos de la prueba P-SIS-01: Solicitudes de servicios reconocimiento a la tarea de reconocimiento SpeechEmotionRecognition.	130
5.4. Diagrama de nodos y tópicos de la prueba P-SIS-01: Solicitudes y respuestas de reconocimiento de las tareas SpeechEmotionRecognition y SpeechSpeakerIdetification	130
5.5. Diagrama de nodos y tópicos de la prueba P-SIS-02: Nodo inicial, gestor remoto.	132
5.6. Diagrama de nodos y tópicos de la prueba P-SIS-02: Activación de las tareas SpeechEmotionRecognition y SpeechSpeakerIdentification del gestor remoto.	132
5.7. Diagrama de nodos y tópicos de la prueba P-SIS-02: Solicitudes de servicios de reconocimiento a la tarea de reconocimiento SpeechEmotionRecognition.	132
5.8. Diagrama de nodos y tópicos de la prueba P-SIS-02: Solicitudes y respuestas de reconocimiento de las tareas SpeechEmotionRecognition y SpeechSpeakerIdetification	132
5.9. Diagrama de nodos y tópicos de la prueba P-SIS-03: Nodos iniciales, gestor local y gestor remoto.	134
5.10. Diagrama de nodos y tópicos de la prueba P-SIS-03: Activación de las tareas SpeechEmotionRecognition y SpeechSpeakerIdentification del gestor remoto.	134
5.11. Diagrama de nodos y tópicos de la prueba P-SIS-03: Solicitudes de servicios de reconocimiento a las tareas de reconocimiento SpeechEmotionRecognition y SpeechSpeakerIdentification.	134

5.12. Diagrama de nodos y tópicos de la prueba P-SIS-03: Instalación y activación de la tarea de reconocimiento SpeechGenreRecognition. . . .	135
5.13. Diagrama de nodos y tópicos de la prueba P-SIS-03: Solicitudes de reconocimiento de las tareas SpeechEmotionRecognition, SpeechSpeakerIdetification y SpeechGenreRecognition.	135
5.14. Diagrama de nodos y tópicos de la prueba P-SIS-03: Solicitudes y respuestas de reconocimiento de las tareas SpeechEmotionRecognition, SpeechSpeakerIdetification y SpeechGenreRecognition.	135
5.15. Consumo de procesamiento y memoria sobre la prueba P-SIS-01 utilizando el software psrecord	136
5.16. Consumo de procesamiento y memoria sobre la prueba P-SIS-01 utilizando el software psrecord	137
5.17. Prueba de retención utilizando el algoritmo SVM: tasa de reconocimiento: 67 %	143
5.18. Prueba de retención utilizando el algoritmo RF: tasa de reconocimiento: 70 %	144
5.19. Prueba de retención utilizando el algoritmo GB: tasa de reconocimiento: 70 %	145
5.20. Prueba de retención utilizando el algoritmo EL: tasa de reconocimiento: 73 %	146
5.21. Áreas bajo la curva ROC de cada emoción utilizando el algoritmo SVM	150
5.22. Áreas bajo la curva ROC de cada emoción utilizando el algoritmo RF .	150
5.23. Áreas bajo la curva ROC de cada emoción utilizando el algoritmo GB .	151
5.24. Áreas bajo la curva ROC de cada emoción utilizando la técnica de aprendizaje combinado EL	151
5.25. Matriz de confusión de la identificación de usuarios: tasa de identificación correcta utilizando 16 componentes y una sola sesión.	157
5.26. Matriz de confusión de la identificación de usuarios: tasa de identificación correcta utilizando 16 componentes y multisesión	158
5.27. Matriz de confusión de la identificación de usuarios: tasa de identificación correcta utilizando 32 componentes y una sola sesión	159

5.28. Matriz de confusión de la identificación de usuarios: tasa de identificación correcta utilizando 32 componentes y multisesión	160
A.1. Diagrama de bloques del Módulo Reconfigurable de Reconocimiento . .	175
A.2. Diagrama de caso de uso: Gestión de tareas de reconocimiento local . .	180
A.3. Diagrama de caso de uso: Gestión de paquetes de reconocimiento local	189
A.4. Diagrama de caso de uso: Gestión de tareas de reconocimiento remoto .	198
A.5. Diagrama de caso de uso: Gestión de paquetes de reconocimiento remoto	207
A.6. Diagrama de caso de uso: Solicitud de servicios de reconocimiento . . .	212
A.7. Diagrama de secuencia CU-18: Solicitar servicio de reconocimiento local.	215
A.8. Diagrama de secuencia CU-19: Solicitar servicio de reconocimiento remoto.	216
A.9. Árbol de directorios de un paquete de reconocimiento.	218
A.10. Clase abstracta “Task” para la implementación de paquetes de reconocimiento	220

Introducción

Los avances en la interacción humano-robot involucran robots que cada vez son más adaptables a las personas con las que interactúan. La adaptación en los robots, se entiende como la capacidad de comprender el ambiente y los usuarios para modificar su comportamiento en consecuencia [15]. Esta característica, se considera como un componente fundamental en la interacción y requiere del uso de diferentes tareas de reconocimiento, aplicadas a: rostros [15, 19, 22, 68], voces [15, 19, 68], gestos [6, 19, 22], emociones [29, 58], objetos [19, 42], entre otros. Estas tareas son implementadas utilizando diferentes algoritmos de aprendizaje de máquina y son usadas en diferentes escenarios de interacción.

Los diferentes robots construidos para el estudio de la interacción humano-robot, contienen módulos específicos dedicados a realizar tareas de reconocimiento. Los diseños de estos módulos suelen incluir tareas de reconocimiento internas, implantadas en el robot y tareas de reconocimiento externas, implantadas en un servidor. La tendencia actual en el reconocimiento y adaptación de la interacción humano-robot involucra múltiples y frecuentes tareas de reconocimiento, no obstante, las tareas de reconocimiento internas suelen estar limitadas por los recursos del robot, relacionados con el almacenamiento y procesamiento, además, las tareas externas carecen de fiabilidad con respecto al tiempo de respuesta, es decir, su tiempo de respuesta es impredecible [42]. En consecuencia, los diseños de estos módulos mantienen una configuración fija, es decir, las tareas de reconocimiento se implementan en el procesamiento interno o externo en función de los recursos del robot y el escenario de interacción. Sin embargo, manejar una configuración fija limita la capacidad de adaptación en los robots, ya que de acuerdo al escenario de interacción, las tareas de reconocimiento y la frecuencia con que éstas son solicitadas puede cambiar. Es

por esto, que se desarrollará un Módulo Reconfigurable de Reconocimiento diseñado para la interacción entre humanos y robots, que permitirá gestionar las tareas de reconocimiento, de forma tal, que puedan ser usadas internamente o de forma externa según las necesidades del robot en un momento determinado.

Las tareas de reconocimiento tanto de emociones como de personas mediante la voz han sido ampliamente investigadas y utilizadas en el ámbito de la interacción humano-robot. Por lo general, estas investigaciones involucran numerosos algoritmos de procesamiento de audio y de aprendizaje de máquina. En particular, el reconocimiento de emociones mediante la voz, se utiliza para determinar las emociones de una persona y forma parte de los tópicos de investigación más recientes en la interacción humano-robot [14, 17, 27, 61]. Por otro lado, el reconocimiento de personas mediante la voz, se utiliza para identificar la persona que habla y ha sido ampliamente investigado en diferentes áreas como la seguridad y las ciencias forenses [2, 40, 43, 47, 66], no obstante, actualmente es muy popular en el área de la interacción humano-robot. Debido a la importancia y popularidad de estas tareas de reconocimiento, en esta investigación se contemplará como caso de estudio el reconocimiento de emociones y de personas, mediante la voz. Además, se construirán bases de datos en español que se utilizarán como parte del conjunto de entrenamiento de las tareas de reconocimiento incluidas en el caso de estudio.

Este trabajo de investigación está organizado de la siguiente manera: en el primer capítulo se encuentra una contextualización que involucra los antecedentes de esta investigación; en el segundo capítulo se encuentra el marco teórico, el cuál comprende una revisión de las tecnologías y algoritmos utilizados para la implementación de este trabajo; en el tercer capítulo se encuentra un resumen del documento de especificación de requisitos, que además se encuentra como anexo de este trabajo de investigación; en el cuarto capítulo se detalla el diseño de cada uno de los componentes del Módulo Reconfigurable de Reconocimiento, incluyendo el diseño de las bases de datos y de las tareas de reconocimiento involucradas en el caso de estudio; en el quinto capítulo se encuentran las pruebas que se realizaron sobre los componentes del Módulo y sobre las tareas de reconocimiento del caso de estudio; finalmente, en el sexto capítulo se encuentran las conclusiones y recomendaciones de este trabajo de investigación.

Agradecimientos

Ante todo, me gustaría expresar mi sincero agradecimiento a mi tutor, el profesor Jesús Pérez, por el continuo apoyo para la realización de esta trabajo de investigación, por su paciencia, motivación, entusiasmo, resiliencia y gran conocimiento. Su orientación fue de gran ayuda en todo el tiempo de investigación y redacción de esta tesis. No podría haber imaginado tener un mejor asesor y tutor para el desarrollo de mi tesis de pregrado. Adicionalmente, quiero agradecer al equipo de trabajo del LaSDAI, su participación y colaboración hicieron este trabajo mucho más ameno.

Este trabajo de investigación no hubiera sido posible sin el apoyo de mi familia, profesores y amigos. A mi familia, gracias por apoyarme en todo. Estoy especialmente agradecido a mis padres, María Ageda Páez de Mórán y Nerio de Jesús Mórán León los admiro mucho y nunca dejo de aprender de ustedes. A mis hermanas María Gisela Mórán Páez y María Gabriela Mórán Páez y a mi amigo Gustavo Marín, gracias por el apoyo en el desarrollo de este trabajo y en toda mi carrera profesional.

Mi más sincero agradecimiento a la futura Licenciada en idiomas modernos Lisseth Chacón por su ayuda y contribución en diferentes temas relacionados con la fonética y fonología del español. Temas que fueron indispensables para el desarrollo de este trabajo de investigación.

Por último, pero no menos importante, me gustaría agradecer a todas las personas que contribuyeron en la construcción de las bases de datos de este trabajo de investigación: Gilberto Carrillo, Adolfo Contreras, Luis Huiza, Katherine Andrade, Yenifer Ramírez, Astrid Rodríguez, Lizandro Zerpa, María Monseguí, Julián Briceño, Juan Vivas, Javier Solsona, Luis Sánchez, Julio Lugo, Ana Guerrero, Víctor Rojas, Gustavo Mejía y Robi Rondón.

Capítulo 1

Contextualización

Actualmente, los robots sociales orientados a la interacción humano-robot, cuentan con diferentes módulos dedicados a procesar solicitudes de reconocimiento. Las tareas de reconocimiento de estos módulos y el lugar donde son implementados (internamente o externamente) influyen directamente en las características del robot como su capacidad de adaptarse, el tiempo de respuesta, la usabilidad, entre otros.

Las diferentes tareas de reconocimiento que utilizan los robots sociales, cada día se encuentran en aumento; esto se debe, a que la tendencia actual en múltiples áreas de investigación se ha enfocado en utilizar algoritmos de aprendizaje de máquina, permitiendo a los robots, utilizar una variedad más amplia y precisa de tareas de reconocimiento, y por consiguiente tener una concepción mucho más certera del entorno donde se encuentra y de las personas con las cuales interactúa.

No obstante, a medida que la cantidad de tareas de reconocimiento aumenta, se necesita de un mecanismo que permita gestionar las tareas de reconocimiento internas y externas del robot, con el fin de establecer diferentes configuraciones para adaptarse a los escenarios de interacción.

En este capítulo se presentan los antecedentes de esta investigación, los cuales comprenden investigaciones sobre robots sociales actuales (haciendo énfasis en el diseño de sus módulos de reconocimiento), investigaciones sobre el reconocimiento de emociones y personas mediante la voz, e investigaciones sobre construcción de bases de datos orientadas al reconocimiento de personas y emociones. Adicionalmente,

se presenta el planteamiento del problema, la justificación, los objetivos generales y específicos, la metodología y el alcance de este trabajo de investigación.

1.1. Antecedentes

Los antecedentes de esta investigación se dividen en 3 categorías: primero, investigaciones sobre los módulos de reconocimiento utilizados o propuestos para robots sociales; segundo, investigaciones sobre algoritmos de aprendizaje de máquina utilizados para el reconocimiento de personas; y tercero, algoritmos de aprendizaje de máquina utilizados en el reconocimiento de emociones.

1.1.1. Módulos de reconocimiento

El reconocimiento en el contexto de la interacción humano-robot, permite que los robots puedan identificar ciertos aspectos de su entorno y de las personas con las que interactúa. Las tareas de reconocimiento implementadas en los robots sociales suelen involucrar diferentes tipos de entradas: audios, imágenes, videos, entre otras. Estas entradas suelen ser procesadas para ser usadas en algoritmos de aprendizaje de máquina y obtener información de las mismas. La información que se obtiene del algoritmo de aprendizaje depende del tipo de reconocimiento que se requiere, no obstante, esta información suele ser discreta o continua. Los robots sociales normalmente se diseñan con módulos específicos de reconocimiento, los cuales utilizan frecuentemente dos estrategias para gestionar las diferentes tareas solicitadas por el robot: localmente, en los procesos internos del robot, y remotamente, haciendo llamadas a servicios externos de reconocimiento.

Los robots sociales actuales, contienen tareas de reconocimiento que pueden ser solicitadas localmente y remotamente. No obstante, la solicitud de servicios locales ofrece ciertas ventajas como: tiempo de respuesta predecible, no requiere conexión a Internet, no depende de otros dispositivos de hardware, entre otras. Es por esta razón, que el reconocimiento local forma parte del diseño de gran parte de los robots sociales actuales, y además, permite satisfacer servicios de reconocimiento que contienen restricciones de tiempo o son altamente demandados.

En [58], publicación titulada “Identification of Multimodal Signals for Emotion Recognition in the Context of Human-Robot Interaction”, se presenta un clasificador multiclase de emociones para un robot social que actúa como agente de ventas. El clasificador, es basado en dos clasificadores unimodales, cuyas entradas son: audio e imágenes de rostros. En esta investigación se desarrolla un escenario de interacción en el cual el robot social ofrece un producto y mientras hace preguntas reconoce las emociones del cliente mediante el clasificador de manera local, para adaptar las respuestas al usuario y de esta forma guiar el proceso de venta.

A pesar de que el reconocimiento local funciona bien para muchos escenarios de interacción, su desempeño tiende a disminuir a medida de que se incorporan más tareas de reconocimiento, esto se debe principalmente a los recursos del robot, que se ven limitados por la sobrecarga de tareas. Es por esta razón, que en aras de extender las capacidades del robot, se utilizan servicios de reconocimiento remoto. No obstante, otra razón importante por la cual se utilizan servicios remotos, es la incompatibilidad entre componentes.

En la investigación [19], titulada “Towards a Robust Interactive and Learning Social Robot”, presentan un robot humanoide llamado Pepper, diseñado para la interacción social. Cuenta con las siguientes tareas de reconocimiento: objetos, rostros, humanos, voz y patrones de movimiento. El robot Pepper provee toda su capacidad interna de reconocimiento mediante el framework NAOqi, para extender sus capacidades se utilizaron librerías externas de percepción. Para la visión se utiliza la librería OpenPose, el cual reconoce poses en los humanos y You Only Look Once (YOLO, por sus siglas en inglés); para la voz se combinó el reconocimiento del framework NAOqi y el de la plataforma Google Cloud Speech. El módulo de reconocimiento humano, consiste en un nodo maestro ROS en el robot y dos nodos externos: uno procesa las imágenes de la cámara del robot Pepper y a través de la biblioteca OpenPose obtiene información de los puntos de los cuerpos humanos; y el otro nodo externo examina los puntos de los cuerpos de cada persona y los procesa para extraer información semántica.

En la investigación [42], titulada “Toward QoS-aware cloud robotic applications: a hybrid architecture and its implementation”, presentan una arquitectura para la robótica en la nube, llamada RobotCloud. Esta arquitectura cuenta con una nube local

privada de propósito específico, encargada de ofrecer servicios para tareas específicas de los robots. Esta arquitectura permite mantener la calidad del servicio utilizando los servicios de la nube local, para tareas que por lo general requieren un uso intensivo y automático, mientras que otras tareas son delegadas a diversos servicios públicos en la nube. Esto con la finalidad de solventar la lentitud de respuesta de los servicios en la nube pública. Como caso de estudio, se realiza un reconocedor de objetos comunes implantado en la nube privada y se utiliza un servicio de reconocimiento de objetos en la nube pública llamada CloudSight.

Las investigaciones actuales hacen uso del Sistema Operativo Robótico (ROS, por sus siglas en inglés), el cual es un framework para el desarrollo de software para robots. Este sistema operativo es ampliamente utilizado por muchos robots sociales, debido a que éste provee servicios estándares como: abstracción de hardware, control de dispositivos de bajo nivel, pase de mensajes entre procesos, entre otros. ROS tiene una arquitectura basada en grafos, donde el procesamiento se da en nodos con los cuales se realizan tareas como: coordinar otros nodos, recibir información, enviar información, entre otros.

En la investigación [22], titulada “Design of a collaborative architecture for human-robot assembly tasks”, desarrollan una arquitectura para tareas de ensamblaje entre humanos y robots utilizando el robot Baxter. Esta arquitectura cuenta con módulos dedicados al reconocimiento y a la interacción humano-robot. El robot Baxter cuenta con reconocedor de gestos y rostros. El reconocedor de gestos utiliza el middleware NiTE 2.2 que procesa los datos de la cámara Kinect v2 del robot. El reconocedor de rostros utiliza el paquete IAI Kinect2 ROS para el procesamiento de las imágenes y el algoritmo de reconocimiento de rostros se encuentra en el paquete ProcRob ROS.

En la investigación [15], titulada “The Impact of Personalisation on Human-Robot Interaction in Learning Scenarios”, se estudia el impacto de las interacciones personalizadas mediante un robot de compañía basándose, en la aceptación social, inteligencia percibida y simpatía en el escenario de la interacción humano-robot. Un módulo de interacción es construido para el aprendizaje de objetos, el cual involucra al usuario con una conversación personalizada antes de enseñarle los objetos al robot

de compañía NICO. El robot NICO cuenta con reconocimiento de rostros, voz y del usuario hablante, además, detección y seguimiento de rostros. Los componentes de su módulo de reconocimiento se diseñan como nodos ROS, que interactúan entre sí mediante el uso de mensajes y servicios. Los 3 componentes son: visión, voz y lenguaje. Para la detección y reconocimiento de rostros un enfoque basado en una red neuronal convolucional (CNN, por sus siglas en inglés) fue utilizado; para el reconocimiento de voz, es utilizado el framework DOCKS, que combina conocimientos específicos con las predicciones del reconocedor de voz basado en la plataforma Google Cloud Speech; para la identificación del usuario mediante la voz, es utilizada la imagen del espectrograma de Mel y el algoritmo del aprendizaje utilizado es una red neuronal convolucional.

En la investigación [6], titulada “Real-time gesture recognition using a humanoid robot with a deep neural architecture”, proponen un modelo de red neuronal profunda para reconocer gestos dinámicos con mínimo procesamiento de imágenes y reconocimiento en tiempo real utilizando el robot humanoide NIMBRO. Se diseñó un escenario de interacción en donde el robot reconoce los gestos del usuario y luego responde de forma verbal el gesto reconocido. Los gestos reconocidos son: parado (sin movimiento), formar un círculo, apuntar a la izquierda, apuntar a la derecha, voltearse, señal de parar. Para los datos de entrenamiento 60 muestras de cada gesto fueron recolectadas y para el reconocimiento fue utilizada una red neuronal convolucional multicanal (MCCNN, por sus siglas en inglés), los resultados mostraron una precisión en promedio para cada gesto del 96.85%. El sistema de comunicación fue implantado en el robot NIMBRO y está compuesto de dos módulos principales: un módulo de reconocimiento de gestos y un módulo para responder el gesto reconocido. Estos módulos se comunican a través de una red ROS implementados como nodos publicadores y nodos subscriptores.

En la investigación [29], titulada “Affective Personalization of a Social Robot Tutor for Children’s Second Language Skills”, se desarrolló un paradigma experimental integrado en el cual los niños juegan un videojuego diseñado para aprender un segundo lenguaje mediante una tablet, en colaboración con un robot social (tutor) de compañía. El sistema mide la valencia y el compromiso mediante un analizador automático de expresiones faciales, que se combinan en una señal de recompensa que alimenta un

algoritmo de aprendizaje por refuerzo, y este, determina el comportamiento verbal y no-verbal del robot. El comportamiento afectivo de apoyo es autónomo, se aprende y se personaliza a cada estudiante mediante múltiples sesiones de tutoría interactiva. El sistema está compuesto de 4 componentes: un robot social llamado Tega que hace el papel de tutor y cambia su comportamiento según la valencia, concentración y respuestas del usuario, una tablet android integrada con el robot social en la cual se encuentra el videojuego, un teléfono inteligente con el paquete de desarrollo Affdex para el análisis de expresiones faciales y una arquitectura cognitiva que integra los componentes anteriores. La integración de estos componentes es llevada a cabo mediante el sistema operativo ROS. El análisis de las expresiones faciales del teléfono inteligente y los resultados del videojuego en la tablet son enviados mediante una conexión WebSocket a otros nodos ROS del sistema permitiendo la adaptación del comportamiento del robot según el usuario.

En la investigación [68], titulada: “iSocioBot: A Multimodal Interactive Social Robot”, se presenta un robot social capaz de interactuar con los humanos a través de distintas modalidades. El robot es capaz de dirigir la atención al hablante dominante, identificar personas mediante el reconocimiento de rostros y reconocimiento del hablante. El identificador de personas utiliza un reconocedor multimodal basado en características del rostro y de la voz, utilizando el algoritmo de aprendizaje k-vecinos más cercanos; para el reconocimiento de voz se utiliza una versión open-source del reconocedor de voz de google.

Las investigaciones actuales sobre robots sociales diseñan sus módulos de reconocimiento y demás componentes de su arquitectura mediante el framework ROS; esta tendencia se debe a que esta plataforma de software permite separar el control de hardware del procesamiento de alto nivel en diferentes programas. Además, permite distribuir, coordinar y comunicar las tareas de procesamiento en diferentes nodos, permitiendo adaptarse a los sistemas de computación distribuida en los cuales se basan los robots actualmente.

1.1.2. Reconocimiento de personas mediante la voz

El reconocimiento de personas mediante la voz, es una de las tareas de reconocimiento más comunes en los robots sociales. Las investigaciones de este tipo, suelen enfocarse en dos aplicaciones: identificación del hablante y verificación o autenticación del hablante. La identificación se encarga de determinar qué persona se encuentra hablando y la verificación se encarga de validar si cierta persona es quien dice ser. Para realizar estos tipos de reconocimiento existen dos categorías: reconocimiento dependiente e independiente del texto. El reconocimiento dependiente de texto utiliza una muestra de voz común para todos los usuarios o única para cada usuario individual, mientras que el reconocimiento independiente de texto no requiere de una muestra en específico para los usuarios.

Uno de los retos más grandes en las investigaciones relacionadas con el procesamiento de voz en general, es la correcta selección de características representativas del audio. Sin embargo, los Coeficientes Ceptrales de la Frecuencia de Mel (MFCCs, por sus siglas en inglés), han mostrado ser las características principales en muchos tipos de aplicaciones como el reconocimiento del hablante (identificación y verificación) y el reconocimiento de emociones. Los enfoques más populares en las investigaciones de reconocimiento de personas mediante la voz utilizan el Modelo de Mezclas Gaussiano (GMM, por sus siglas en inglés) y la cuantización vectorial (VQ, por sus siglas en inglés). En [47], en su investigación titulada “Speaker Recognition for Hindi Speech Signal using MFCC-GMM Approach”, presentan un reconocedor o identificador de hablantes para muestras de voz en Hindi utilizando dos enfoques: el primero, utilizando los MFCCs con la cuantización vectorial (VQ, por sus siglas en inglés); y el segundo los MFCCs con el algoritmo de aprendizaje Modelo de Mezclas Gaussianas (GMM, por sus siglas en inglés), de forma dependiente e independiente del texto. Los resultados obtenidos mostraron una tasa de reconocimiento del 77.64 % y 86.27 % para la identificación independiente del texto, utilizando la VQ y el GMM respectivamente y una tasa de reconocimiento del 85.49 % y 94.12 % para la identificación dependiente del texto, utilizando VQ y el GMM respectivamente.

La identificación y verificación de personas mediante la voz, es una de las tareas más comunes en los escenarios donde la seguridad es necesaria, también es una tarea

presente en muchos robots sociales, para la identificación y la personalización de las interacciones [15, 58, 68].

Una de las ventajas de la identificación dependiente de texto, es que el número de muestras de audio sobre el texto seleccionado para el entrenamiento es menor en comparación a la identificación independiente de texto. Adicionalmente, se simplifica la complejidad al limitar el conjunto de posibles palabras a una sola. En [40], en su investigación titulada “Text dependent voice recognition system using MFCC and VQ for security applications”, presenta la implementación de un reconocedor de voz para asegurar sistemas. En esta investigación se utilizan los MFCCs como vector de características de las muestras de voz del usuario y para identificar al usuario se utiliza la cuantización vectorial utilizando el algoritmo Linde-Buzo-Gray (LBG, por sus siglas en inglés). La menor distancia euclidiana entre la muestra y los centroides correspondientes a las regiones de la VQ se corresponde con la predicción del sistema. Los resultados obtenidos mostraron un 80% de exactitud para la identificación del usuario en ambientes de poco ruido y un 73.3% en ambientes con ruido moderado.

Actualmente, existen muchos esfuerzos por encontrar diversas formas de identificar personas, esto se debe a que la extracción de características (como los MFCCs) de las muestras de audio es una tarea compleja. En [5], en su investigación titulada “Aplicación de una red neuronal convolucional para el reconocimiento de personas a través de la voz”, se presenta un identificador de personas basado en los espectrogramas de frecuencia de las muestras de audio. En esta investigación se utiliza una red neuronal convolucional para realizar el entrenamiento utilizando los espectrogramas de la voz. El objetivo de este trabajo no solo es identificar personas, sino además, disminuir la cantidad de muestras de audio necesarias para el entrenamiento. Los resultados mostraron que utilizando 5 muestras para el entrenamiento se obtuvo una exactitud del 93,34%, utilizando una ganancia de 0.9 sobre los espectrogramas de las muestras.

Realizar tareas de verificación dependientes del texto, requiere que la persona pase por un proceso de recolección de muestras para el entrenamiento. No obstante, para todas las aplicaciones no siempre es factible el proceso de recolección de muestras. En [66], en su investigación titulada “GMM based automatic speaker verification system development for forensics in Bahasa Indonesia”, se presenta un sistema automático

independiente de texto para la verificación del hablante. Este sistema automático utiliza los MFCCs para representar las muestras de voz y el algoritmo de aprendizaje GMM para realizar las predicciones. La base de datos está constituida de dos tipos de muestras: entrevistas grabadas mediante un micrófono y conversaciones naturales grabadas mediante un teléfono. Ambos tipos de muestras simulan escenarios forenses en Bahasa Indonesia. Los resultados obtenidos en las pruebas utilizando conversaciones naturales obtuvieron un porcentaje de verificación del 93.55 % para hombres y 87.1 % para mujeres, por otra parte, los resultados obtenidos en las pruebas utilizando las entrevistas obtuvieron un porcentaje de verificación de 83.87 % para hombres y 87.10 % para mujeres.

Realizar tareas de identificación independientes del texto, es una de las tareas de reconocimiento que mayor atención requieren con respecto a las muestras de entrenamiento. Esto se debe a que estas muestras deben ser seleccionadas con la intención de representar las características particulares de cada uno de los usuarios, para que la exactitud no se vea afectada. En la investigación realizada por [43], titulada “An MFCC-based text-independent speaker identification system for access control”, se —desarrolla un sistema de identificación de usuarios para el control de acceso (MSIAC, por sus siglas en inglés), que identifica el hablante mediante la señal de su voz, llevando la señal al dominio de la frecuencia. Para realizar el entrenamiento se utilizan 43 palabras de un diccionario, con un total de 15 personas. Las características utilizadas para representar la señal de la voz son los MFCCs y el algoritmo de aprendizaje es el GMM. Dentro de los resultados de interés de esta investigación se encuentran dos tipos de experimentos: identificación del hablante utilizando las palabras usadas en el entrenamiento, con la cual se obtuvo una tasa de reconocimiento mínima de 86.87 % e identificación del hablante utilizando palabras distintas a las del entrenamiento, con la cual se obtuvo una tasa de reconocimiento mínima de 84.18 % para las 15 personas.

Uno de los retos encontrados en la identificación de personas mediante la voz, es que la muestra de voz es comparada con todo el conjunto de entrenamiento de personas, es decir, para identificar el hablante se debe hacer una comparación uno a muchos. Esto, trae como consecuencia un bajo nivel de rendimiento en tiempo y en precisión, si el número de personas en el conjunto de entrenamiento es grande. En [2], en su

publicación titulada “Improved Text-independent Speaker Identification System For Real Time Applications”, se presenta un modelo de identificación de hablantes basado en los Coeficientes Ceptrales de la Frecuencia Mel (MFCCs, por sus siglas en inglés) y cuantización vectorial (VQ, por sus siglas en inglés) con detección de género. La detección de género, permite reducir el número de posibles hablantes y la cuantización vectorial permite obtener un vector de características reducido, permitiendo disminuir el tiempo de procesamiento para identificar el hablante. El algoritmo de aprendizaje utilizado se llama Modelo de Mezclas Gaussianas (GMM, por sus siglas en inglés). La base de datos utilizada se conoce como CHAINS [16]. Los resultados obtenidos mostraron que utilizando la VQ y el GMM se obtuvo una tasa de reconocimiento del 88 %, mientras que utilizando la detección de género se obtuvo una tasa de reconocimiento del 91 % con una reducción en tiempo de procesamiento de casi la mitad.

1.1.3. Reconocimiento de emociones mediante la voz

La voz ha sido la forma más directa y natural de expresar ideas, comunicarse con otros y realizar algún tipo de interacción [43]. La voz, es el principal medio de comunicación en los seres humanos y se considera que contiene mucha información sobre el estado emocional de la persona que la emite. Las investigaciones relacionadas al reconocimiento de emociones mediante la voz, se basan en la extracción de características del audio para obtener una representación matemática. Esta representación, en conjunto con los algoritmos de aprendizaje de máquina, permiten realizar predicciones sobre muestras de voz de un usuario en particular. Gran parte de las investigaciones se basan en mejorar la precisión o tasa de reconocimiento de los clasificadores, enfocándose principalmente en 3 aspectos: algoritmos de aprendizaje de máquina utilizados, características del audio y bases de datos empleadas.

En la investigación [61], se utilizan 6 tipos de clasificadores para comparar la tasa de reconocimiento en la predicción de las 6 emociones universales [7], utilizando como entrada la voz. La base de datos utilizada para esta investigación se conoce como eNTERFACE05 [46], la cual es una base de datos audio-visual, que contiene muestras de las 6 emociones universales mencionadas anteriormente. Cada video es convertido

en formato WAV utilizando la herramienta MATLAB. Las características extraídas del audio fueron las siguientes: los Coeficientes Cepstrales de la Frecuencia de Mel (MFCCs, por sus siglas en inglés), coeficientes de predicación lineal cepstral (LPC, por sus siglas en inglés), método de los momentos, segundo método de los momentos, centroide espectral, punto de caída espectral, flujo espectral, compacidad, variabilidad del centroide espectral, media cuadrática, fracción del marco de baja energía, tasa de cruces por cero, frecuencia máxima mediante la tasa de cruces y la transformada discreta de Fourier. Los algoritmos de aprendizaje supervisado utilizados fueron los siguientes: SVM lineal y polinomial, árboles de decisión, redes neuronales, redes bayesianas, algoritmo de los k-vecinos más cercanos y Bayes ingenuo. Los resultados obtenidos mostraron que el árbol de decisión obtuvo la mejor tasa de reconocimiento, la cual fue de 96.21 %.

Uno de los clasificadores de mayor popularidad en el área de reconocimiento de emociones a través de la voz es el SVM. La principal motivación se debe a que el SVM ha demostrado ser uno de los algoritmos que mayor tasa de reconocimiento tiene cuando se trata de pruebas dependientes e independientes de la persona. En la investigación [17], se utiliza el clasificador SVM con 4 núcleos distintos. La base de datos utilizada es la base de datos Pocala [30] y las emociones utilizadas fueron: ira, miedo, felicidad, tristeza y aburrimiento. Las características seleccionadas fueron: el tono, los formantes, la tasa de cruces por cero, los MFCCs y parámetros estadísticos. Se utilizaron los núcleos: lineal, cuadrático, radial y polinomial. Los resultados obtenidos mostraron que el SVM con núcleo radial obtuvo la mejor tasa de reconocimiento del 84 %. Como conclusión se obtuvo que los núcleos lineal y cuadrático tienen una mejor tasa de reconocimiento en las emociones: ira, miedo y tristeza. A diferencia, el núcleo polinomial tiene la peor tasa de reconocimiento.

Un factor que influye en el desempeño del algoritmo son los hiper-parámetros. En la investigación [14], utilizan la base de datos EmoDB [11]. Se utilizan 5 emociones: la ira, la tristeza, la alegría, la neutralidad y el miedo. Las características seleccionadas del audio son: los MFCCs y los Coeficientes del Espectro Dinámico de la Energía de Mel (MEDC, por sus siglas en inglés). La tasa de reconocimiento obtenida utilizando un clasificador SVM con núcleo radial, fue de 93.75 %. En comparación a la investigación

[27], a pesar de que las mismas bases de datos fueron usadas, el cambio de las características y una selección óptima de los hiper-parámetros, produjo que el SVM radial obtuviera una mejor tasa de reconocimiento con respecto a los árboles de decisión.

1.1.4. Bases de datos orientadas al reconocimiento de personas mediante la voz

El reconocimiento de personas mediante la voz, particularmente la identificación y verificación independiente de texto, es una de las tareas de reconocimiento con mayor interés en muchas áreas de investigación, no obstante para entrenar y probar los algoritmos y técnicas utilizadas para el reconocimiento es necesario la creación de bases de datos. El mayor beneficio de crear una base de datos propia, es la posterior evaluación de diferentes técnicas utilizando la misma base de datos y por lo tanto la evaluación cualitativa de métodos y criterios para el reconocimiento de personas a través de la voz [23].

El reconocimiento de palabras a través de la voz, ha sido una de las tareas de reconocimiento con mayor desarrollo en los últimos años, y debido a esto, muchas son las bases de datos que se orientaron específicamente para permitir desarrollar esta área. No obstante, este tipo de bases de datos también permitieron desarrollar otras investigaciones como el reconocimiento de personas. En la investigación [25], se presenta una base de datos conocida como TIMIT, la cual fue diseñada para proveer muestras de audio para la adquisición de conocimiento acústico y fonético, y para el desarrollo y evaluación de sistemas de reconocimiento automático. Esta base de datos está constituida por 630 mensajes de voz, provenientes de 438 hombres y 192 mujeres, donde cada persona lee 10 declaraciones diferentes. Una de las características más notorias de esta base de datos es la clasificación de las declaraciones utilizadas para su construcción, en este caso se utilizaron 3 tipos de declaraciones: dialécticas, fonéticamente compactas y fonéticamente diversas. Adicionalmente, esta base de datos es de sesión única, esto quiere decir que las muestras de cada persona fueron obtenidas en una sola sesión.

Uno de los parámetros más importantes en las bases de datos orientadas al

reconocimiento de personas son las características intra-locutor e inter-locutor [23]. Las variaciones entre las muestras de voz sobre una misma declaración de la misma persona, se le conoce como características intra-locutor. Estas características permiten mejorar la tasa de reconocimiento de los algoritmos y técnicas de reconocimiento. Adicionalmente, puede ser estimada si la misma declaración es leída varias veces por la misma persona. Por otro lado, la variación dada entre diferentes muestras de voz de diferentes personas se le conoce como características inter-locutor. Estas características son causadas por las variaciones individuales en los sistemas vocales, involucrando la articulación del tracto vocal, la excitación de la fuente, entre otros. Ambas características son importantes para permitir la factibilidad de los sistemas de reconocimiento de personas.

Las bases de datos orientadas al reconocimiento de personas mediante la voz, pueden clasificarse en dos: sesión única y multi-sesión. Las bases de datos de sesión única no son óptimas para evaluar sistemas de reconocimiento de personas, debido a la falta de características intra-locutor [23]. En la investigación [32], se construye una base de datos conocida como Polycost, la cuál está constituida por 134 Ingleses nativos y no nativos (74 hombres y 60 mujeres), de 13 diferentes países de Europa. La base de datos fue construida mediante 5 sesiones durante varias semanas en ambientes de oficina y de hogares mediante la red de teléfono. Esta base de datos fue creada para el proyecto COST 250 cuya acción se titula “Speaker Recognition in Telephony” cuyo inicio fue en 1995 y finalizó en 1998.

En el reconocimiento de personas, se encuentran por lo general dos tipos de tareas: identificación y verificación. La verificación consiste en determinar si una persona es quien dice ser, esta tarea es ampliamente utilizada en sistemas de seguridad. En la investigación [12], se presenta una base de datos conocida como YOHO, la cual fue diseñada para la evaluación de verificación de personas en un contexto independiente del texto para aplicaciones de seguridad. Esta base de datos está constituida por 138 personas (106 hombres y 32 mujeres), para cada hablante se realizaron 4 sesiones de inscripción con 24 declaraciones y 10 sesiones de verificación con 4 declaraciones cada una. Las grabaciones fueron realizadas en una oficina con dispositivos de alta calidad durante un periodo de 3 meses.

Las bases de datos orientadas al reconocimiento de personas siguen diversos criterios de construcción, enfocándose en aspectos como: las declaraciones utilizadas, los fonemas de las palabras, la multi-sesión, entre otros. En la investigación [23], se realiza una revisión de las bases de datos utilizadas para el reconocimiento de personas, estableciendo una taxonomía y adicionalmente construyendo una base de datos conocida como ELSDSR, cuyo objetivo principal es proveer muestras de audio para el desarrollo y evaluación de sistemas de reconocimiento de personas. El texto se encuentra en inglés, los participantes de la base de datos son 20 Daneses, 1 Islandés y 1 Canadiense (12 hombres y 10 mujeres). A diferencia de otras bases de datos como TIMIT [25], la cual incluye únicamente frases y palabras cortas, los mensajes en esta base de datos son extensos y comprensivos, para permitir capturar características intra-locutor. La base de datos está constituida por 154 oraciones de entrenamiento, cuya selección intenta representar todas las posibles pronunciaciones del idioma inglés; y 44 declaraciones de validación.

1.1.5. Bases de datos orientadas al reconocimiento de emociones mediante la voz

Uno de los desafíos que involucra el reconocimiento de emociones, es la capacidad de reconocer emociones en ambientes poco controlados. Gran parte de las investigaciones orientadas al reconocimiento de emociones, logra una tasa de reconocimiento considerablemente alta en las pruebas realizadas, como en [14] [34] [56]. No obstante, en la práctica estos clasificadores no suelen tener el mismo desempeño. Esto se debe a muchos factores, tales como las diferencias que existen entre las muestras de las personas que forman parte del conjunto de entrenamiento y el conjunto de prueba, por ejemplo: las condiciones de ambiente, los dispositivos utilizados, diferencias culturales, el idioma, la edad, entre otros. Adicionalmente, existen otros factores que afectan la tasa de reconocimiento en las emociones, por ejemplo el desbalance en las bases de datos, la calidad de las emociones capturadas, la diversidad de declaraciones, el número de emociones, entre otros. Muchos son los criterios utilizados para diseñar una base de datos orientada al reconocimiento de emociones, entre las características más

relevantes se encuentran: número de personas, origen de las personas, idioma utilizado, declaraciones utilizadas, entre otros. A continuación se presentan las bases de datos más populares orientadas al reconocimiento de emociones.

En la investigación [11], se presenta una base de datos conocida como Emo-DB. Esta base de datos, fue construida utilizando 10 actores (5 mujeres y 5 hombres), simulando o actuado emociones. Las declaraciones seleccionadas conforman un conjunto de oraciones (5 cortas y 5 largas) usadas diariamente e interpretables en todas las emociones aplicadas. Las grabaciones fueron realizadas en una cámara anecoica ¹, con equipo de grabación de alta calidad. La base de datos consistió en 800 declaraciones, en las cuales están contenidas 7 emociones: neutralidad, ira, miedo, alegría, tristeza, asco y aburrimiento. La base de datos fue evaluada mediante una prueba de percepción con respecto a su reconocibilidad y su naturalidad. Las declaraciones que fueron reconocidas con un porcentaje mayor al 80% y juzgadas con un porcentaje mayor al 60% como natural fueron seleccionadas y etiquetadas. Para mejorar la calidad de las muestras, se utilizaron diferentes audios para ayudar a los actores a reproducir cada una de las emociones. Una de las características más notorias de esta base de datos, es que las 10 declaraciones son expresadas para las 7 emociones distintas, y aunado a eso, estas declaraciones son interpretables en cada de estos casos.

Existen muchas técnicas desarrolladas para reconocer emociones, una de las formas de aumentar la capacidad de reconocer emociones, es mediante el uso de información multimodal. Estos algoritmos utilizan información de diferentes canales de entrada como: la voz y la imagen del rostro; para mejorar la capacidad de reconocimiento de los clasificadores. En la investigación [46], se presenta una base de datos audio-visual conocida como eNTERFACE'05, cuyo propósito es la evaluación de algoritmos de reconocimiento de emociones (unimodal y multimodal). Para reproducir las emociones a cada uno de los participantes se les pidió escuchar 6 historias sucesivas, cada una de ellas evocando una emoción en particular. Cada uno de ellos debía reaccionar en su propio idioma a cada una de las situaciones mientras eran grabados, luego, dos jurados detallaban si el sujeto reaccionaba de manera auténtica, y según este criterio se añadía

¹Una cámara anecoica o anecoide es una sala diseñada para absorber en su totalidad las reflexiones producidas por ondas acústicas o electromagnéticas en cualquiera de las superficies que la conforman (suelo, techo y paredes laterales).

la muestra a la base de datos. No obstante, la distribución geográfica de las personas que fueron parte de la base de datos era muy dispersa y debido a esto, las características como las variaciones del tono y la tasa del habla no eran comunes entre los participantes. Por lo tanto, se tomó la decisión de realizar el mismo protocolo pero reaccionando en inglés. En el segundo protocolo se tomó la decisión de predefinir las respuestas ante los distintos escenarios de las historias, debido a que cuando los actores reaccionaron libremente a cada uno de los escenarios no se expresaron de una manera completamente espontánea. El protocolo final se realizó de la siguiente manera: cada sujeto escuchaba una pequeña historia por cada emoción para intentar inmergirse en el escenario, luego el sujeto reaccionaba mediante cada una de 5 declaraciones predefinidas. La base de datos consistió en 1166 secuencias de video, de las cuales 264 eran constituidas por mujeres y 902 por hombres. Una de las características más notorias de esta base de datos es la distribución geográfica de las personas que fueron parte de la misma, además, esta base de datos está constituida por 5 declaraciones diferentes por emoción.

Una de las dificultades más comunes en la construcción de bases de datos es la captura de emociones auténticas. Aunque gran parte de las bases de datos utilizan emociones actuadas o simuladas, existe un gran esfuerzo por validar la reconocibilidad y naturalidad de cada una de las emociones que conforman la base de datos. En la investigación [30], se presenta una base de datos polaca, la cual está conformada por muestras extraídas de discusiones naturales en programas de televisión. Esta base de datos está conformada por declaraciones de interacciones espontáneas y además provee un amplio rango de emociones básicas y complejas. Cada una de las muestras extraídas fueron etiquetadas por un amplio grupo de expertos y voluntarios. La base de datos está constituida de 15 estados emocionales, los cuales se dividen en primarias: ira, anticipación, alegría, miedo, sorpresa, tristeza, asco; y secundarias: rabia, molestia, éxtasis, serenidad, terror, detención, dolor, pensamiento. La base de datos consistió en 784 muestras. La característica más notoria de esta base de datos es que está conformada por declaraciones espontáneas, además, contiene un rango de emociones mucho más amplio y altamente diferenciado.

Numerosas investigaciones han propuesto diferentes tipos de bases de datos para el reconocimiento de emociones, entre ellas: la base de datos SAVEE [35], la cual

cuenta con 480 muestras de audio con las emociones: ira, asco, sorpresa, alegría, miedo, tristeza y neutralidad. Una de las características más notorias de esta base de datos, es que las emociones son inducidas mediante videos. Existen otros tipos de bases de datos cuyo propósito general no fue la evaluación de algoritmos de aprendizaje para el reconocimiento de emociones, no obstante, son utilizadas para ese fin, entre ellas: la base de datos SUSAS [31], cuyo propósito principal fue el análisis y formulación de algoritmos del reconocimiento del habla en condiciones de ruido y estrés. Otras bases de datos populares como AIBO [67], son construidas a partir de escenarios naturales; en este caso, grabaciones de niños mientras interactúan con un robot. La base de datos tiene 110 diálogos y 29200 palabras en 11 categorías emocionales de ira, aburrimiento, enfático, indefenso, ironía, alegría, maternas, represión, descanso, sorpresa y tacto. El etiquetado de los datos se basa en el juicio de los oyentes. Adicionalmente, existen otras bases de datos orientadas para realizar análisis sentimental como [65], la cual presenta una base de datos de videos en español con 105 muestras etiquetadas mediante su polaridad: positiva o negativa.

1.2. Planteamiento del problema

Actualmente los robots orientados a la interacción con las personas cuentan con diversas tareas de reconocimiento. Las formas en que estas tareas de reconocimiento son utilizadas y demandadas varían de acuerdo a muchos factores, tales como: la persona con la cual interactúa, el escenario de interacción, los recursos internos del robot, el tiempo de respuesta, el tipo de reconocimiento en sí, entre otros.

Las formas más comunes en las cuales los robots actuales solicitan tareas de reconocimiento son: servicios implantados en el robot, servicios públicos disponibles en Internet y servicios privados implantados en servidores locales. Diversas tecnologías son utilizadas para comunicar estos servicios con el robot, entre ellas: APIs, WebSockets, pase de mensajes mediante nodos ROS, entre otros. No obstante, adaptar las tareas de reconocimiento para ser usadas de forma interna (local) o de forma externa (remota) sigue siendo un inconveniente para abarcar muchos escenarios de interacción. Esto se debe a que la configuración de estas tareas de reconocimiento se implementan de forma

fija, por lo que se debe cambiar manualmente la implementación cada vez que cambia el escenario de interacción o los recursos del robot se ven limitados.

Las tareas de reconocimiento utilizadas por los robots actuales, utilizan diferentes tipos de entradas para obtener información sobre su entorno. El entorno de un robot puede llegar a ser muy amplio y por ende puede llegar a requerir múltiples tareas de reconocimiento, no obstante, no todas las tareas se enfocan el entorno, existen tareas orientadas a la interacción con las personas, tales como el reconocimiento de personas y de emociones, que conforman actividades fundamentales para la adaptación y la interacción humano-robot.

En este sentido y siguiendo la tendencia actual, es necesario que los robots sociales cuenten con módulos reconfigurables de reconocimiento, que les permitan configurar sus tareas de reconocimiento según sus necesidades en un momento determinado, permitiendo la adaptación a diferentes escenarios y personas.

1.3. Justificación

Hoy en día, los robots sociales utilizan diversas formas de reconocimiento para mejorar la interacción con los humanos. Muchos de estos robots, utilizan esta información para tomar decisiones inteligentes, mejorar su percepción del mundo y adaptarse.

El diseño de los módulos de reconocimiento de los robots sociales actuales, utilizan una configuración fija de sus servicios, es decir, las tareas de reconocimiento son implementadas en el procesamiento interno o externo de los mismos. Este tipo de configuración fija limita la capacidad de adaptación del robot a diferentes escenarios de interacción.

El desarrollo de un Módulo Reconfigurable para el Reconocimiento, permitirá desarrollar trabajos futuros orientados a una completa adaptación autónoma de tareas de reconocimiento en robots según información de su ambiente o escenario.

Por otro lado, las tareas de reconocimiento más populares en los robots sociales orientados a la adaptación son: el reconocimiento de personas y reconocimiento de emociones. Esto se debe, a que permite a los robots adaptar su comportamiento de

acuerdo a la persona y a su estado emocional.

Es por esta razón, que se abre el interés en desarrollar un Módulo Reconfigurable de Reconocimiento que permita facilitar la adaptación de los robots mediante la gestión de tareas de reconocimiento de forma local o remota según los recursos del robot y de las necesidades involucradas en los escenarios de interacción humano-robot. Además, utilizando tareas de reconocimiento utilizadas en los robots sociales para la adaptación hacia los usuarios.

1.4. Objetivos

1.4.1. Objetivo general

Desarrollar un Módulo Reconfigurable de Reconocimiento para la interacción humano-robot.

1.4.2. Objetivos específicos

- Describir los módulos de reconocimiento utilizados en robots sociales para la interacción humano-robot.
- Identificar los algoritmos de aprendizaje de máquina utilizados para el reconocimiento de emociones y de personas mediante la voz.
- Realizar el análisis de requisitos del Módulo Reconfigurable de Reconocimiento.
- Implementar el Módulo Reconfigurable de Reconocimiento según los requisitos.
- Implementar el reconocedor de personas y emociones mediante la voz, utilizando algoritmos de aprendizaje de máquina.
- Verificar el funcionamiento del Módulo Reconfigurable de Reconocimiento utilizando las tareas de reconocimiento implementadas.

1.5. Metodología

1.5.1. Fase de diagnóstico

- Revisión de artículos científicos relacionados a los módulos de reconocimiento en robots sociales.
- Revisión de artículos científicos relacionados con el reconocimiento de emociones mediante la voz.
- Revisión de artículos científicos relacionados con el reconocimiento de personas mediante la voz.
- Revisión de artículos científicos relacionados con la construcción de bases de datos para el reconocimiento de emociones y para el reconocimiento de personas mediante la voz.

1.5.2. Fase de diseño

- Identificar las herramientas utilizadas para la implementación de servicios de reconocimiento internos y remotos en robots sociales.
- Identificar los algoritmos de aprendizaje de máquina y características utilizadas para el reconocimiento de emociones mediante la voz.
- Identificar los algoritmos de aprendizaje de máquina y características utilizadas para el reconocimiento de personas mediante la voz.
- Identificar los pasos y características que debe tener una base de datos de voces para el reconocimiento de emociones y personas.

1.5.3. Fase de implementación

- Implementar el gestor interno del módulo reconfigurable.
- Implementar el gestor externo del módulo reconfigurable.

- Crear la base de datos en español para reconocer emociones y personas.
- Implementar el reconocedor de emociones mediante la voz.
- Implementar el reconocedor de personas mediante la voz.

1.5.4. Fase de pruebas

- Realizar pruebas para evaluar la tasa de reconocimiento sobre el algoritmo de aprendizaje utilizado para el reconocimiento de emociones.
- Realizar pruebas para evaluar la tasa de reconocimiento sobre el algoritmo de aprendizaje utilizado para el reconocimiento de personas.
- Realizar pruebas unitarias y de integración sobre el Módulo Reconfigurable de reconocimiento.

1.6. Alcance

En esta investigación se pretende realizar un Módulo Reconfigurable de Reconocimiento capaz de gestionar tareas y paquetes de reconocimiento, y a su vez ofrecer servicios de reconocimiento locales y remotos. Este módulo permitirá reconfigurar las tareas de reconocimiento en tiempo de ejecución. Además, como caso de estudio, se implementarán dos tareas de reconocimiento: reconocimiento de personas y de emociones.

El módulo reconfigurable será implementado mediante el sistema operativo robótico (ROS, por sus siglas en inglés), las pruebas se realizaran mediante dos computadoras conectadas a una red local, donde una de ellas servirá como gestor remoto y la otra como gestor local. Para realizar las pruebas sobre el módulo, se simulará un módulo de emisión y recepción que permitirá enviar solicitudes de reconocimiento y obtener las respuestas de las mismas, además, este módulo utilizará las operaciones disponibles para el robot.

La base de datos de entrenamiento para implementar las tareas de reconocimiento contempladas en el caso de estudio, utilizará muestras de audio provenientes de estudiantes universitarios que se presten como voluntarios a esta investigación.

Capítulo 2

Marco Teórico

El ámbito y escala de las aplicaciones robóticas, ha tenido un gran desarrollo en los últimos años. Las áreas de aplicación de la robótica, cada vez son mayores, como: educación, medicina, industria, entretenimiento, entre otras. Los campos de investigación con mayor desarrollo en la actualidad, se enfocan en la construcción de robots dedicados a la interacción humano-robot, es decir, robots sociales con la capacidad de interactuar con las personas. Uno de los retos, en el desarrollo de aplicaciones robóticas es su programación; esto se debe, a que las arquitecturas para robots utilizan diferentes configuraciones de hardware, haciendo una tarea no trivial la reutilización de código. Por esta razón, se han creado frameworks para facilitar el desarrollo de aplicaciones robóticas como: ROS [59], CARMEN [49], PLAYER [26], OROCOS [10], entre otros. Estos frameworks, no solo permiten la reutilización de código, sino además, están diseñados para permitir la escalabilidad en el desarrollo de aplicaciones robóticas, es decir, la integración continua de funcionalidades.

Los robots sociales y en general las aplicaciones robóticas relacionadas con la interacción humano-robot, cuentan con diversas tareas de reconocimiento; esto permite a los robots la posibilidad de obtener información de las personas y de su entorno para actuar en función de las mismas. El principal medio de comunicación de los seres humanos es la voz, por lo que muchas tareas de reconocimiento encontradas en los robots sociales se encargan de procesar la voz mediante algoritmos para extraer información valiosa de la misma. La aplicación más popular en este ámbito es el

reconocimiento de palabras, no obstante, investigaciones recientes utilizan el recurso de la voz para identificar la persona que habla [2, 43, 47, 66] y además la emoción que emite en determinadas oraciones [14, 17, 27, 61].

Para realizar las tareas de reconocimiento descritas anteriormente, es necesario el uso de algoritmos de aprendizaje de máquina que permitan realizar la clasificación de diferentes muestras de audio. Uno de los principales retos en estos tipos de reconocimiento, es el procesamiento en las señales de audio de la voz, esto se debe a que para permitir la identificación de patrones, se deben analizar las señales de audio para extraer características que representen de forma matemática cada una de las muestras de audio utilizadas por las tareas de reconocimiento.

En este capítulo se realiza una introducción al framework ROS (haciendo énfasis en sus componentes principales), una introducción y descripción formal de los algoritmos de aprendizaje máquina utilizados para realizar el reconocimiento de personas y emociones; y por último, se explican de manera detallada los conceptos, análisis y procedimientos utilizados en el procesamiento del audio para la extracción de características.

2.1. Sistema operativo robótico (ROS)

El sistema operativo robótico (ROS, por sus siglas en inglés), es un framework de código abierto para uso modular en aplicaciones robóticas, originalmente diseñado por Willow Garage y actualmente mantenido por la Fundación de Robótica de Código Abierto (OSRF, por sus siglas en inglés) [18]. ROS en realidad no es un sistema operativo en el concepto tradicional de gestión y programación; en cambio, éste provee una capa estructurada de comunicación sobre el sistema operativo en el cual se encuentra instalado [59].

El framework ROS, se ha convertido en el estándar para investigaciones de índole robótica. A nivel general ROS, ofrece grandes servicios como: abstracción de hardware, control de dispositivos de bajo nivel, pase de mensajes entre procesos y gestión de paquetes. La infraestructura de comunicación de ROS, es su ventaja principal, permitiendo comunicar diferentes nodos mediante un mecanismo de publicación

y suscripción basado en tópicos, (ver Figura 2.1). Los nodos son unidades de procesamiento, pertenecientes a un grafo computacional que utilizan canales (tópicos), para comunicarse entre ellos.

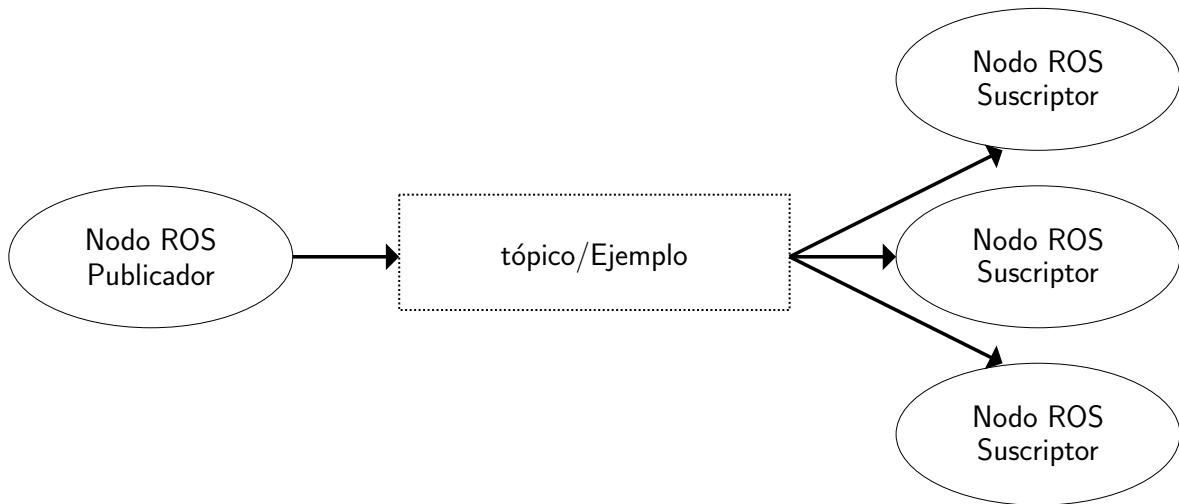


Figura 2.1: Mecanismo de publicación y suscripción de tópicos en ROS

2.1.1. Distribuciones de ROS

ROS fue diseñado para trabajar con el sistema operativo Ubuntu, no obstante, actualmente existe soporte para Debian. Diferentes versiones de este framework han sido publicadas durante los últimos años, una lista de las diferentes versiones puede ser encontrada en la Tabla 2.1.

Versión de ROS	Versión de Ubuntu	Fin del Soporte
Lunar Loggerhead	17.04(Zesty),16.10(Yakkety)y 16.04(Xenial)	Mayo 2019
Kinetic Kame	16.04(Xenial) y 15.10(Wily)	Abril 2021
Jade Turtle	15.04(Wily), 14.10(Utopic) y 14.04(Trusty)	Mayo 2017
Indigo Igloo	14.04(Trusty) y 13.10(Saucy)	Abril 2019
Hydro Medusa	13.04(Raring), 12.10(Quantal) y 12.04(Precise)	Mayo 2015
Groovy Galapagos	12.10(Quantal), 12.04(Precise) y 11.10(Oneiric)	Julio 2014

Tabla 2.1: Distribuciones de ROS desde el 31 de diciembre del 2012

2.1.2. Beneficios de ROS

El framework ROS, también es conocido como meta-sistema operativo [18], que no solo se encarga de ofrecer servicios comunes de sistemas operativos (abstracción de hardware, gestión de recursos, gestión de procesos, entre otros) sino que adicionalmente, ofrece funcionalidades de alto nivel como llamadas sincrónicas y asíncronas, bases de datos centralizadas, sistema de configuración de archivos, entre otros. Adicionalmente, provee librerías y herramientas para obtener, construir, escribir y ejecutar código en múltiples computadoras.

ROS fue diseñado tomando en cuenta los desafíos que presentan los servicios robóticos en gran escala, no obstante la arquitectura resultante fue mucho más general que los dominios para los cuales fue diseñado [59]. Los principios filosóficos de diseño de ROS fueron los siguientes: clientes conectados a una topología P2P, basado en herramientas, multi-lenguaje, liviano, libre y de código abierto. Los beneficios de utilizar ROS para el desarrollo de aplicaciones robóticas son: computación distribuida, reutilización de software y pruebas de desarrollo ágiles [55].

- **Computación distribuida:** Las arquitecturas de robots modernas dependen de software que se extiende a través de múltiples procesos que se ejecutan sobre múltiples computadoras. La necesidad principal es la de comunicación entre múltiples procesos que no necesariamente se encuentran en la misma computadora. ROS provee diferentes mecanismos de comunicación para solventar esta tarea.
- **Reutilización de software:** Las investigaciones en robótica, han aportado una gran cantidad de algoritmos para tareas comunes como: navegación, planificación, mapas, entre otros. Los paquetes estándares de ROS, proveen una gran cantidad de algoritmos listos para ser usados, además, la comunidad de ROS ofrece constantemente paquetes públicos listos para ser usados.
- **Pruebas de desarrollo ágiles:** Unos de los problemas más frecuentes en el desarrollo de pruebas en software robótico, es que muchas veces no se cuenta con el robot, no obstante, cuando el robot esta disponible, el proceso de pruebas es

lento y desafiante. Para solventar estos problemas, ROS provee esencialmente dos soluciones: separación de control de hardware a bajo nivel y toma de decisiones en alto nivel; y registro y reproducción de datos de sensores. Mediante la separación de componentes de alto nivel y bajo nivel, se pueden simular los datos de estos componentes sin la necesidad de usar el robot, además, mediante el registro y reproducción de los datos de los sensores, se pueden reproducir las condiciones del ambiente o generar datos para reproducir escenarios potenciales para el robot.

2.1.3. Niveles conceptuales de ROS

La arquitectura de ROS, ha sido diseñada y dividida en 3 niveles conceptuales [45]: nivel de sistema de archivos, nivel de grafo computacional y nivel de comunidad. El nivel de sistema de archivos, es un grupo de conceptos sobre la organización de los recursos que se encuentran en ROS, como paquetes, descripciones de mensajes y servicios. El nivel del grafo computacional, es la red P2P de los procesos de ROS, y comprende todos los componentes que procesan datos en la red, como los nodos (procesos computacionales) y mensajes. El nivel de la comunidad, es el grupo de conceptos que permite a las diferentes comunidades intercambiar software y conocimiento.

2.1.3.1. Nivel de sistema de archivos

El nivel de sistemas de archivos, es un grupo de conceptos que explican cómo está formado ROS a nivel de organización de recursos, la estructura de las carpetas y el mínimo número de archivos necesarios para trabajar, (ver Figura 2.2).

- **Paquetes:** Los paquetes tiene la mínima estructura y contenido para realizar un programa en ROS. Este puede contener procesos en ejecución (nodos), archivos de configuración, entre otros.
- **Manifiesto del paquete:** El manifiesto del paquete provee información sobre el paquete como: licencias, dependencias, banderas de compilación, entre otros.
- **Metapaquete:** Cuando se requiere agregar diferentes paquetes en un grupo, se utiliza un metapaquete. También son conocidos como stacks (pilas, en español).

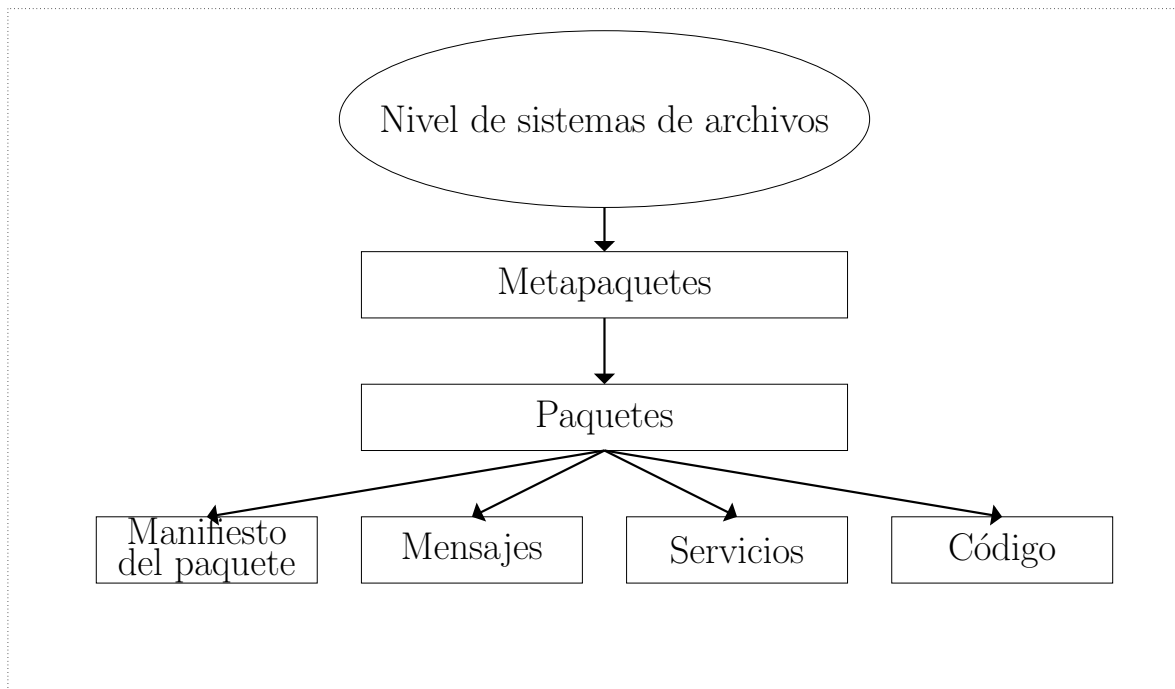


Figura 2.2: Diagrama del nivel de sistemas de archivos de ROS [45]

- **Mensajes:** Un mensaje es la información que un proceso envía a otro proceso. Éstos se definen mediante la especificación de una estructura de datos, no obstante, ROS contiene diferentes tipos de mensajes estándares.
- **Servicios:** Los procesos en ROS ofrecen servicios, éstos se definen mediante la especificación de las estructuras de datos utilizadas para las peticiones y la respuestas.

2.1.3.2. Nivel del grafo computacional

El nivel del grafo computacional, es donde la comunicación entre procesos y sistemas se lleva a cabo. ROS crea una red donde todos los procesos están conectados. Cualquier nodo del sistema puede acceder a la red, interactuar con otros nodos, observar la información que se está enviando y transmitir datos a la red, (ver Figura 2.3).

- **Nodos:** Los nodos son procesos donde la computación se lleva a cabo. Usualmente un sistema contiene múltiples nodos conectados a la red, cada uno con diferentes funciones.

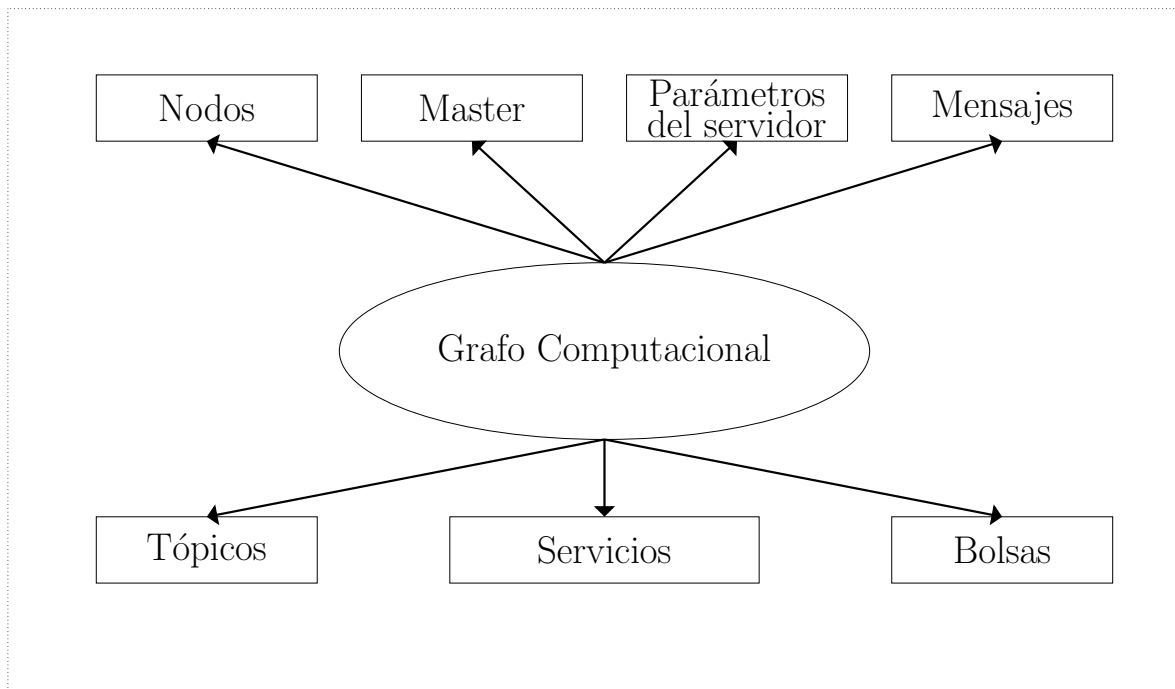


Figura 2.3: Diagrama del grafo computacional de ROS [45]

- **El maestro:** El maestro provee el registro de nombres y un servicio de búsqueda de todos los nodos, además, se encarga de configurar las conexiones entre los nodos. Sin el maestro, no se pueden comunicar los elementos pertenecientes a el grafo computacional, por lo que éste siempre debe ejecutarse. En el caso de un sistema distribuido, debe ser activado en una de las computadoras.
- **Parámetros del servidor:** Los parámetros del servidor ofrecen la posibilidad de almacenar información de forma centralizada. Con estos parámetros, es posible configurar el funcionamiento de un nodo mientras se encuentra en ejecución.
- **Mensajes:** Los nodos se comunican unos con otros mediante mensajes. Un mensaje contiene datos que proveen información a otros nodos. ROS tiene muchos tipos de mensajes estándares, pero además es posible definir nuevos tipos de mensajes.
- **Tópicos:** Cada mensaje debe tener un nombre para ser enrutado por la red de ROS. Cuando un nodo envía datos, se dice que está “publicando” en un tópico y cuando un nodo recibe datos, se dice que está “suscrito” a un tópico. Un nodo

puede publicar a través de un tópico incluso cuando no hay nadie suscrito, lo cual permite desacoplar la producción de mensajes del consumo de los mismos.

- **Servicios:** Cuando requerimos peticiones sincrónicas, es decir, una respuesta por parte de un nodo utilizamos los servicios. Los servicios permiten interactuar con los nodos.
- **Bolsas:** Es una forma en la que ROS permite almacenar y reproducir los mensajes de ROS. Es un importante mecanismo para almacenar información.

2.1.3.3. Nivel de la comunidad

El nivel de la comunidad de ROS describe un conjunto de herramientas y conceptos para compartir conocimientos, algoritmos, y código entre desarrolladores e investigadores. Los recursos de ROS, que permiten intercambiar software y conocimiento son: las distribuciones, los repositorios, la wiki de ROS, las listas de correos, los blogs, entre otros.

2.2. Aprendizaje de máquina

El aprendizaje de máquina, mayormente conocido en inglés como “machine learning”, se ha convertido en una de las ramas más importantes de la inteligencia artificial. Es un área de alto crecimiento cuyo principal objetivo es diseñar y analizar algoritmos que permitan a las computadoras aprender. A pesar de que se considera una disciplina reciente, ha sido ampliamente investigada y utilizada en diferentes áreas de investigación. Particularmente en el área de la interacción humano-robot, ha sido utilizada para realizar una gran cantidad de tareas de reconocimiento como: detección de objetos [42], reconocimiento de voz [15, 19, 68], reconocimiento de emociones [14, 17, 27, 29, 58, 61], reconocimiento de personas [2, 15, 19, 22, 43, 47, 66, 68], entre otros.

En la literatura se encuentran diferentes algoritmos de aprendizaje de máquina, éstos se clasifican según el tipo de aprendizaje, el cual puede ser: supervisado, no supervisado, semi-supervisado y por refuerzo. A continuación se presentan los

algoritmos que serán utilizados para implementar las tareas de reconocimiento involucradas en el caso de estudio de este Proyecto de Grado (reconocimiento de personas y emociones).

2.2.1. Máquina de vectores de soporte

El algoritmo máquina de vectores de soporte (SVM, por sus siglas en inglés), es uno de los algoritmos más populares en el área del aprendizaje de máquina. Particularmente en el área de la interacción humano-robot, el algoritmo SVM, ha sido ampliamente utilizado en el reconocimiento de emociones mediante la voz [14, 17, 27, 61].

El algoritmo SVM, es un modelo de aprendizaje supervisado que permite realizar análisis de clasificación y regresión. Dado un conjunto de datos de entrenamiento, cada uno de ellos etiquetados en una de dos categorías, el algoritmo SVM construye un modelo que permite asignar a nuevos datos una categoría, haciéndolo un modelo de clasificación binario lineal y no probabilístico. A continuación se detallan los componentes principales del algoritmo SVM según [9]:

Considérese un conjunto de vectores de características, el cual se quiere clasificar:

$$X = \{\bar{X}_1, \bar{X}_2, \bar{X}_3, \dots, \bar{X}_n\} \text{ donde } \bar{X}_i \in R^m \quad (2.2.1.1)$$

Para simplificar, se realizará una clasificación binaria, cuyas etiquetas se corresponden con -1 y 1:

$$Y = \{Y_1, Y_2, Y_3, \dots, Y_n\} \text{ donde } Y_n \in \{-1, 1\} \quad (2.2.1.2)$$

El objetivo principal es encontrar los hiperplanos que mejor separen el conjunto de datos, cuya ecuación es la siguiente

$$\bar{W}^T \bar{X} + b, \text{ donde } \bar{W} = \begin{pmatrix} W_1 \\ \vdots \\ W_m \end{pmatrix} \text{ y } \bar{X} = \begin{pmatrix} X_1 \\ \vdots \\ X_m \end{pmatrix} \quad (2.2.1.3)$$

De esta forma, el clasificador puede ser escrito de la siguiente manera:

$$y = f(\bar{X}) = \text{sgn}(\bar{W}^T \bar{X} + b)$$

En un escenario realista, las dos clases son normalmente separadas por un margen con dos límites que dependen de unos pocos elementos. Estos elementos son conocidos como vectores de soporte. Para una expresión matemática más genérica, es preferible renormalizar el conjunto de datos de tal manera que los vectores de soporte estén definidos por los siguientes hiperplanos.

$$\begin{cases} \bar{W}^T \bar{X} + b = -1 \\ \bar{W}^T \bar{X} + b = 1 \end{cases} \quad (2.2.1.4)$$

El objetivo es maximizar la distancia entre estos dos hiperplanos y por lo tanto minimizar la probabilidad de realizar una mala clasificación.

Considerando que los hiperplanos son paralelos, la distancia entre ellos es definida por la longitud de un segmento perpendicular a ambos. Considerando X_1 y X_2 , como puntos que forman un vector perpendicular entre ambos hiperplanos, su vector director viene dado por:

$$\bar{X}_2 - \bar{X}_1 = t\bar{W} \quad (2.2.1.5)$$

Ahora considerando las ecuaciones de los hiperplanos, se obtiene:

$$\bar{W}^T \bar{X}_2 + b = \bar{W}^T (\bar{X}_1 + t\bar{W}) + b = (\bar{W}^T \bar{X}_1 + b) + t\|\bar{W}\| = 1 \quad (2.2.1.6)$$

Simplificando los primeros términos, de la ecuación se obtiene que la distancia máxima entre los dos hiperplanos viene dada por:

$$t\|\bar{W}\| = 2 \quad , \quad t = \frac{2}{\|\bar{W}\|} \quad (2.2.1.7)$$

Todos los elementos descritos anteriormente pueden ser observados en la Figura 2.4

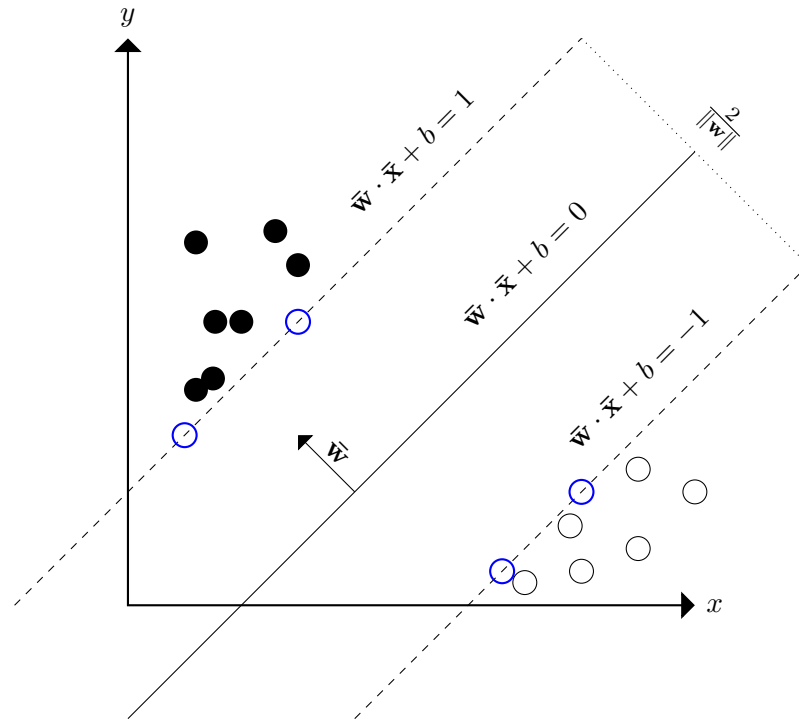


Figura 2.4: Elementos del algoritmo máquina de vectores de soporte

2.2.1.1. Núcleos

En la sección anterior, se asume que los datos de entrenamiento son linealmente separables, sin embargo, los datos del mundo real no siempre se comportan de esa manera. El algoritmo de aprendizaje de máquina SVM, utiliza un conjunto de funciones matemáticas definidas como núcleo (kernel en inglés). Las funciones del núcleo permiten mapear el espacio de los datos a uno linealmente separable, permitiendo a los clasificadores ser aplicables.

Las funciones núcleo retornan el producto interno entre dos puntos en un espacio adecuado, de tal forma, que puede ser interpretada como un tipo de medida de similitud entre los datos de entrada. En la práctica los siguientes núcleos han demostrado ser apropiados para la mayoría de aplicaciones.

- **Núcleo Polinomial:** Es núcleo de propósito general, popular en el procesamiento de imágenes.

$$K(X_i, X_j) = (X_i \cdot X_j + 1)^d, \text{ donde } d \text{ es el grado del polinomio.} \quad (2.2.1.8)$$

- **Núcleo Gaussiano:** Es un núcleo de propósito general; utilizado cuando no existe un conocimiento previo de los datos.

$$K(x, y) = \exp\left(-\frac{\|x - y\|^2}{2\sigma^2}\right) \quad (2.2.1.9)$$

- **Función de base radial Gaussiana (RBF):** Es un núcleo de propósito general; utilizado cuando no hay conocimiento previos de los datos.

$$k(X_i, X_j) = \exp(-\gamma\|X_i - X_j\|^2), \text{ para } \gamma > 0 \text{ o } \gamma = \frac{1}{2}\sigma^2 \quad (2.2.1.10)$$

- **Núcleo Laplace-RBF:** Es un núcleo de propósito general; utilizado cuando no hay conocimiento previos de los datos.

$$K(x, y) = \exp\left(-\frac{\|x - y\|}{\sigma^2}\right) \quad (2.2.1.11)$$

2.2.2. Modelo de mezclas Gaussianas

El modelo de mezclas Gaussianas (GMM, por sus siglas en inglés), es un algoritmo de aprendizaje de máquina considerado como generativo. Particularmente en el área de interacción humano-robot, el algoritmo GMM, ha sido ampliamente utilizado para el reconocimiento de personas mediante la voz [2, 43, 47, 66].

El algoritmo GMM, es un modelo que puede ser usado para el aprendizaje supervisado y no-supervisado. Dado un conjunto de datos de entrada X_1, X_2, \dots, X_N , cuya distribución de los mismos es desconocida (probablemente distribuidos normalmente), el algoritmo GMM intenta encontrar la combinación de distribuciones de probabilidad Gaussiana multidimensional que mejor modelan la entrada.

2.2.2.1. Distribución Gaussiana

La distribución Gaussiana, distribución normal o campana de Gauss es la más importante y famosa de todas las distribuciones estadísticas, (ver Figura 2.5).

La función que describe la distribución Gaussiana es la siguiente:

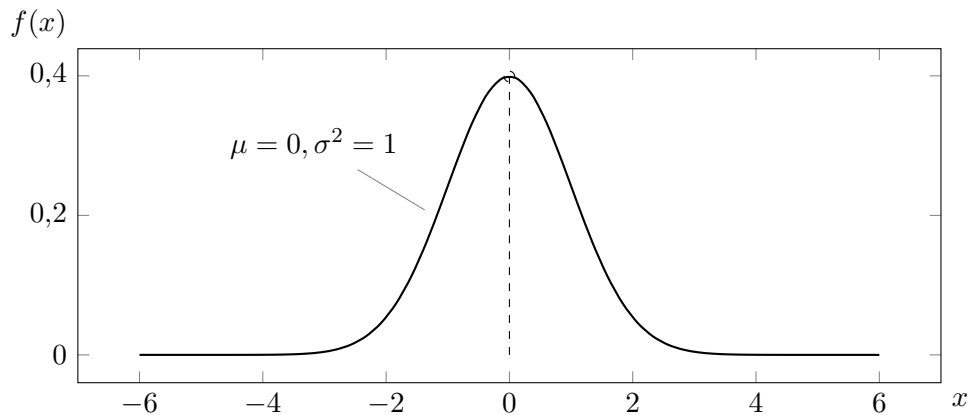


Figura 2.5: Distribución Gaussiana o Normal unidimensional, $\mu = 0$ y $\sigma^2 = 1$

$$N(X; \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \cdot \exp \left\{ -\frac{(x - \mu)^2}{2\sigma^2} \right\} \quad (2.2.2.1)$$

Los dos parámetros de esta función, se conocen como la media μ y la desviación estándar σ . La media de la distribución Gaussiana simplemente mueve el centro de la misma y la desviación estándar es una medida del esparcimiento de la distribución. De forma más específica, la función que representa la distribución Gaussiana se conoce como función de densidad de probabilidad, e indica la probabilidad de observar una entrada X en particular.

De manera general, para d dimensiones, la distribución Gaussiana de un vector $(X^1, X^2, \dots, X^d)^T$ es definida por:

$$N(X; \mu, \sigma^2) = \frac{1}{\sqrt{(2\pi)^{d/2} \sqrt{|\Sigma|}}} \cdot \exp \left\{ -\frac{(x - \mu)^T \Sigma^{-1} (x - \mu)}{2} \right\} \quad (2.2.2.2)$$

donde μ es la media y Σ es la matriz de covarianza de la distribución Gaussiana. En la Figura 2.6, se puede observar una distribución Gaussiana para $d = 2$.

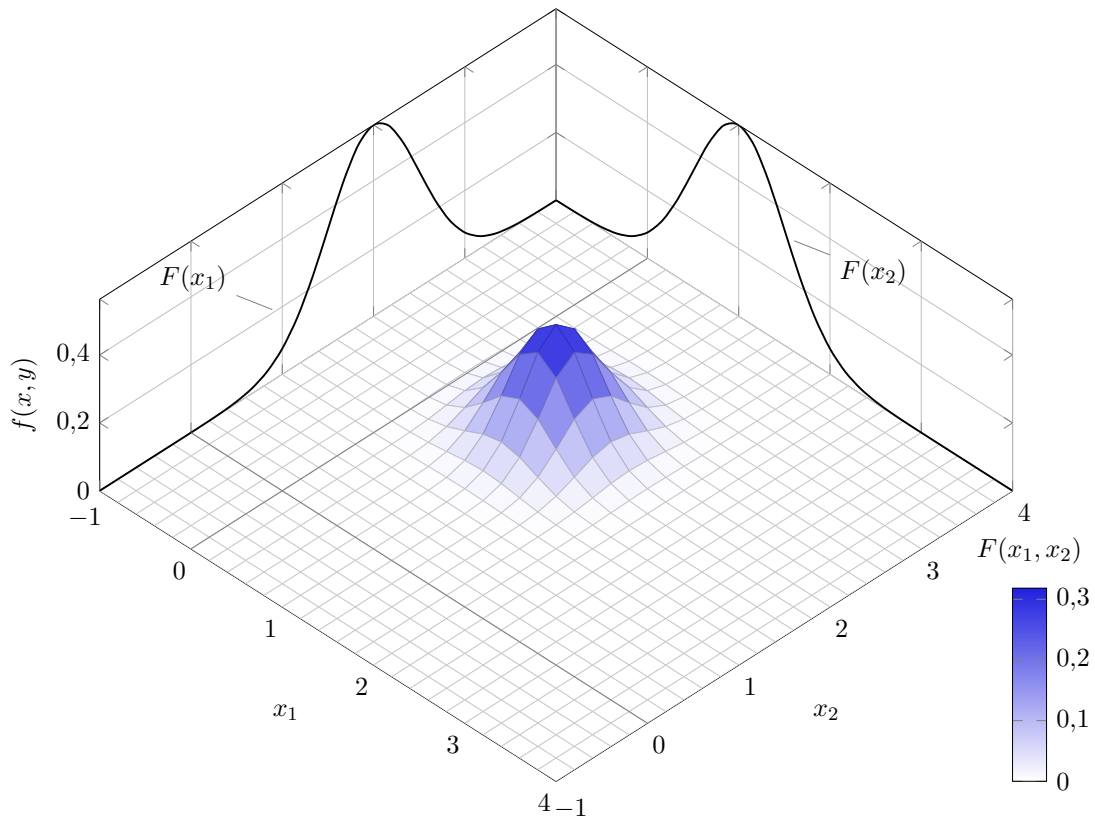


Figura 2.6: Distribución Gaussiana o Normal bivariada, $\mu = (1, 2)^T$ y $\Sigma = \{(0,25, 0), (0,25, 0)\}$

2.2.2.2. Modelo de mezclas Gaussianas (GMM) y el algoritmo K-means

Por lo general el algoritmo GMM suele ser explicado junto con el algoritmo de aprendizaje no supervisado K-means, principalmente por el tipo de agrupamiento que éstos utilizan. El algoritmo K-means, realiza una “asignación fuerte” de los puntos a los agrupamientos, en el cual cada punto es asociado de manera única a un agrupamiento en particular, en cambio, el algoritmo GMM realiza “una asignación suave” basándose en probabilidades posteriores [52].

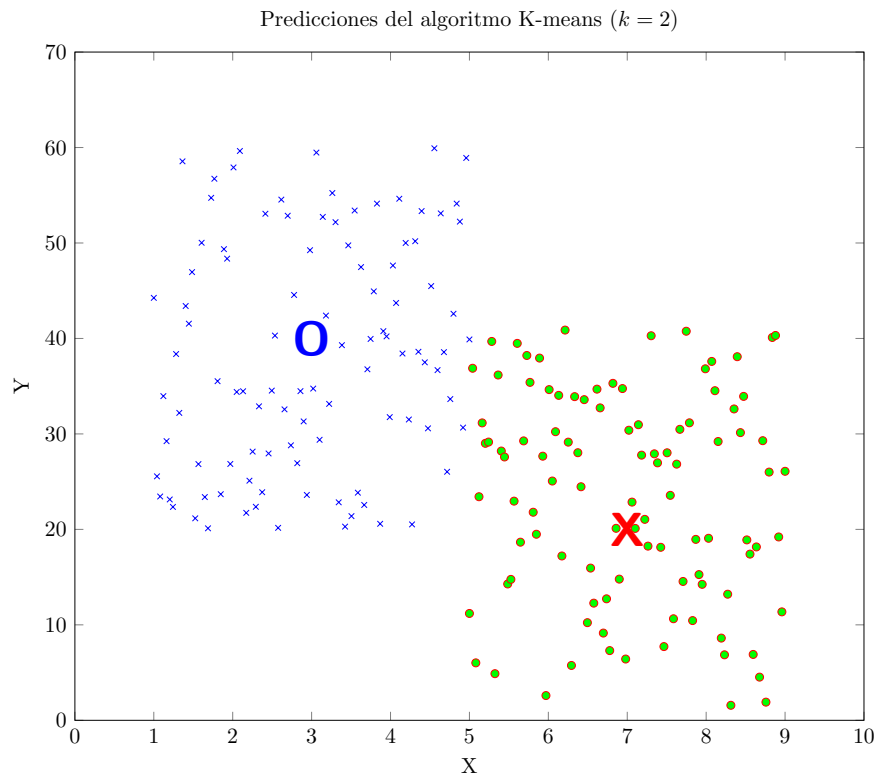


Figura 2.7: Diagrama del algoritmo K-means: 2 agrupamientos

El agrupamiento es parte fundamental de cualquier análisis de datos. Utilizar un algoritmo como K-means, implica una asignación fuerte, es decir, cada punto es asignado de manera definitiva a un agrupamiento en particular, (ver Figura 2.7). Esto representa un gran problema en los casos donde los agrupamientos están esparcidos o se solapan, (ver Figura 2.8).

La distribución de muchos conjuntos de datos es Gaussiana, en otras palabras, se distribuyen normalmente. Por esta razón, el algoritmo GMM toma ventaja de este hecho, para intentar encontrar la combinación de distribuciones de probabilidad Gaussiana que mejor modelan los datos, en lugar de asignar un punto en el centro de un agrupamiento.

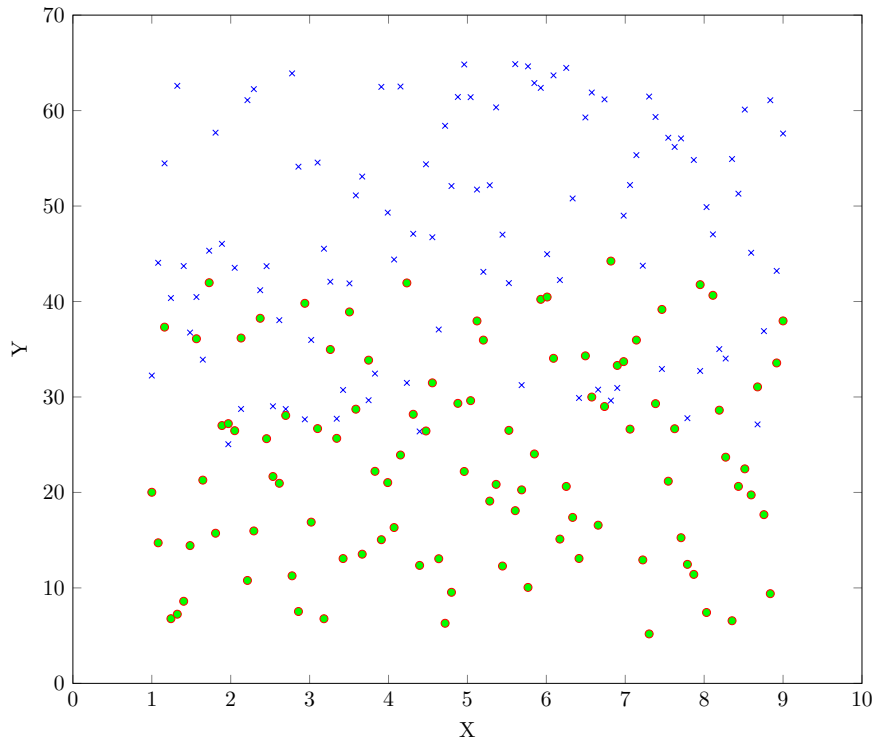


Figura 2.8: Diagrama: datos solapados

2.2.2.3. Clasificación mediante el modelo de mezclas Gaussianas

La idea detrás del algoritmo GMM, es encontrar los parámetros de las distribuciones Gaussianas que mejor modelan los datos de entrada. Ésta es la razón por la cual se le denomina un modelo generativo, se asume que los datos están distribuidos normalmente y se encuentran los parámetros que maximizan la probabilidad de observar estos datos. En otras palabras, se considera cada punto como una combinación de distribuciones Gaussianas y luego se calcula esa probabilidad.

La probabilidad dada en una combinación de K distribuciones Gaussianas es:

$$P(x) = \sum_{j=1}^K \pi_j N(X; \mu_j, \Sigma_j) \quad (2.2.2.3)$$

$$\sum_{j=1}^k \pi_j = 1 \quad y \quad 0 \leq \pi_j \leq 1 \quad (2.2.2.4)$$

Donde π_j es la probabilidad previa (peso) de la j -ésima distribución Gaussianas. La

primera ecuación indica que un punto particular X , es una combinación lineal de K distribuciones Gaussianas. La probabilidad previa π_j , indica el peso de una distribución Gaussiana en particular. La segunda ecuación es una restricción sobre los pesos: la suma de todos debe sumar 1. Para realizar el entrenamiento del modelo, 3 parámetros deben ser actualizados: los pesos π_j , las medias μ_j y las covarianzas Σ_j de cada una de las K distribuciones Gaussianas. Para estimar estos valores, se utiliza el algoritmo de esperanza-maximización.

2.2.2.4. Algoritmo Esperanza-Maximización

Para actualizar los parámetros, se utiliza un algoritmo conocido como esperanza-maximización, también conocido como algoritmo EM [21]. Este algoritmo se divide en dos pasos fundamentales: esperanza y maximización.

La primera parte, se encarga de calcular la probabilidad $W_j^{(i)}$ que tiene cada dato de ser generado por cada una de las K distribuciones Gaussianas. En contraste con la asignación dura del algoritmo k-means, este tipo de asignación se le denomina suave debido a que utiliza probabilidades.

Matemáticamente, se puede observar el cálculo de la probabilidad de la esperanza como una matriz cuyas filas son los datos y las columnas las distribuciones Gaussianas. Un elemento en la fila i y columna j , es la probabilidad de que x^i haya sido generada por la distribución Gaussiana j .

$$W_j^{(i)} = \frac{\pi_j N(X^{(i)}; \mu_j, \Sigma_j)}{\sum_{q=1}^k \pi_q N(X^{(i)}; \mu_q, \Sigma_q)} \quad (2.2.2.5)$$

La segunda parte de este algoritmo es la maximización. En este paso se necesita actualizar los pesos, las medias y las covarianzas. Ésto se realiza mediante las probabilidades (esperanzas) calculadas en el paso anterior.

Para actualizar el peso π_j , se suman todas las probabilidades de que cada punto haya sido generado por una distribución Gaussiana j en particular y se divide por el número total de puntos.

$$\pi_j = \frac{1}{N} \sum_{i=1}^N W_j^{(i)} \quad (2.2.2.6)$$

Para actualizar las medias μ_j , se calcula el promedio de todos los puntos pesados por la probabilidad de que esos puntos hayan sido generados por una distribución Gaussiana j en particular.

$$\mu_j = \frac{\sum_{i=1}^N W_j^{(i)} X^i}{\sum_{i=1}^N W_j^{(i)}} \quad (2.2.2.7)$$

Para la covarianza Σ_j , se calcula la covarianza de todos los puntos pesados por la probabilidad de que estos puntos hayan sido generados por una distribución Gaussiana j en particular. Se realiza esto para cada una de las j -ésimas distribuciones Gaussianas.

$$\Sigma_j = \frac{\sum_{i=1}^N W_j^{(i)} (X^{(i)} - \mu_j)(X^{(i)} - \mu_j)^T}{\sum_{i=1}^N W_j^{(i)}} \quad (2.2.2.8)$$

2.2.3. Árboles de decisión

Árboles de decisión (DT, por sus siglas en inglés) es un algoritmo de aprendizaje supervisado, utilizado para la clasificación y la regresión. Los árboles de decisión son estructuras basadas en un proceso secuencial de decisiones, estos árboles aprenden de los datos para generar reglas de decisión, que luego son utilizadas para realizar la clasificación [9].

Los modelos de clasificación o de regresión de los árboles de decisión son construidos en una estructura con forma de árbol. Su construcción involucra la separación de un conjunto de datos en pequeños subconjuntos mientras que, al mismo tiempo, un árbol de decisión asociado es construido. El resultado final es un árbol con nodos y nodos hoja, que se corresponden con ramificaciones y clasificaciones (ver Figura 2.9).

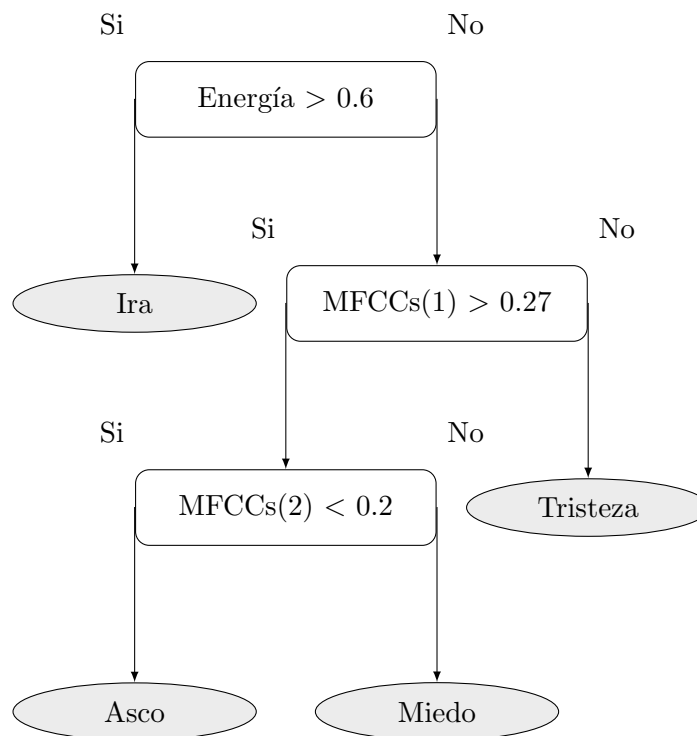


Figura 2.9: Estructura del árbol de decisión: clasificación de emociones basado en la energía y los dos Coeficientes de Mel

El proceso de clasificación inicia desde la raíz, una característica es evaluada y uno de las ramas es seleccionada. Este procedimiento es repetido hasta que un nodo hoja es alcanzado, el cual representa la decisión o clasificación.

2.2.3.1. Decisiones binarias

Considérese el siguiente conjunto de datos de entrada:

$$X = \{\bar{X}_1, \bar{X}_2, \bar{X}_3, \dots, \bar{X}_n\} \text{ donde } \bar{X}_i \in R^m \quad (2.2.3.1)$$

Cada vector \bar{X}_i , contiene m características, por lo que cada una de ellas puede ser una buena candidata para crear un nodo constituido por la tupla: (característica, límite).

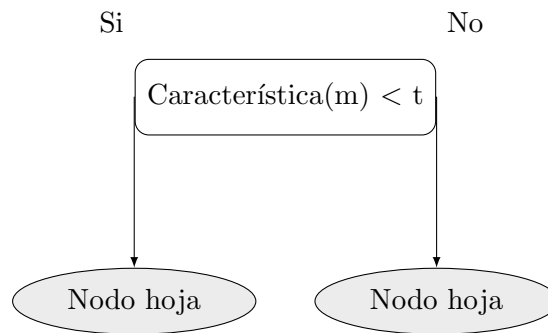


Figura 2.10: Estructura del árbol de decisión: característica y límite de un nodo

De acuerdo con la característica y el límite (ver Figura 2.10), la estructura del árbol puede cambiar. Idealmente, se debe seleccionar la característica que mejor separe el conjunto de datos. En problemas reales realizar esta tarea es frecuentemente imposible, por lo que se debe encontrar la característica que minimice el número de pasos para tomar la decisión.

De manera formal, dada la tupla de decisión de un nodo de la forma:

$$\sigma = \langle i, t_k \rangle \quad (2.2.3.2)$$

donde i , representa el índice de la característica seleccionada para dividir el conjunto de datos en un nodo dado, y t_k representa el límite que determina la selección de la rama izquierda o derecha. El objetivo es reducir la impureza en el menor número de divisiones, para optimizar el proceso de decisión o clasificación.

La impureza es una medida que se utiliza para determinar la homogeneidad (pureza) o heterogeneidad (impureza) de un conjunto de datos basándose en las clases objetivo. Un conjunto de datos se le considera homogéneo o puro, si este contiene una sola clase. Si el conjunto de datos contiene diferentes clases, se le considera heterogéneo o impuro.

Existen muchas formas de medir el grado de impureza de un conjunto de datos, y estas medidas son utilizadas por los algoritmos para la construcción del árboles de decisión. Para definir las medidas de impureza, es necesario considerar el número total de clases:

$$Y = \{Y_1, Y_2, Y_3, \dots, Y_n\} \text{ donde } Y_n \in (0, 1, 2, \dots, P) \quad (2.2.3.3)$$

En un nodo arbitrario j , se puede definir la probabilidad $P(i|j)$, donde i es un índice entre $[1, n]$ asociado con cada una de las clases. Este valor representa una proporción entre el número de clases pertenecientes a la clase i y el número total de muestras en el nodo. A continuación se presentan las medidas de impureza más populares para la construcción de árboles de decisión.

- **Índice de impureza de Gini:** Dada una muestra, el índice de impureza de Gini mide la probabilidad de una clasificación incorrecta si la etiqueta es seleccionada aleatoriamente utilizando la probabilidad de distribución de la rama.

$$I_{gini} = \sum_i^n P(i|j)P(1 - P(i|j)) \quad (2.2.3.4)$$

- **Índice de entropía cruzada de impureza:**

$$I_{entropia-cruzada}(j) = - \sum_i^n P(i|j) \log P(i|j) \quad (2.2.3.5)$$

Este índice es muy similar al índice de impureza de Gini. Esta medida permite seleccionar la división que minimiza la incertidumbre sobre la clasificación, mientras que el índice de impureza de Gini minimiza la probabilidad de realizar una clasificación incorrecta.

2.2.4. Aprendizaje combinado

Los algoritmos descritos en las secciones anteriores (SVM, GMM y DT), generan modelos basados en una sola instancia, a este tipo de modelos se le considera de “aprendizaje fuerte” (strong learners en inglés), cuyo funcionamiento principal es resolver un problema de forma optimizada mediante la búsqueda de las mejores soluciones [9]. Otro enfoque utilizado, se basa en el uso de modelos de “aprendizaje débiles” (weak learners en inglés), que son utilizados de forma conjunta para realizar una clasificación basada en un sistema de votación. Estos métodos pueden ser clasificados en dos categorías:

- **Agregación Bootstrap:** La agregación Bootstrap, también conocida como Bagging, es un meta-algoritmo diseñado para conseguir combinaciones de modelos a partir de una familia inicial, provocando una disminución en la varianza y evitando sobreajuste. El algoritmo Bosques Aleatorios (RF, por sus siglas en inglés) es un ejemplo de múltiples árboles de decisión utilizando la agregación Bootstrap.
- **Boosting:** A diferencia del Bagging, en el boosting no se crean versiones del conjunto de entrenamiento, sino que se trabaja siempre con el conjunto completo de entrada, y se manipulan los pesos para generar modelos distintos. El algoritmo Aumento del Gradiente (GB, por sus siglas en inglés) es un ejemplo de múltiples árboles de decisión utilizando el Boosting.

2.2.4.1. Clasificación basada en votación

La clasificación basada en votación es una solución de aprendizaje combinado en la cual se utilizan diversos modelos (preferiblemente utilizando diferentes algoritmos) para generar un resultado utilizando votación. La decisión final es tomada mediante la selección de la predicción con mayoría de votos. Las estrategias de votación mayormente utilizadas son dos:

- **Votación fuerte:** La votación fuerte (hard voting en inglés), se basa en la selección de la clase con mayor número de votos, $N_c(Y_t)$.

$$\tilde{y} = \operatorname{argmax}(N_c(y_t^1), N_c(y_t^2), \dots, N_c(y_t^n)) \quad (2.2.4.1)$$

- **Votación suave:** La votación suave (soft voting en inglés), se basa en la suma de los promedios de los vectores de probabilidad de cada una de las clases de los clasificadores. La clase ganadora es aquella con el promedio más alto.

$$\tilde{y} = \operatorname{argmax} \frac{1}{\text{Número de clasificadores}} \sum_{\text{Clasificadores}} (P_1, P_2, P_3, \dots, P_n) \quad (2.2.4.2)$$

A continuación se presenta una pequeña descripción de dos algoritmos basados en aprendizaje combinado que utilizan árboles de decisión: Bosques aleatorios y Aumento del Gradiente.

2.2.4.2. Bosques aleatorios

El algoritmo bosques aleatorios es un conjunto de árboles de decisión construidos sobre muestras aleatorias con diferentes políticas de división en los nodos. En lugar de buscar la mejor opción, en cada modelo, un conjunto aleatorio de características es utilizado por cada uno de los árboles de decisión construidos, con la intención de encontrar el límite que mejor separa el conjunto de datos. Como resultado, se entrenarán diversos árboles de decisión y cada uno de estos producirá diferentes soluciones.

Existen dos maneras de interpretar estos resultados; el enfoque más común es basado en votación por mayoría (la clase con más votos es seleccionada). Sin embargo, otra forma de interpretar los resultados, es a través del promedio de las predicciones probabilísticas. Incluso cuando los modelos entrenados son en teoría diferentes, el promedio probabilístico del RF no puede ser muy diferente de la mayoría de predicciones.

2.2.4.3. Aumento del gradiente

El algoritmo aumento del gradiente, es un conjunto de árboles de decisión construidos con el objetivo de minimizar una función de pérdida. La salida del conjunto de modelos puede ser representada de la siguiente manera:

$$Y_E = \sum_i \alpha_i f_i(\bar{x}) \quad (2.2.4.3)$$

donde $f_i(\bar{x})$ es una función que representa uno de los modelos de árboles de decisión entrenados. El algoritmo es basado en el concepto de añadir nuevos árboles de decisión en cada paso, con el objetivo de minimizar una función de pérdida global utilizando el método de descenso del gradiente.

$$Y_E^{n+1} = Y_E^n + \alpha_{n+1} f_{n+1}(\bar{x}) \quad (2.2.4.4)$$

Luego de aplicar el gradiente, la expresión anterior se convierte en:

$$Y_E^{n+1} = Y_E^n + \alpha_{n+1} \sum_i \nabla L(y_{T_i}, y_{E_i}) \text{ donde } y_{T_i} \text{ es la clase objetivo} \quad (2.2.4.5)$$

2.3. Procesamiento de señales de audio

El procesamiento de señales es el área de investigación que se encarga de manipular las señales de información mediante operaciones matemáticas con el fin de modificarla o mejorarla en algún sentido. En general, las operaciones que se realizan sobre éstas, están orientadas al mejoramiento de la interpretación de las mismas. Particularmente, en el área de la interacción humano-robot, el procesamiento de señales de voz ha sido ampliamente utilizado para el reconocimiento de palabras, no obstante, la voz se ha caracterizado por contener mucha información relevante del individuo que la emite, y debido a ésto, muchas investigaciones utilizan la voz como principal fuente de información para reconocer personas y emociones.

Las señales de audio son generalmente conocidas como señales que pueden ser audibles por los seres humanos. Estas señales, usualmente provienen de una fuente de sonido que vibra en un rango de frecuencias audibles. Las vibraciones empujan el aire para formar ondas de presión que viajan a unos 340 metros por segundo. Nuestros oídos internos pueden recibir estas señales de presión y enviarlas a nuestro cerebro para un mayor reconocimiento [36].

Las señales de audio pueden ser caracterizadas de distintas de maneras. Si se consideran los patrones repetidos dentro de una señal de audio, se puede clasificar en dos categorías, (ver Figura 2.13):

- **Sonido cuasi-periódico:** Las formas de la onda consisten en patrones similares repetidos, de modo que es posible percibir el tono. Los ejemplos de tales sonidos incluyen la reproducción monofónica de la mayoría de los instrumentos de música y el canto/habla humana, (ver Figura 2.11).
- **Sonido aperiódico:** Las formas de onda no consisten en patrones repetitivos obvios, por lo que no es posible percibir un tono estable. Los ejemplos de estos sonidos incluyen: golpes de truenos, palma de las manos, partes sordas en palabras humanas como la 's', (ver Figura 2.12).

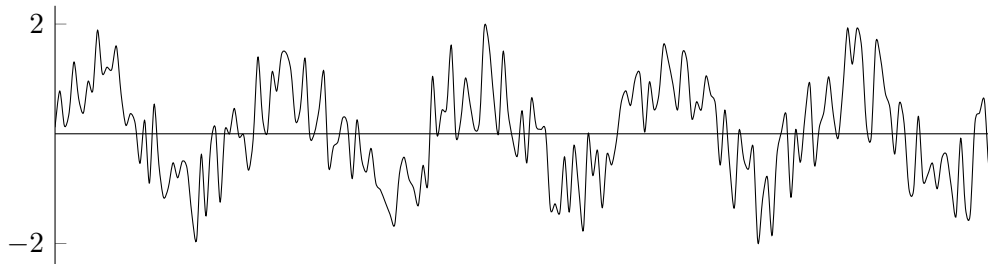


Figura 2.11: (A) Señal cuasi-periódica

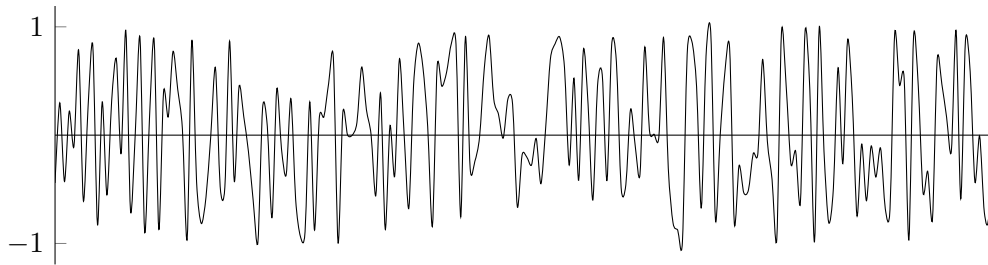


Figura 2.12: (B) Señal aperiódica

Figura 2.13: Clasificación de las señales según sus patrones: aperiódica y cuasi-periódica

En principio, se pueden dividir las señales de audio de voz en segmentos cortos (también conocidos como marcos) con una longitud de aproximadamente 20ms, los cuales se pueden clasificar en:

- **Sonidos de voz:** Estos sonidos son producidos por la vibración de las cuerdas vocales. Dado que se producen por la vibración regular de las cuerdas vocales, se pueden observar los periodos fundamentales en un marco.
- **Sonidos sin voz:** Estos no son producidos por la vibración de las cuerdas vocales. En cambio son producidos por el rápido flujo de aire a través de la boca, nariz o dientes.

2.3.0.1. Parámetros de las señales de audio

Las señales de audio representan la presión del aire como una función del tiempo, la cual es continua en el tiempo y la frecuencia. Cuando se quieren almacenar señales digitales en la computadora se deben considerar los siguientes parámetros:

- **Frecuencia de muestreo:** Este es el número de puntos muestreados por segundo, en la unidad de Hertz (HZ). Una mayor frecuencia de muestreo indica

una mejor calidad de sonido lo que implica que el espacio de almacenamiento es mayor.

1. **8 KHz**: Calidad de sonido para teléfonos y juguetes.
 2. **16 KHz**: Comúnmente utilizado para reconocimiento de voz.
 3. **44.1 KHz**: Calidad de sonido de CDs.
- **Resolución de bits**: El número de bits usado para representar cada punto de muestreo de señales de audio.
 1. **8-bit**: El rango correspondiente es de $0 \sim 255$ o $-128 \sim 127$.
 2. **16-bit**: El rango correspondiente es de $0 \sim 65535$ o $-32768 \sim 32767$.
 - **Canales dobles o simples**: Mono para un único canal y estéreo para dos canales.

2.3.0.2. Procesamiento a corto plazo de las señales de audio

Cuando se realiza el procesamiento de señales de audio, usualmente se usa el término análisis a corto plazo (short-term en inglés), debido a que muchas señales de audio son más o menos estables dentro de un periodo de tiempo corto, por ejemplo 20ms. Estos periodos de tiempo en los cuales se considera que la señal de audio es estable, se le denominan marcos (m_1, m_2, \dots, m_n), los cuales son agrupados en bloques (ver Figura 2.14) y pueden estar superpuestos para capturar cambios sutiles en las señales de audio.

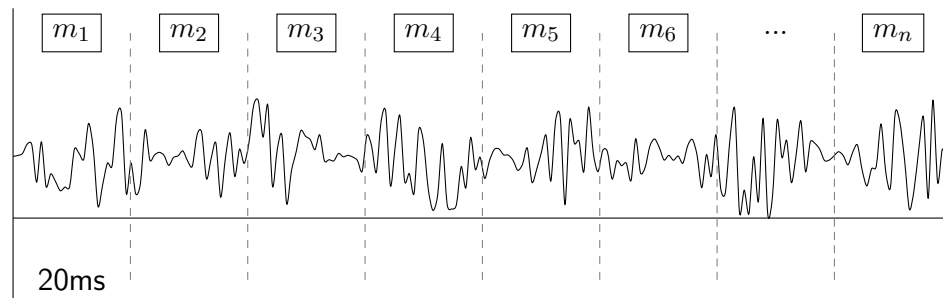


Figura 2.14: Análisis a corto plazo del audio: bloques de marcos

El audio se caracteriza por tener grandes cantidades de información, usualmente esta información es representada en forma matemática, a través características que representan aspectos importantes de las mismas. Para realizar esta extracción se procesan las señales de audio mediante un análisis a corto plazo, el cual se explicará a continuación de manera similar a [36]:

1. Se realizan bloques de marcos, de tal forma que el flujo de señales de audio es convertido en un conjunto de marcos. El tiempo de duración para cada marco usualmente es entre 20ms y 30ms. Si la duración de los marcos es muy grande no se podrá atrapar las características de variación en el tiempo de las señales de audio; si la duración es muy corta no se podrá extraer características válidas.

Usualmente el tamaño del marco (en términos de puntos de muestreo) es una potencia de 2 (256, 512, 1024) de tal forma que es compatible con algoritmos como la transformada rápida de fourier (FFT, por sus siglas en inglés).

2. Si se quiere reducir las diferencias entre marcos, se puede permitir solapamiento o superposición entre ellos. El solapamiento es de $1/2$ a $2/3$ del marco original.
3. Asumiendo que las señales de audio dentro de un marco son estacionarias, se pueden extraer características acústicas en cada uno de ellos. Lo cual permitirá realizar un análisis suprasegmental del audio.

Cuando se utilizan estos procedimientos para la extracción de características, se deben tomar en cuenta los siguientes términos:

- **Tamaño del marco (frame size):** Los puntos de muestra de cada marco.
- **Solapamiento del marco (frame overlap):** Los puntos de muestra de solapamiento entre marcos consecutivos.
- **Paso del marco (frame step):** Es igual al tamaño del marco menos el solapamiento.
- **Tasa de marcos (frame rate):** Es el número de marcos por segundo, el cual es igual a la frecuencia de muestreo dividida por el paso del marco.

Los marcos de audio son procesados para obtener las características que servirán como los datos de entrada de los algoritmos de aprendizaje de máquina. Para realizar este procesamiento, se realiza un análisis a largo plazo. El proceso de extracción de características a largo plazo consiste en obtener el promedio de las características de mediano plazo que dependen del procesamiento del audio a corto plazo de la señal de audio. Esta forma de procesar el audio también se le conoce como segmental (corto y mediano plazo) y suprasegmental (largo plazo) [4].

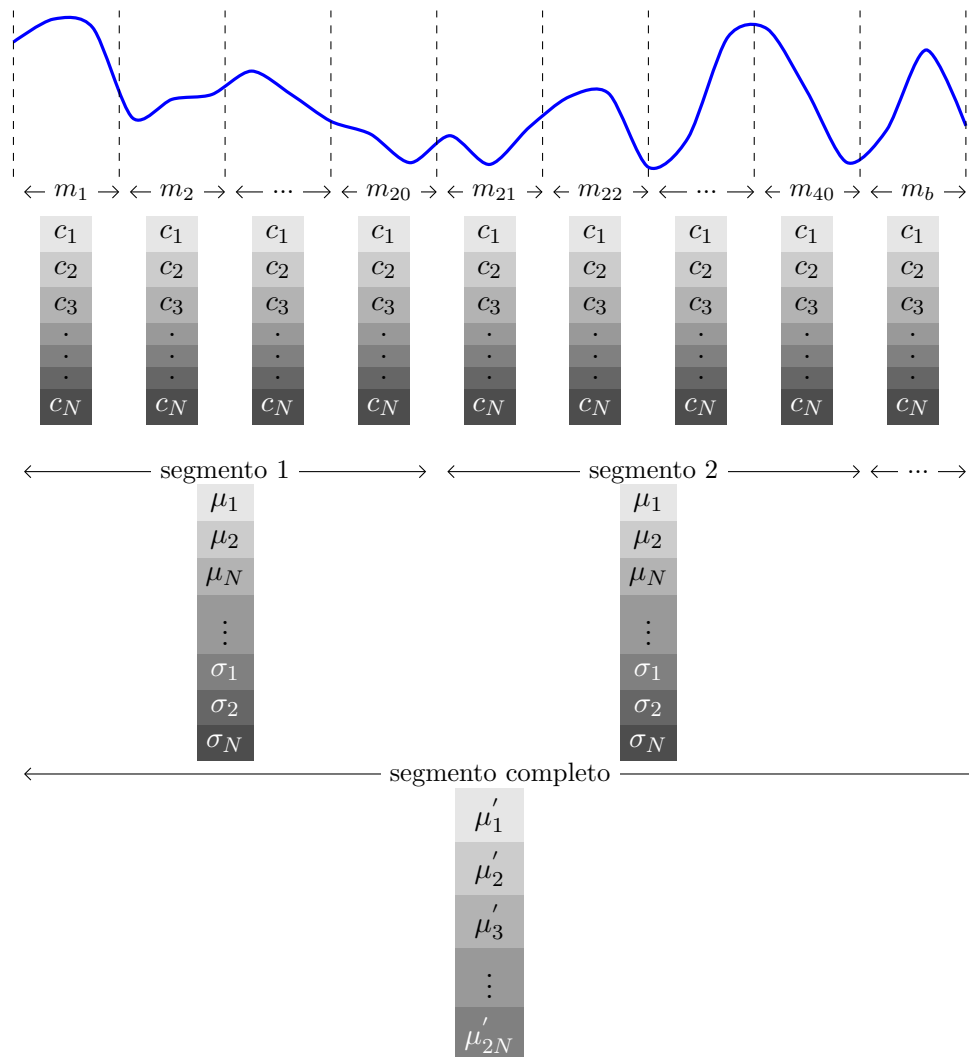


Figura 2.15: Procesamiento a largo plazo o suprasegmental del audio

En la Figura 2.15, se puede observar un esquema que muestra como se lleva a cabo el proceso de extracción de características. En la primera fase del procesamiento (análisis a corto plazo), se obtienen las características c_1, c_2, \dots, c_N de cada marco (m_1, m_2, \dots, m_b) ; en la segunda fase del proceso (análisis a mediano plazo), se extraen estadísticas de las características particulares de cada uno de los marcos del bloque, en este caso los promedios $\mu_1, \mu_2, \dots, \mu_N$ y las desviaciones estándar $\sigma_1, \sigma_2, \dots, \sigma_N$; por último, se realiza un procesamiento suprasegmental (análisis a largo plazo), el cual consiste en obtener los promedios $\mu'_1, \mu'_2, \dots, \mu'_{2N}$ de las estadísticas de la fase anterior.

2.3.1. Características del audio

Las características utilizadas en el procesamiento de audio son variadas, y una de las tareas más importantes en las aplicaciones de aprendizaje de máquina que utilizan estas características es la correcta selección de las mismas. A continuación se muestra una descripción de las características utilizadas en este Proyecto de Grado, basadas en [28].

- **Energía o potencia de la señal:** La energía se define como la suma de los cuadrados de las muestras, que usualmente se normaliza dividiendo entre la longitud de la muestra. La energía es la característica más básica en el procesamiento de señales de la voz. Ésta juega un papel importante en el reconocimiento de emociones. Por ejemplo, las emociones como la felicidad o la ira contienen una mayor energía en comparación a la tristeza. Es una de las características más populares utilizadas en [17, 27].

Sea $X_i(n), n = 1, \dots, W_L$ la secuencia de muestras de audio en el i -ésimo marco, donde W_L es el tamaño del marco. La energía a corto plazo es calculada como sigue:

$$E(i) = \sum_{n=1}^{W_L} |X_i(n)|^2 \quad (2.3.1.1)$$

Usualmente la energía es normalizada dividiéndola por W_L para remover la

dependencia de la longitud del marco, quedando el cálculo de la siguiente manera:

$$E(i) = \frac{1}{W_L} \sum_{n=1}^{W_L} |X_i(n)|^2 \quad (2.3.1.2)$$

- **MFCCs:** los coeficientes cepstrales de las frecuencias de Mel han sido muy populares en el campo del análisis de voz. En la práctica, los MFCCs son los coeficientes de la transformación discreta del coseno del espectro de potencias logarítmicas de frecuencias de la escala de Mel. Los MFCCs han sido ampliamente utilizados en el reconocimiento de voz, reconocimiento de personas, reconocimiento de emociones y muchos otros tipos de aplicaciones de análisis de audio y aprendizaje de máquina. Caracterizan la magnitud del espectro y por lo general son usados los 12 primeros coeficientes. En la gran mayoría de investigaciones los MFCCs han mostrado ser la característica que mejores cualidades tiene para el reconocimiento de emociones [14, 17, 27, 61] y personas [2, 40, 43, 47, 66].

Para extraer los coeficientes cepstrales de las frecuencias de Mel de un marco, son necesarios los siguientes pasos:

1. La transformada discreta de Fourier (DFT, por sus siglas en inglés) es calculada. Esta es usada para derivar la representación de la señal en el dominio de la frecuencia (espectral), la cual sirve como entrada para la obtención de muchas características importantes.

Dada una señal discreta en el dominio del tiempo $x(n)$, $n = 0, \dots, N - 1$, con N muestras de longitud, su DFT es calculada como sigue:

$$X(k) = \sum_{n=0}^{N-1} x(n) \exp(-j \frac{2\pi}{N} kn), k = 0, \dots, N - 1, \text{ donde } j \equiv \sqrt{-1} \quad (2.3.1.3)$$

2. El espectro resultante es utilizado como entrada a un banco de filtros de la escala de Mel que consiste en L filtros. Los filtros usualmente tienen una frecuencia triangular superpuesta. La escala de Mel introduce una función

de distorsión de frecuencia que intenta ajustarse a ciertas observaciones psicoacústicas, (ver Figura 2.16). A través de los años varias funciones de distorsión de frecuencias han sido propuestas por ejemplo:

$$f_w = 2595 * \log(1 + f/700) \quad (2.3.1.4)$$

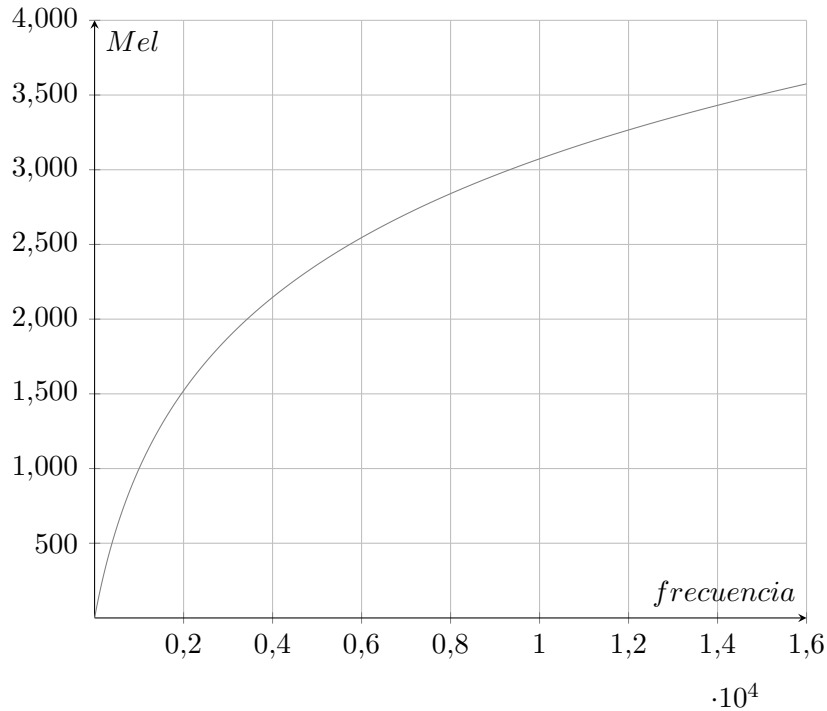


Figura 2.16: Función de distorsión de frecuencias f_w

Si $\tilde{O}_k, k = 1, \dots, L$, es la potencia en la salida del k -ésimo filtro, entonces los MFCCs están dados por la siguiente ecuación

$$C_m = \sum_{k=1}^L (\log \tilde{O}_k) \cos\left[m\left(k - \frac{1}{2}\right) \frac{\pi}{L}\right], m = 1, \dots, L. \quad (2.3.1.5)$$

De esta manera, de acuerdo a la ecuación anterior, los MFCCs son los coeficientes de la transformación discreta del coseno del espectro de potencias logarítmicas de frecuencias de la escala de Mel.

- **Tasa de cruces por cero:** La tasa de cruces por cero (ZRC, por sus siglas en inglés), de un marco de audio, es la tasa de cambios de signo de la señal en el marco. En otras palabras, es el número de veces que la señal cambia de valor, de negativo a positivo y viceversa, dividido por la longitud del marco. La ZRC se define mediante la siguiente ecuación:

$$Z_i = \frac{1}{W_L} \sum_{n=1}^{W_L} |sgn[X_i(n)] - sgn[X_i(n-1)]| \quad (2.3.1.6)$$

donde $sgn(\cdot)$ es la función definida por:

$$sgn[X_i(n)] = \begin{cases} 1, & X_i(n) > 0 \\ -1, & X_i(n) < 0. \end{cases} \quad (2.3.1.7)$$

Esta característica es ampliamente utilizada en aplicaciones de reconocimiento de la voz, reconocimiento de emociones y recuperación de información musical. Adicionalmente, es utilizada como una medida del ruido de una señal [28].

Capítulo 3

Análisis de requisitos

En este capítulo se presentan de manera resumida los aspectos más relevantes del documento de especificación de requisitos (SRS, por sus siglas en inglés). Una descripción de cada uno de los componentes principales del Módulo Reconfigurable de Reconocimiento es realizada basándose en su funcionalidad.

El documento de especificación de requisitos, se encuentra como anexo de este trabajo de investigación, el cual fue realizado basándose en la especificación de requisitos de software ANSI/IEEE 830 [1]. En dicho documento se define de manera clara y concisa cada una de las características del Módulo; no sólo a nivel de especificación de requerimientos, sino que además, se detalla de manera muy precisa la descripción del Módulo, su perspectiva, su alcance, su funcionalidad, entre otros.

Finalmente se detallan los requisitos no funcionales del Módulo, entre los cuales se encuentran: restricciones de hardware, requisitos de ejecución, especificación de los paquetes de reconocimiento, entre otros.

3.1. Descripción general del Módulo Reconfigurable de Reconocimiento

El Módulo Reconfigurable de Reconocimiento, es un subcomponente que forma parte de un sistema de mayor complejidad y escala, orientado a la interacción humano-robot. Gran parte de los robots actuales orientados a la interacción humano-robot, cuentan con diversas tareas de reconocimiento. Es por esta razón, que la idea principal de este Módulo se orienta a la gestión de tareas y paquetes de reconocimiento, además de proveer la capacidad de solicitar dichas tareas a un robot en particular. Los usuarios de este Módulo son dos: el robot, quien solicita y gestiona las tareas de reconocimiento de forma local y remota; y el operador, quien gestiona las tareas y paquetes de reconocimiento en el gestor remoto.

Dado que la principal funcionalidad de este producto es la gestión de tareas y paquetes de reconocimiento, es importante resaltar estos conceptos para comprender la utilidad del mismo. Una tarea de reconocimiento es un elemento del sistema de software que ofrece servicios de reconocimiento, en otras palabras, es un algoritmo que permite clasificar muestras de cierto tipo con un propósito específico; y un paquete de reconocimiento, es un paquete de datos que contiene todos los archivos necesarios para ejecutar una tarea de reconocimiento. Una descripción detallada de todos los elementos que componen un paquete de reconocimiento puede ser encontrado en el documento SRS, en la sección A.3.8.

Un aspecto fundamental en el Módulo reconfigurable de reconocimiento son su gestor local y remoto, que representan el procesamiento interno y externo del robot, respectivamente. Los robots sociales orientados a la interacción humano robot, cuentan con arquitecturas de hardware variadas, generalmente utilizan clusters heterogéneos de computadoras para distribuir las tareas del robot. En este caso particular, se hace énfasis en que los gestores representan diferentes unidades de procesamiento o computadoras.

3.1.1. Funciones del Módulo Reconfigurable de Reconocimiento

El Módulo Reconfigurable de Reconocimiento está compuesto de dos partes principales: el gestor local y el gestor remoto. Estos gestores definen gran parte de la funcionalidad del Módulo. Una descripción detallada de éstos puede encontrarse en la sección A.2.2 del documento SRS. A continuación se detallan las funcionalidades de cada gestor y los usuarios con los cuales éstos interactúan.

3.1.1.1. Gestor local

El gestor local representa el procesamiento interno del robot, y contiene toda la funcionalidad asociada con el robot. El gestor local debe proveer al robot de toda la funcionalidad necesaria para gestionar sus tareas y paquetes de reconocimiento, y además, debe ofrecer al robot la capacidad de solicitar la instalación de paquetes.

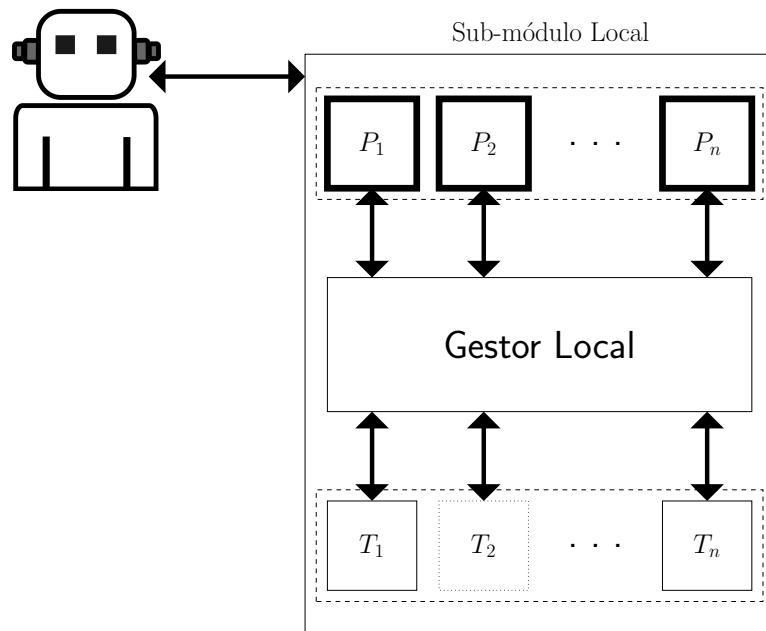


Figura 3.1: Diagrama de bloques del gestor local del Módulo Reconfigurable de Reconocimiento.

En la Figura 3.1, se puede apreciar un diagrama donde se muestran los 3 componentes principales del gestor local: El robot, las tareas de reconocimiento

T_1, T_2, \dots, T_n (ubicadas en memoria primaria) y los paquetes de reconocimiento P_1, P_2, \dots, P_n (ubicados en memoria secundaria). El robot interactúa con el gestor local para gestionar las tareas de reconocimiento, permitiendo decidir qué paquetes se encuentran instalados y qué tareas requiere activadas y desactivadas. No obstante, el robot también puede interactuar con el gestor remoto, utilizando al gestor local como intermediario para solicitar la instalación de paquetes de reconocimiento.

3.1.1.2. Gestor remoto

El gestor remoto representa el procesamiento externo del robot. El gestor remoto, debe proveer al operador de toda la funcionalidad necesaria para gestionar las tareas y paquetes de reconocimiento, y además, proveer la capacidad de responder las solicitudes de paquetes de instalación del gestor local.

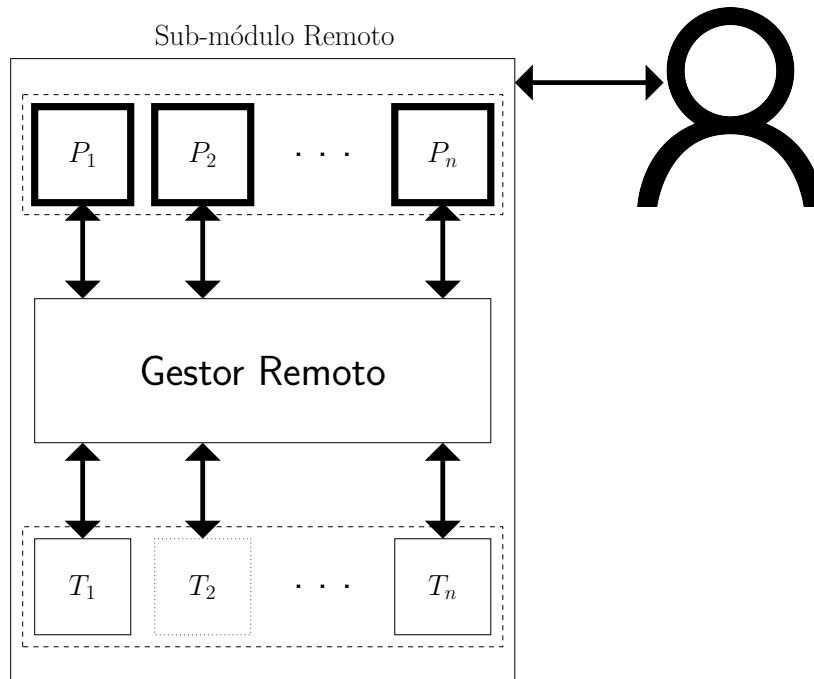


Figura 3.2: Diagrama de bloques del gestor remoto del Módulo Reconfigurable de Reconocimiento

En la Figura 3.2, se puede apreciar un diagrama donde se muestran los 3 componentes principales del gestor remoto: El operador, las tareas de reconocimiento T_1, T_2, \dots, T_m (ubicadas en memoria primaria) y los paquetes de reconocimiento

P_1, P_2, \dots, P_m (ubicados en memoria secundaria). El operador interactúa únicamente con el gestor remoto gestionando qué paquetes se encuentran instalados y qué tareas se encuentran activadas y desactivadas.

3.1.1.3. Servicios de reconocimiento

Otras de las funciones principales del Módulo Reconfigurable de Reconocimiento, se encuentra de manera implícita en sus tareas de reconocimiento; y se trata de los servicios de reconocimiento que estas tareas ofrecen. El Módulo reconfigurable debe proveer los servicios de reconocimiento de cada una de las tareas que se encuentran en el gestor local y remoto; esto con la finalidad, de que el robot realice las solicitudes de dichas tareas de reconocimiento.

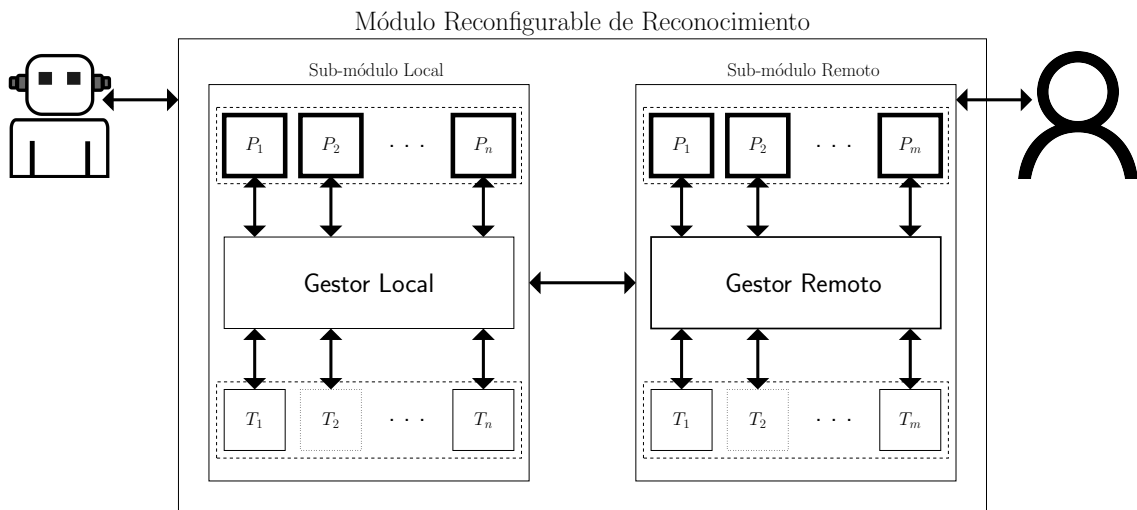


Figura 3.3: Diagrama de bloques del Módulo Reconfigurable de Reconocimiento

En la Figura 3.3, se puede observar el diagrama de bloques del Módulo Reconfigurable de Reconocimiento. En este diagrama se puede apreciar las dos partes principales del Módulo: El sub-módulo local que se corresponde con el gestor local y el sub-módulo remoto que se corresponde con el gestor remoto. Además, se puede apreciar las interacciones de los usuarios con el Módulo y de las tareas y paquetes con sus respectivos gestores. En el caso del sub-módulo local la tarea T_2 está desactivada por el robot, en cambio, la tarea T_2 del sub-módulo remoto se encuentra desactivada por el operador.

3.1.2. Interfaces

Las interfaces del Módulo Reconfigurable de Reconocimiento se dividen en dos: las interfaces de los usuarios y las interfaces de software. Dado que los usuarios del Módulo son dos: el robot y el operador, existen dos interfaces para ellos. El robot utiliza las interfaces de programación de aplicaciones (APIs, por sus siglas en inglés) que ofrece ROS [59]; estas APIs permiten al robot comunicarse con los diferentes elementos que conforman el grafo computacional de ROS. Por otro lado, el operador utiliza el terminal de comandos para conectarse mediante shell seguro (SSH, por sus siglas en inglés) a la computadora que sirve de gestor remoto y de esta manera utilizar las interfaces de comandos que ofrece ROS para realizar sus actividades.

Las interfaces de software de este producto son dos: el sistema operativo robótico (ROS), y el sistema operativo Ubuntu 16.04 Xenial Xerus. A pesar de que ROS puede ser utilizado en distintos sistemas operativos, el soporte oficial del mismo se encuentra solamente para distintas versiones de Ubuntu, en este caso se seleccionó la versión 16.04 debido a que ésta soporta la versión de ROS (Kinectic Kame), cuya duración de soporte es la más prolongada.

Una descripción detallada de las distintas interfaces del Módulo puede encontrarse en el documento SRS, específicamente en la sección A.3.1

3.1.3. Requisitos funcionales

Los requisitos funcionales del Módulo reconfigurable de reconocimiento se dividen en 3: gestión de tareas y paquetes de reconocimiento local, gestión de tareas y paquetes de reconocimiento remoto y servicios de reconocimiento. Gran parte de estos requisitos son similares entre sí, es por eso, que en esta sección se detallarán los requisitos funcionales más importantes. Para describir los requisitos funcionales se realizaron diversos diagramas de casos de uso y para detallar las interacciones entre los componentes del Módulo reconfigurable se realizaron diversos diagramas de secuencia correspondiente a cada caso de uso. Finalmente, se realizó una tabla de especificación de requisitos (ver Tabla 3.1) para cada uno de los casos de uso de los diagramas.

Tabla 3.1: Formato de la tabla de descripción de casos de uso

Caso de Uso:	Identificador del caso de uso.
Nombre:	Nombre del caso de uso.
Evento disparador:	Evento que genera el caso de uso.
Actores:	Actores involucrados en el caso de uso.
Dependencias:	Casos de usos relacionados/dependientes.
Descripción:	Descripción breve del caso de uso.
Pre-condiciones:	Pre-condiciones para que se ejecute correctamente el caso de uso.
Post-condiciones:	Post-condiciones luego de haberse ejecutado el caso de uso.
Flujo Normal:	Flujo normal de eventos que se necesitan para ejecutar el caso de uso.
Flujo Alternativo:	Flujo alternativo de eventos que pueden ocurrir para ejecutar el caso de uso.

La tabla de especificación de requisitos anterior (Tabla 3.1), se utilizó como parte de la especificación de cada uno de los casos de uso del Módulo. Dichas tablas pueden ser encontradas en la sección A.3.5 del documento SRS.

3.1.3.1. Gestión de tareas de reconocimiento local y remota

La gestión de tareas de reconocimiento local y remoto comprenden las actividades de activación y desactivación de tareas en el gestor local y remoto, respectivamente. De manera ideal todos los paquetes de reconocimiento que instale el operador en el gestor remoto deben ser activados, no obstante, se establece la posibilidad de desactivarlos en caso de una actualización o algún error. Por otro lado, el robot puede tener la necesidad de activar o desactivar una tarea específica basándose en el escenario de interacción, por sus recursos, entre otros. A continuación en la Figura 3.4, se puede observar el diagrama de caso de uso de la gestión de tareas de reconocimiento local, la cual es bastante similar a la gestión de tareas de reconocimiento remoto.

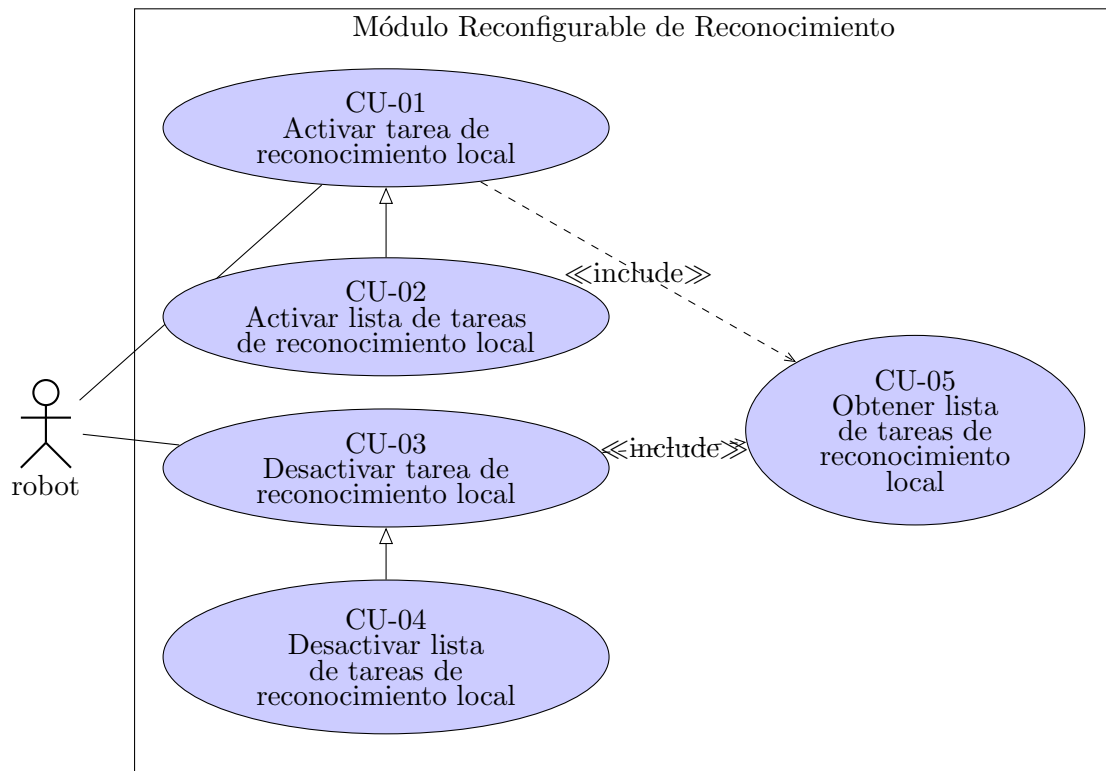


Figura 3.4: Diagrama de caso de uso: Gestión de tareas de reconocimiento local

Los casos de uso más relevantes de este diagrama son dos: CU-01, activar tarea de reconocimiento local; y CU-03, desactivar tarea de reconocimiento local. Cuando se habla de los términos activación y desactivación de tareas, de manera implícita, se hace referencia a los procesos donde residen estas tareas. Por lo tanto, activar una tarea implica ejecutar o levantar un proceso; mientras que, desactivar una tarea implica la finalización o la muerte¹ de un proceso.

En la Figura 3.5, se puede observar el diagrama de secuencia CU-01 correspondiente a la activación de una tarea de reconocimiento local.

¹Término que usualmente se utiliza de su traducción del inglés “kill”

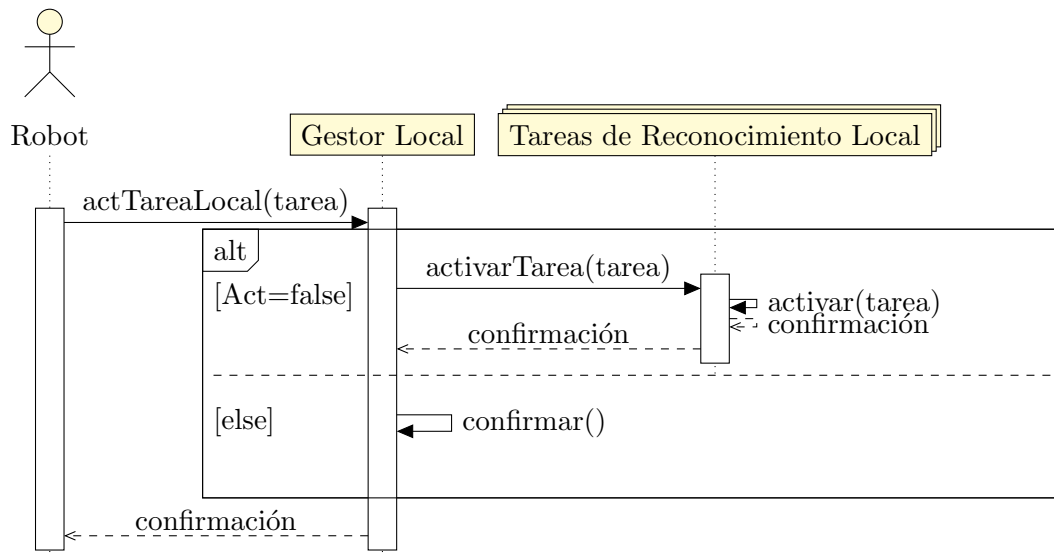


Figura 3.5: Diagrama de secuencia CU-01: Activar tarea de reconocimiento local

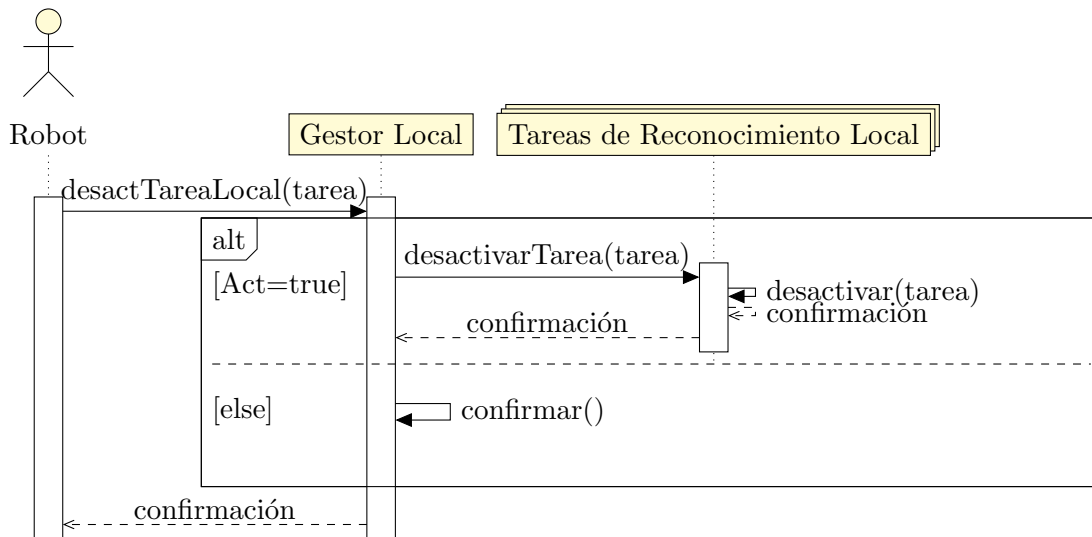


Figura 3.6: Diagrama de secuencia CU-03: Desactivar tarea de reconocimiento local

En la Figura 3.5, se puede observar cómo el robot interactuá con el gestor local para realizar la activación de una tarea de reconocimiento. El gestor local se encarga de verificar si dicha tarea ya se encuentra activada; si este no es el caso, la activa y realiza la confirmación. En el caso de la desactivación, las interacciones entre los elementos son similares (ver Figura 3.6).

En el caso de la gestión de tareas de reconocimiento remoto, las interacciones son similares a excepción de que el usuario que realiza estas acciones es el operador.

3.1.3.2. Gestión de paquetes de reconocimiento local y remoto

La gestión de paquetes de reconocimiento local y remoto comprenden las actividades de instalación y eliminación de paquetes en el gestor local y remoto, respectivamente. El gestor local debe proveer al robot de servicios para instalar paquetes de reconocimiento ubicados en el gestor remoto, por otro lado, el gestor remoto debe proveer al operador la posibilidad de instalar dichos paquetes de reconocimiento; por lo que la instalación de paquetes en los gestores se realiza de manera distinta (ver Figura 3.7). Por otro lado, la eliminación de paquetes en el gestor local y en el gestor remoto se realiza de manera similar.

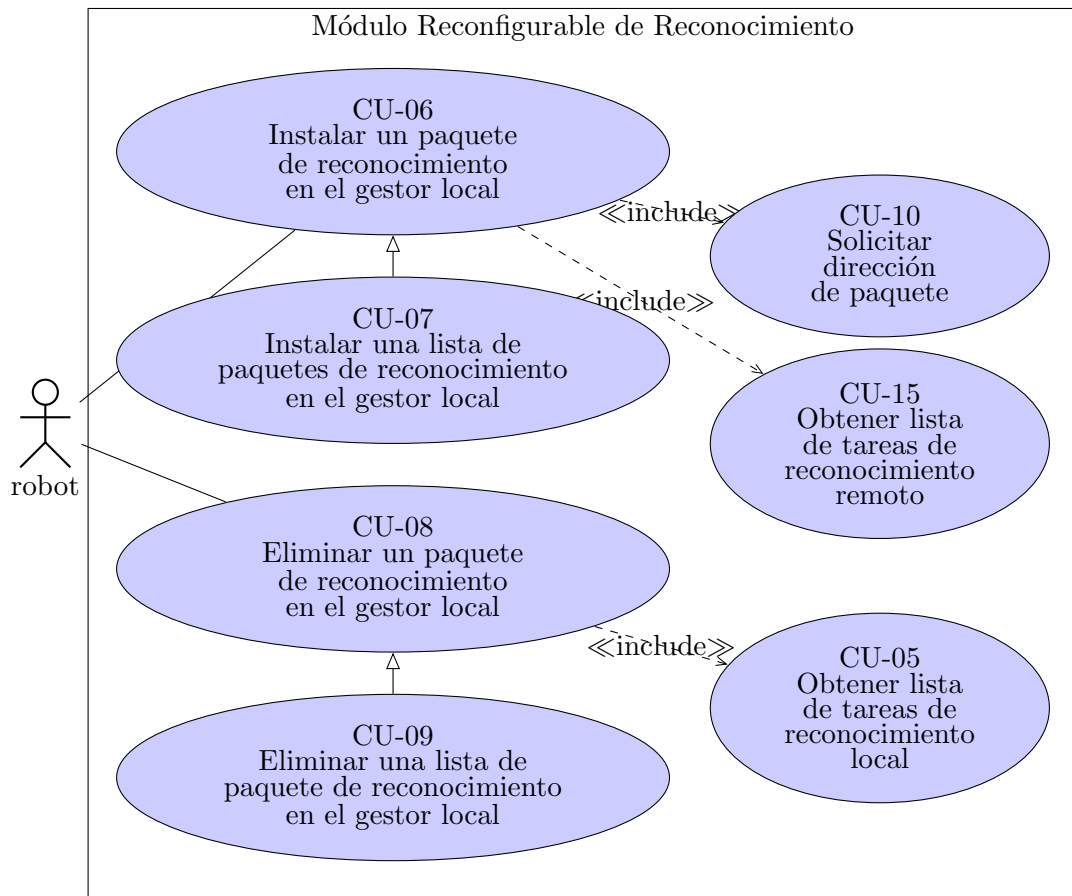


Figura 3.7: Diagrama de caso de uso: Gestión de paquetes de reconocimiento local

Los casos de uso más relevantes del diagrama anterior, son: CU-06, instalar un paquete de reconocimiento en el gestor local; CU-08: eliminar un paquete de reconocimiento en el gestor local; y CU-10, solicitar dirección del paquete. La instalación de un paquete de reconocimiento en el gestor local, es una actividad que necesita de dos elementos fundamentales: la lista de posibles paquetes (tareas de reconocimiento) y la dirección del paquete a instalar. Este caso de uso, es uno de los más importantes ya que permite al robot utilizar los servicios de reconocimiento que ofrece ese paquete de manera local, y por ende con un mejor tiempo de respuesta. Por otro lado, el robot puede eliminar los paquetes de reconocimiento que tiene instalados, esto por si en algún momento requiere liberar espacio en su disco duro.

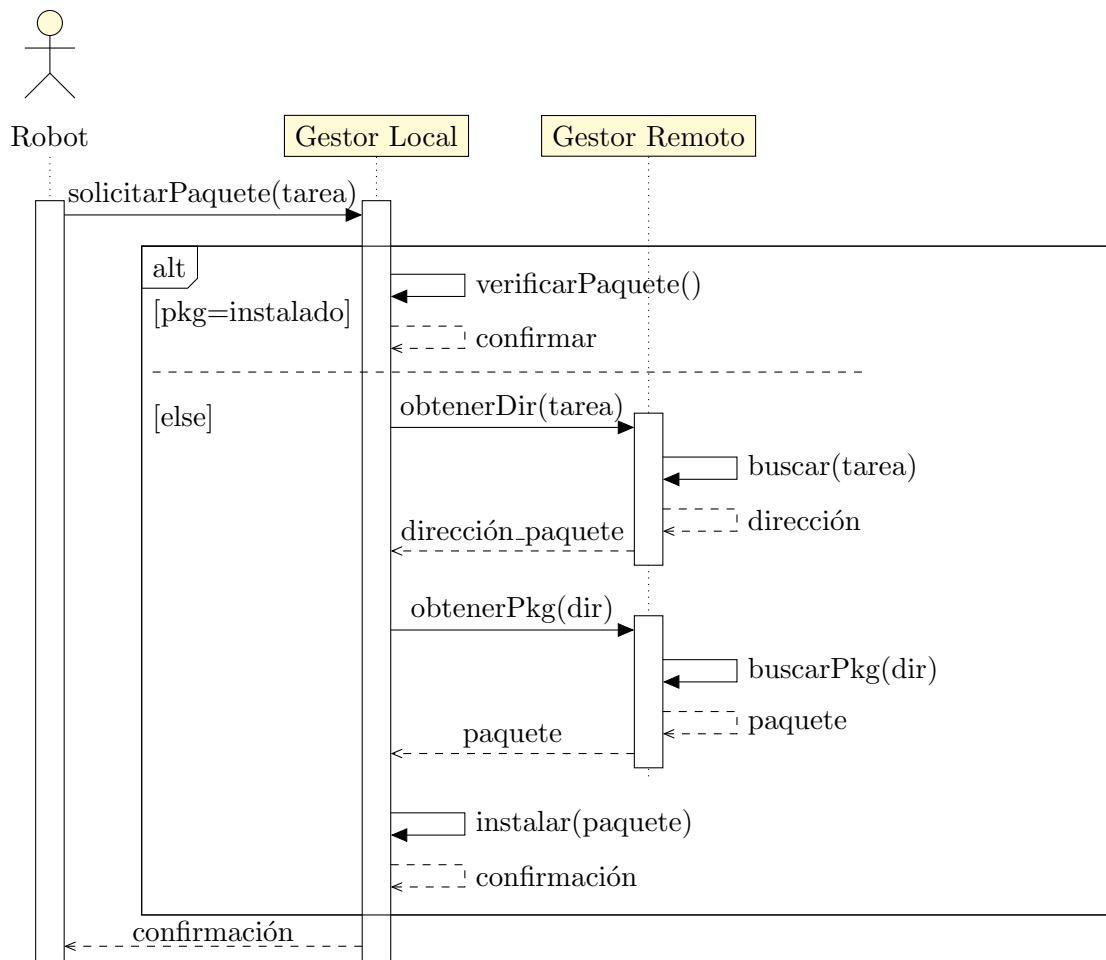


Figura 3.8: Diagrama de secuencia CU-06: Instalar un paquete de reconocimiento en el gestor local

En la Figura 3.8, se puede observar como el robot utiliza al gestor local como intermediario para instalar un paquete de reconocimiento proveniente del gestor remoto. El diagrama de secuencia anterior se puede resumir como sigue: se verifica que el paquete solicitado no se encuentre ya instalado, se solicita la dirección del paquete solicitado al gestor remoto, se obtiene el paquete utilizando la dirección y finalmente se instala el paquete en el gestor local. Por otro lado, la instalación de paquetes de reconocimiento en el gestor remoto, es una actividad que realiza el operador utilizando las interfaces de ROS para ubicar y desempaquetar los archivos del paquete en un directorio específico.

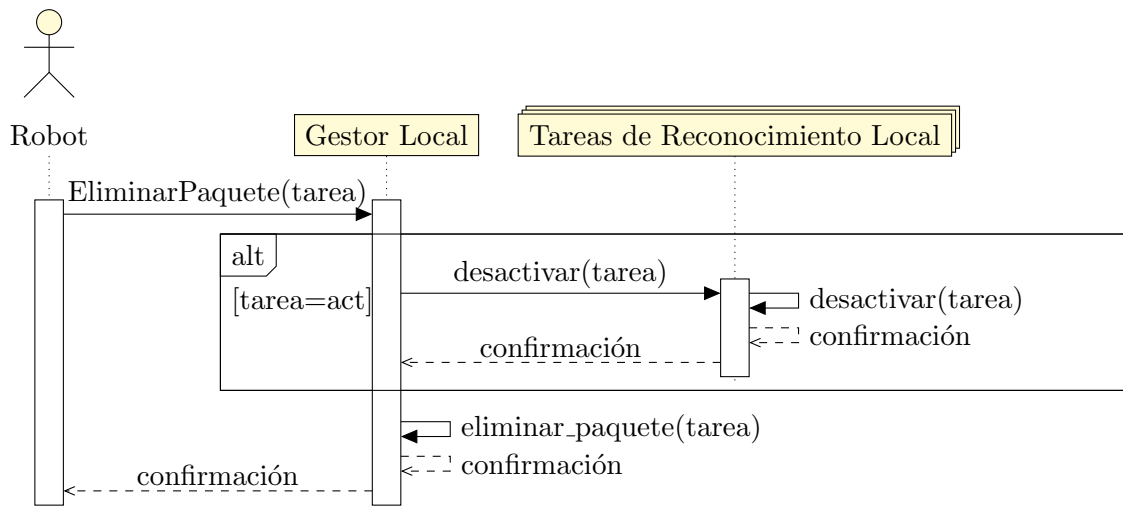


Figura 3.9: Diagrama de secuencia CU-08: Eliminar un paquete de reconocimiento en el gestor local

La eliminación de paquetes de reconocimiento local y remoto son tareas muy similares, básicamente lo único que cambia es el actor que realiza estas tareas (ver Figura 3.9). Para eliminar un paquete de reconocimiento es necesario verificar si existe una tarea activa asociada al mismo. Si este es el caso, se debe desactivar dicha tarea y luego proceder a eliminar el paquete; caso contrario, se elimina directamente.

3.1.3.3. Servicios de reconocimiento local y remoto

Una vez que las tareas asociadas a los paquetes de reconocimiento instalados se encuentren activas, se debe ofrecer todos estos servicios al robot. Todas las tareas de reconocimiento activas deben estar disponibles para ser utilizadas por el robot, incluyendo aquellas que se encuentran en el gestor local y en el gestor remoto. Para que el robot pueda realizar la solicitud de una tarea de reconocimiento, debe poseer la lista de tareas de reconocimiento (local o remota), para que de esta manera pueda conocer el repertorio de tareas de las cuales puede disponer.

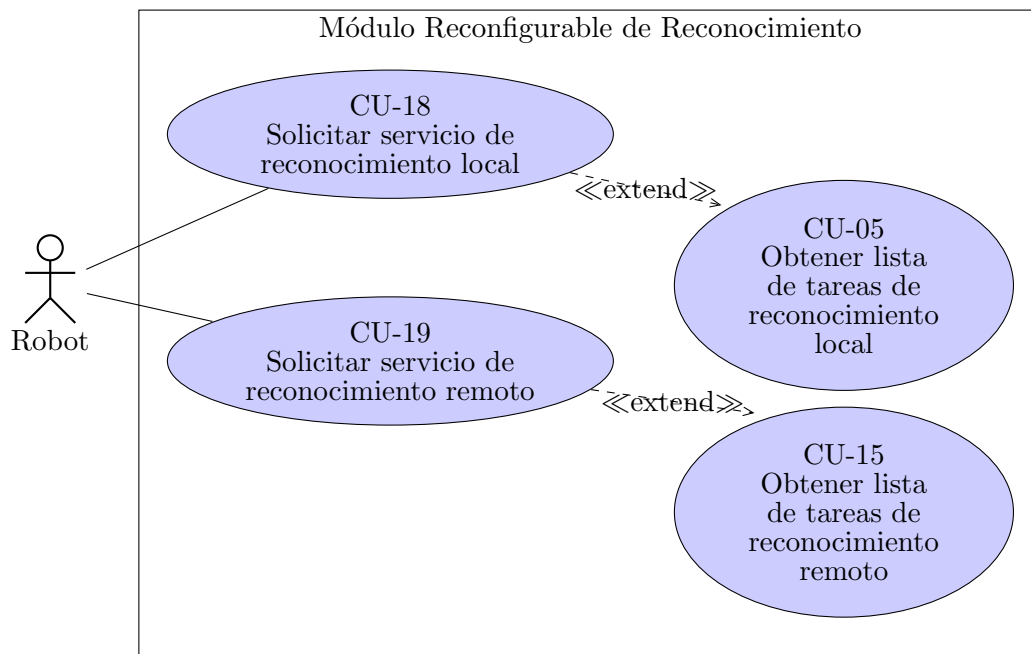


Figura 3.10: Diagrama de caso de uso: Solicitud de servicios de reconocimiento

Los casos de uso más resaltantes del diagrama anterior (ver Figura 3.10), son: CU-18, Solicitud de servicio de reconocimiento local y CU-19, solicitud de reconocimiento remoto. Ambas actividades son muy similares, no obstante, existen varias compensaciones a la hora de seleccionar un servicio de reconocimiento local o remoto. En el caso de solicitar servicios locales, su tiempo de respuesta es mucho mayor en comparación a los servicios remotos; en el caso de los servicios remotos, su procesamiento se realiza de manera externa por lo que los recursos del robot no se agotan o sobrecargan.

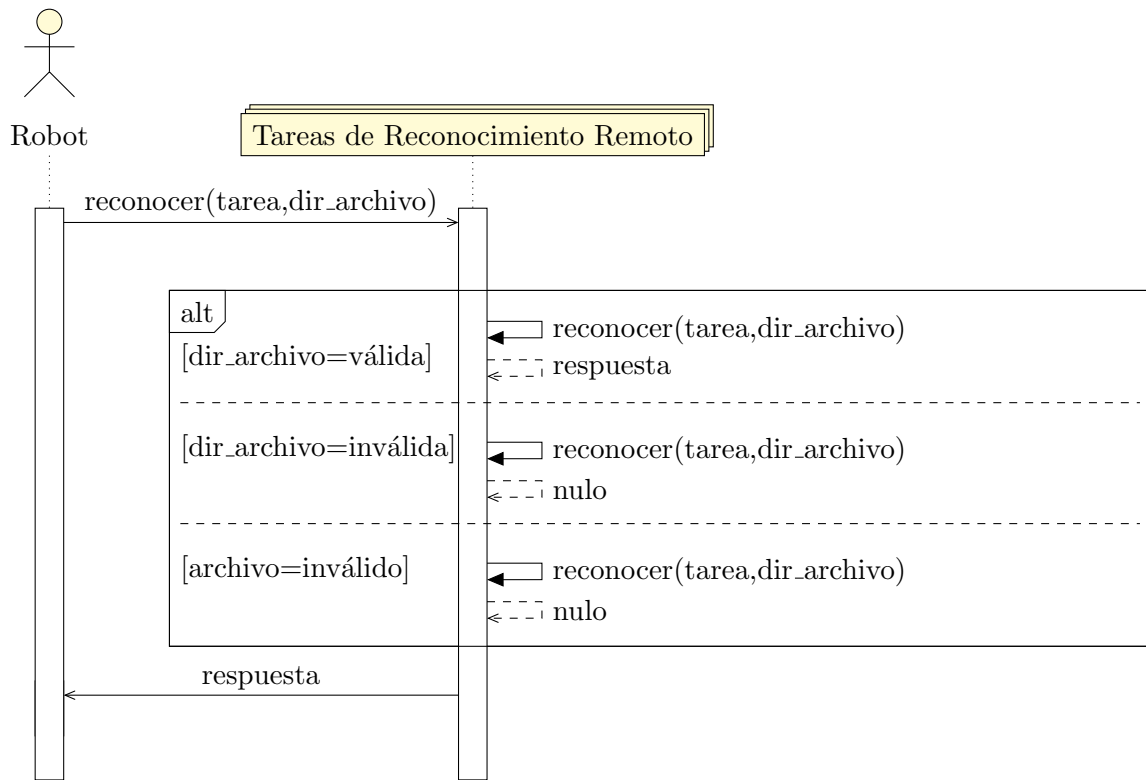


Figura 3.11: Diagrama de secuencia CU-19: Solicitar servicio de reconocimiento remoto.

En la Figura 3.11, se puede observar el diagrama de secuencia correspondiente al caso de uso CU-19: solicitar servicio de reconocimiento remoto. Para realizar una solicitud de reconocimiento remoto, se debe indicar el identificador de la tarea y la dirección del archivo a reconocer. Si la dirección o el archivo es inválido, el resultado del reconocimiento es nulo, de otra manera, la dirección del archivo es llevado a la tarea correspondiente y es devuelto el resultado al robot. Para la solicitud de servicios de reconocimiento local, el diagrama de secuencia es similar.

3.1.4. Requisitos no funcionales

Los requisitos no funcionales de la aplicación se pueden dividir en 3: requisitos de ejecución, restricciones de diseño y otros. Los requisitos de ejecución, son parámetros establecidos que necesitan ser cumplidos para que el sistema o un subconjunto de requisitos funcionales en general se ejecute de manera correcta; las restricciones de diseño son todos aquellos requisitos no funcionales que restringen el diseño del producto, por ejemplo: sistema operativo, hardware, puertos de entrada, entre otros. Por último, otros requisitos; esta sección contiene todos los demás requisitos que no entran en la sección anterior como por ejemplo: tecnologías a utilizar, selección de colores para las interfaces, restricciones de tiempo, entre otros.

3.1.4.1. Requisitos de ejecución

La principal y única restricción de ejecución, es que el Módulo reconfigurable de reconocimiento debe ejecutarse en 2 computadoras. Una debe representar el procesamiento interno del robot (gestor local) y la otra representar el procesamiento externo del mismo (gestor remoto).

3.1.4.2. Restricciones de diseño

Para realizar la implementación del Módulo reconfigurable de reconocimiento, se debe utilizar el sistema operativo robótico (ROS). Específicamente la version Kinetic Kame disponible para la versión 16.04 Xenial Xerus de Ubuntu. Las especificaciones de hardware mínimo para este sistema operativo pueden encontrarse en la sección A.3.7 del documento de especificación de requisitos.

3.1.4.3. Otros requisitos

A continuación se presentan de manera breve la especificación y requerimientos de la estructura general y contenido de los paquetes de reconocimiento.

1. Todos los archivos deben ser compatibles con python 3.
2. Estructura general de un paquete de reconocimiento debe ser la siguiente:

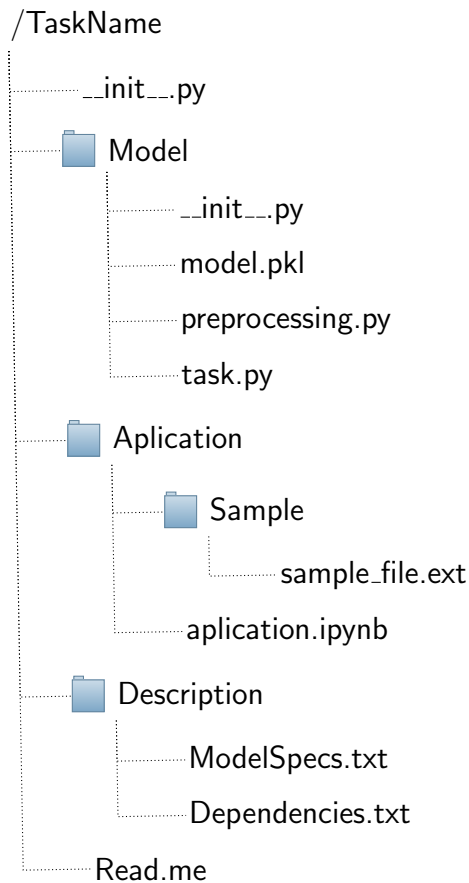


Figura 3.12: Árbol de directorios de un paquete de reconocimiento.

En la Figura 3.12, se puede observar la estructura general de un paquete de reconocimiento. Todos estos paquetes deben seguir una estructura similar para poder ser usados en el Módulo reconfigurable de reconocimiento. A continuación se describen de manera breve todos los elementos que conforman un paquete de reconocimiento:

- **TaskName:** Es el directorio principal, este tiene como nombre el identificador de la tarea de reconocimiento. Esta conformado por 3 carpetas: Model, Application y Description. Adicionalmente, contiene dos archivos: `__init__.py` y `Read.me`.
- **Model:** Es un directorio en el cual se almacenan los archivos que contienen el pre-procesamiento de los datos, la versión persistente del modelo o algoritmo entrenado y el archivo de código python correspondiente a la clase task (ver Figura 3.14), que permiten integrar los elementos anteriores para realizar las tareas de reconocimiento. Adicionalmente, contiene el archivos: `__init__.py`.
- **Application:** Es un directorio en el cual se genera una pequeña prueba de funcionamiento del algoritmo. Estos archivos se utilizan para que el operador pueda comprobar el funcionamiento del mismo antes de ser instalado. Para realizar esta prueba, es necesario utilizar un archivo `.ipynb`, correspondiente a un jupyter notebook; esto con la finalidad de reproducir de manera sencilla las pruebas sobre el algoritmo entrenado. Para realizar estas pruebas es necesario utilizar los archivos contenidos en el directorio **Model** y además es necesario añadir una o más muestras (directorio **Sample**), para mostrar detalladamente el pre-procesamiento de la misma en el jupyter notebook.
- **Description:** Es un directorio que contiene dos archivos con especificación detallada del algoritmo de reconocimiento y sus dependencias. El archivo **ModelSpecs.txt** contiene la información de: número de clases, formato de entrada, tipo de salida, nombres de salidas, hiperparámetros, tasa de reconocimiento, entre otros, por otro lado, el archivo **Dependencies.txt**, contiene las dependencias del paquete de reconocimiento.

Para obtener información más detallada sobre la especificación de los paquetes de reconocimiento se puede ir a la sección A.3.8, del documento de especificación de requisitos. A continuación se presentan ejemplos concretos de los paquetes de reconocimiento presentes en el caso de estudio (ver Figura 3.13).

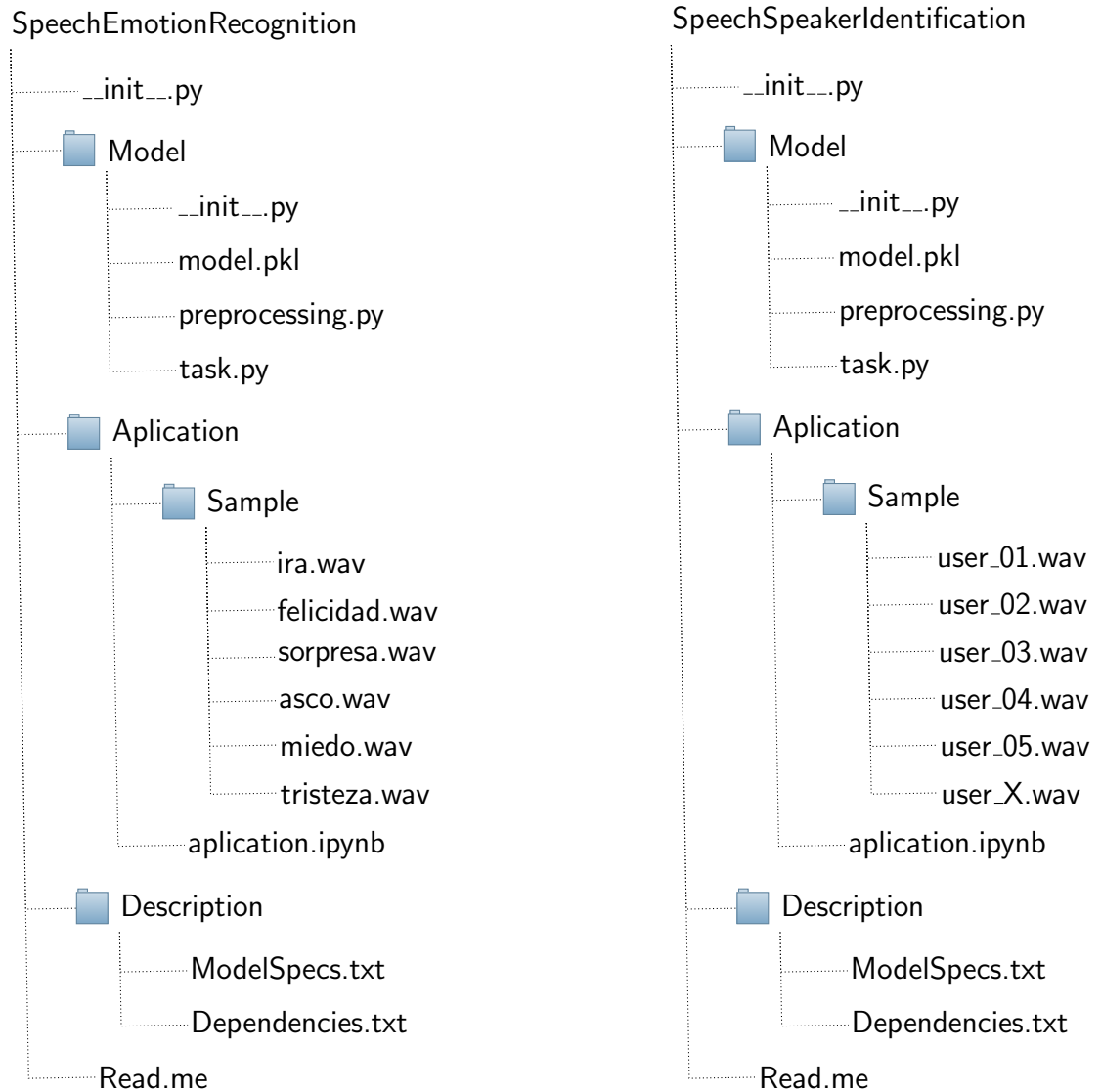


Figura 3.13: Paquetes de reconocimiento para los casos de estudio: reconocedor de emociones y reconocimiento de personas

Para permitir una alta cohesión entre los componentes de la arquitectura, el archivo **task.py** debe contener una clase con el nombre “RosTask” que implemente los métodos de la siguiente clase abstracta (ver Figura 3.14):

Task
<i>+ execute(String dir): String</i>
<i>- restore_model(String dir): Model</i>
<i>- preprocessing(String dir) : Vector</i>
<i>- prediction(Vector vc): String</i>

Figura 3.14: Clase abstracta: Task

A continuación se presentan en la Figuras 3.15 y 3.16, ejemplos de los archivos **Dependencies.txt** y **ModelSpecs.txt** respectivamente.

```

abs1-py==0.2.0
asn1crypto==0.24.0
audioread==2.1.5
cyclor==0.10.0
decorator==4.3.0
deepaffects==1.1.1
entrypoints==0.2.3

```

Figura 3.15: Ejemplo del archivo **Dependencies.txt**, generado mediante el comando “pip freeze”

```

1 Algoritmo de aprendizaje de máquina: Support Vector Machine
2 Número de clases: 6
3 Hiperparámetros: c= 1, r =0.01, kernel= rbf
4 Tasa de reconocimiento = 80%
5 Muestras de entrenamiento = 150
6 Muestras de prueba = 850
7 Nombre de las bases de datos usadas = Se creó una propia.
8 Formato de entrada = wav
9 Tipo de reconocimiento = Emociones
10 Tipo de salida = discreta
11 Nombre de las clases = Ira , Felicidad , Tristeza , Sorpresa ,
Asco , Miedo

```

Figura 3.16: Ejemplo del archivo **ModelSpecs.txt**

Capítulo 4

Diseño

En este capítulo se presentan todos los aspectos del diseño e implementación del Módulo Reconfigurable de Reconocimiento para la interacción humano-robot. En particular, se diseñarán cada uno de los componentes principales del módulo, basándose en el análisis de requisitos descrito en el capítulo anterior. Para realizar la implementación de estos módulos se utilizará el framework ROS [59].

El Módulo Reconfigurable funciona utilizando diversas tareas de reconocimiento, las cuales son implementadas mediante algoritmos de aprendizaje de máquina. Una revisión de los algoritmos utilizados para realizar las tareas descritas en el caso de estudio es realizada. Adicionalmente, se detallará la selección y justificación de los algoritmos utilizados para realizar las tareas presentes en el caso de estudio de este trabajo de investigación.

Los algoritmos de aprendizaje de máquina, necesitan datos para ser entrenados. En este trabajo de investigación se presentan 3 bases de datos, las cuales siguen diversos criterios de diseño y construcción. Una revisión de las bases de datos disponibles para realizar las tareas de reconocimiento presentes en el caso de estudio es realizada. Finalmente, se presenta de manera detallada el diseño de las bases de datos creadas para realizar este trabajo de investigación.

4.1. Diseño del Módulo Reconfigurable de Reconocimiento

El diseño de la arquitectura propuesta para el Módulo Reconfigurable de Reconocimiento está compuesto de dos componentes principales: el gestor local y el gestor remoto (ver Figura 4.1). Los objetivos de cada uno de estos componentes siguen una orientación distinta, no obstante, su funcionalidad es similar.

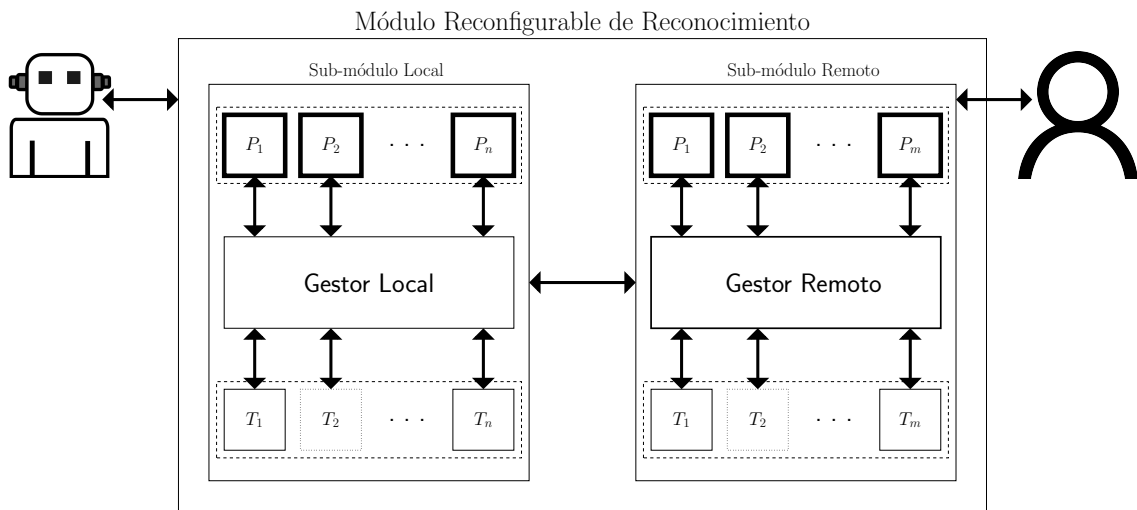


Figura 4.1: Diagrama de bloques del Módulo Reconfigurable de Reconocimiento

En el capítulo de análisis de requisitos (ver Capítulo 3), se puede encontrar de manera concreta todos los requisitos del Módulo Reconfigurable de Reconocimiento. Adicionalmente, el documento de especificación de requisitos del módulo se encuentra como anexo de este trabajo de investigación. A continuación se presentan los componentes principales (gestores: local y remoto), detallando los casos de uso asociados a cada uno de ellos.

4.1.1. Casos de uso del Módulo Reconfigurable de Reconocimiento

Los casos de uso del Módulo Reconfigurable de Reconocimiento pueden ser divididos en 3 grupos: gestor local, gestor remoto y servicios de reconocimiento. Las funcionalidades asociadas al gestor local, comprenden gran parte de las interacciones del robot, mientras que, las funcionalidades asociadas al gestor remoto comprenden gran parte de las interacciones con el operador. Otras funcionalidades están relacionadas con los servicios de reconocimiento que manejan ambos gestores. En la Tablas 4.1, 4.2 y 4.3, se pueden encontrar los casos de uso que se corresponden con las funcionalidades del gestor local, gestor remoto y los servicios de reconocimiento, respectivamente.

Tabla 4.1: Casos de usos relacionados con el gestor local

Caso de uso	Identificador
UC-01	Activar tarea de reconocimiento local
UC-02	Activar lista de tareas de reconocimiento local
UC-03	Desactivar tarea de reconocimiento local
UC-04	Desactivar lista de tareas de reconocimiento local
UC-05	Obtener lista de tareas de reconocimiento local
UC-06	Instalar un paquete de reconocimiento local
UC-07	Instalar una lista de paquetes de reconocimiento local
UC-08	Eliminar un paquete de reconocimiento en el gestor local
UC-09	Eliminar una lista de paquetes de reconocimiento en el gestor local
UC-10	Solicitar dirección de paquete
UC-15	Obtener lista de tareas de reconocimiento remoto

Tabla 4.2: Casos de usos relacionados con el gestor remoto

Caso de uso	Identificador
UC-11	Activar tarea de reconocimiento remoto
UC-12	Activar lista de tareas de reconocimiento remoto
UC-13	Desactivar tarea de reconocimiento remoto
UC-14	Desactivar lista de tareas de reconocimiento remoto
UC-15	Obtener lista de tareas de reconocimiento remoto
UC-16	Instalar un paquete de reconocimiento en el gestor remoto
UC-17	Eliminar un paquete de reconocimiento en el gestor remoto

Tabla 4.3: Casos de usos relacionados con los servicios de reconocimiento

Caso de uso	Identificador
UC-18	Solicitar servicio de reconocimiento local
UC-19	Solicitar servicio de reconocimiento remoto

4.1.2. Implementación del Módulo Reconfigurable de Reconocimiento

El Módulo Reconfigurable de Reconocimiento será implementado mediante el framework ROS [59]. A continuación se presentan cada uno de sus componentes, haciendo énfasis en su diseño en el framework ROS y en sus correspondientes casos de uso.

4.1.2.1. Implementación del gestor local

Los casos de uso relacionados con el gestor local se pueden dividir en dos: los que son utilizados para interactuar con el robot y los que son utilizados para interactuar con el gestor remoto.

Los casos de uso: CU-01, CU-02, CU-03, CU-04, CU-05, CU-06, CU-07, CU-08 y CU-09. Son parte de la funcionalidad de gestor local, y su función principal es proveer al robot con herramientas para gestionar las tareas de reconocimiento de su procesamiento interno. El gestor local es implementado mediante un nodo del grafo computacional de ROS, y los casos de uso mencionados anteriormente, son implementados mediante servicios de nodo ROS.

Por otro lado, los casos de uso: CU-10 y CU-15, están orientados a la interacción con el gestor remoto. Particularmente, estos casos de uso involucran información sobre las tareas de reconocimiento del gestor remoto, que deben ser consistentes en ambos gestores. Para almacenar y compartir esta información entre los gestores, se utiliza los parámetros del servidor de ROS, cuya función principal es precisamente permitir el acceso a esta información entre los nodos del grafo computacional.

4.1.2.2. Implementación del gestor remoto

Los casos de uso relacionados con el gestor remoto: CU-11, CU-12, CU-13, CU-14, CU-15, CU-16 y CU-17; comprenden interacciones que se desarrollan mayormente entre el gestor remoto y el operador. Estas funcionalidades se encargan de ofrecer al operador la posibilidad de gestionar las tareas de reconocimiento en el procesamiento externo. El gestor remoto es implementado mediante un nodo del grafo computacional de ROS, y los casos de uso mencionados anteriormente, son implementados mediante servicios de nodos ROS.

4.1.2.3. Implementación de los servicios de reconocimiento

Una de las funciones principales del Módulo Reconfigurable de Reconocimiento es proveer servicios de reconocimiento locales y remotos al robot. El gestor local y el remoto cumplen esta tarea. Particularmente, los casos de uso: CU-10 y CU-15 están relacionados con esta funcionalidad.

Para realizar la implementación de los servicios de reconocimiento, se utilizaron los mensajes de ROS. Los mensajes en ROS son enviados mediante tópicos, que permiten desacoplar la entrega y recepción de solicitudes. Esto permite, que el robot pueda realizar diversas solicitudes de reconocimiento asíncronas y continuar con la interacción sin bloquear el hilo principal de ejecución. Cada una de las tareas disponibles en los servicios de reconocimiento se corresponde con un nodo del grafo computacional de ROS; estos son creados de forma dinámica mediante un nodo genérico diseñado exclusivamente para utilizar los paquetes de reconocimiento. Cuando se realiza la activación de una tarea de reconocimiento, un nuevo nodo es creado por el gestor (local o remoto) y desde ese momento empieza a estar disponible para responder solicitudes de reconocimiento.

Para utilizar los servicios de reconocimiento son necesarios dos pasos:

- **Envío de solicitud:** El robot realiza el envío de una solicitud de reconocimiento local o remota. Cada uno de los nodos genéricos correspondientes a tareas de reconocimiento están a la espera de un mensaje mediante un tópico con el siguiente formato: `/gl_nombre_tarea` ó `/gr_nombre_tarea`, donde el prefijo

gl hace referencia al gestor local y el prefijo **gr** hace referencia al gestor remoto. Para realizar el envío de dicha solicitud, se debe crear una instancia de la clase `SendMessage` y transformar en binario (Ver Figura 4.2); en esta instancia se debe especificar el identificador de la solicitud y la dirección del archivo o el archivo binario que se corresponda con la entrada de la tarea de reconocimiento.

SendMessage
+ <code>get_file_id(): String</code> + <code>get_file_dir(): String</code> + <code>get_file_bin(): Bytes</code> + <code>set_file_id(String): void</code> + <code>set_file_dir(String): void</code> + <code>set_file_bin(Bytes): void</code>
- <code>file_id: String</code> - <code>file_dir: String</code> - <code>file_bin: Bytes</code>

Figura 4.2: Servicios de reconocimiento: clase correspondiente a la solicitud.

- **Recepción de solicitud:** Una vez que se procesa la entrada del reconocimiento, cada uno de los nodos genéricos envía la respuesta de reconocimiento al robot mediante el tópico **/reconocimiento**. La respuesta es una instancia en binario de la clase Message (ver Figura 4.3), y ésta contiene información sobre: la tarea de reconocimiento que respondió la solicitud, y dos marcas de tiempo que se corresponden con el inicio y fin del procesamiento y reconocimiento del archivo que se envió.

Message
+ get_answer(): String + get_timestamp_request_prediction(): String + get_timestamp_response_prediction(): String + get_file_id(): String + set_answer(String): void + set_timestamp_request_prediction(String): void + set_timestamp_response_prediction(String): void + set_file_id(String): void
- answer: String - timestamp_request : String - timestamp_response: String - file_id: String

Figura 4.3: Servicios de reconocimiento: clase correspondiente a la respuesta.

En la Figura 4.4, se puede observar el diagrama correspondiente al envío de solicitud de reconocimiento. El robot utiliza los tópicos a los cuales subscriben los nodos que se corresponden con las tareas de reconocimiento de ambos gestores (gl_tarea1, gl_tarea2, gr_tarea1, y gr_tarea2) para solicitar los servicios de reconocimiento. Los nodos correspondientes a estas tareas de reconocimiento pueden estar ubicados en el gestor local o remoto.

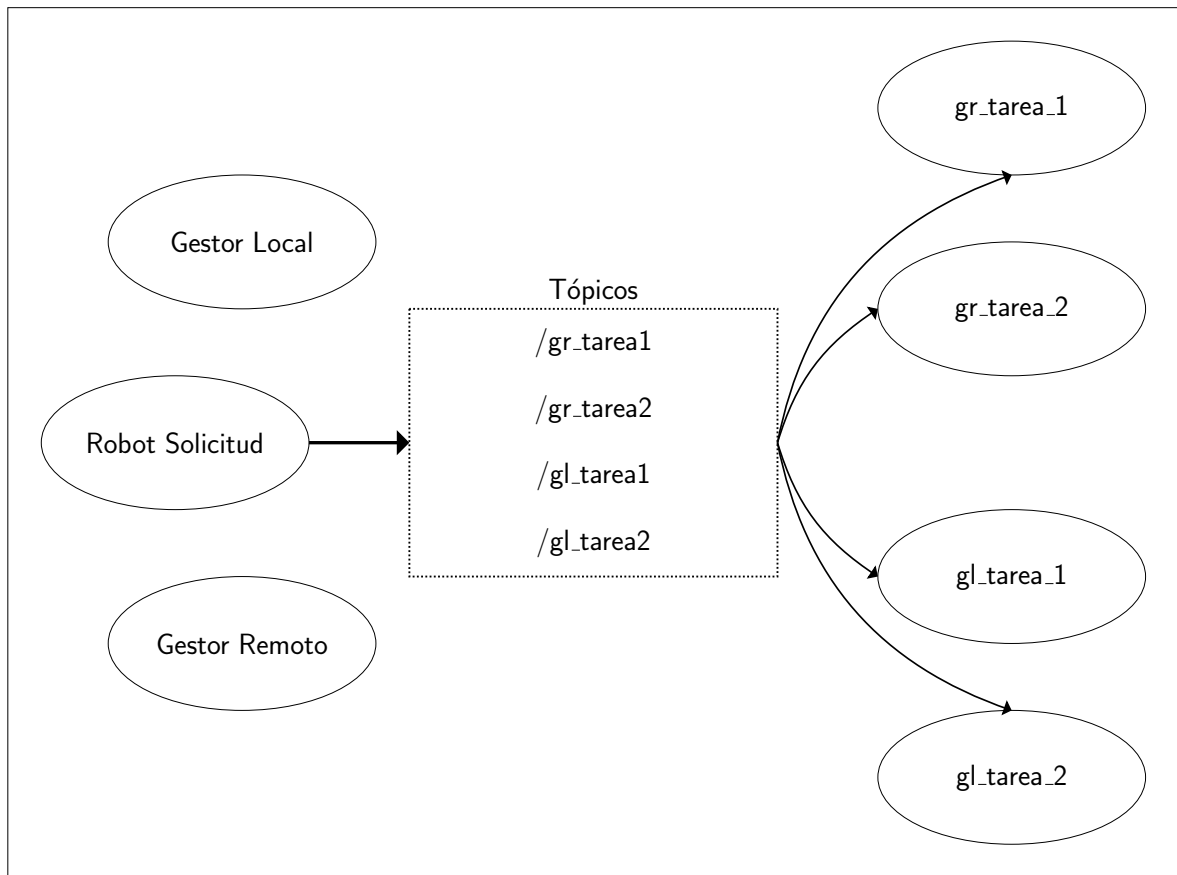


Figura 4.4: Servicios de reconocimiento: envío de solicitud de reconocimiento

En la Figura 4.5, se puede observar el diagrama correspondiente a la recepción de respuestas de reconocimiento. Cuando los nodos correspondientes a las tareas de reconocimiento del gestor local o remoto finalizan su procesamiento, envían la respuesta correspondiente al reconocimiento (ver Figura 4.3), mediante un tópico al cual subscribe el nodo correspondiente al robot.

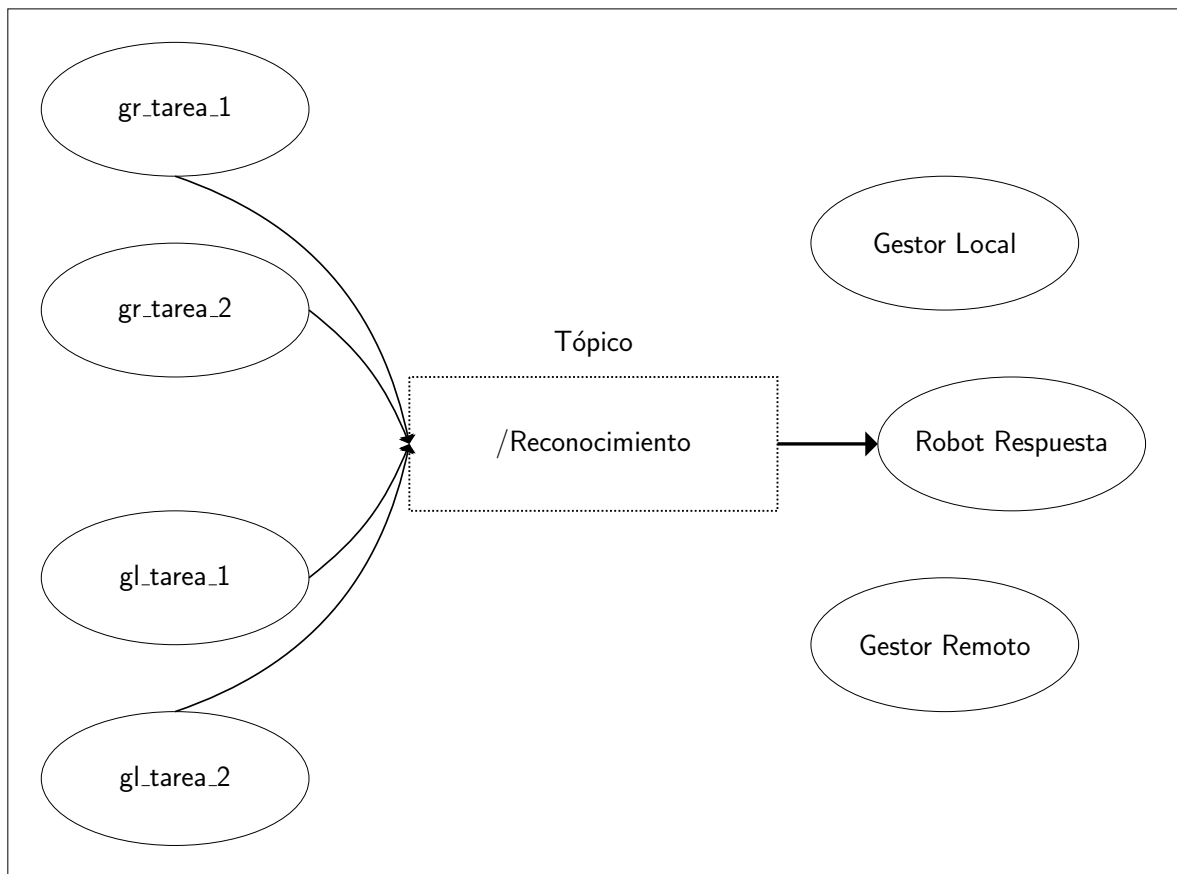


Figura 4.5: Servicios de reconocimiento: recepción de respuestas de reconocimiento

4.2. Diseño de la base de datos para el reconocimiento de emociones

Una de las tareas más complejas en las investigaciones del reconocimiento de emociones mediante la voz, es la construcción de la base de datos. Las bases de datos orientadas a este tipo de tareas enfrentan diferentes tipos de desafíos, tales como: la selección de frases que permitan representar las emociones, la complejidad de la captura de la expresión, y adicionalmente, la consistencia de las expresiones y su validación de forma objetiva. El diseño y construcción de esta base de datos forma parte de investigaciones anteriores en esta área [50]. A continuación se presenta los criterios utilizados para la construcción de esta base de datos.

Las bases de datos emocionales, pueden ser clasificadas en tres tipos según la forma en que se pide a las personas expresar las emociones [17]:

- **Lenguaje actuado:** Se pide a los actores expresar directamente una emoción predefinida.
- **Lenguaje de la vida real:** Respuestas naturales de conversaciones, las cuales son auténticas por naturaleza.
- **Lenguaje emocional evocado:** Las emociones son inducidas y son auto-reportadas en lugar de ser etiquetadas, es decir, la persona reconoce su propia emoción y le asigna por sí mismo una etiqueta.

Entre las bases de datos que se basan en lenguaje de la vida real se tiene la base de datos: “Polish Emotional Natural Speech Database” [30] y “Automatic Classification of Emotion-Related User States in Spontaneous Children Speech” [67]. Basada en lenguaje actuado: “A Database of German Emotional Speech” [11]; y basada en la evocación de emociones: “The eNTERFACE’05 audio-visual emotion database” [46] y “Surrey Audio-Visual Expressed Emotion (SAVEE) database” [35].

Actualmente, gran parte de las investigaciones relacionadas al reconocimiento de emociones no son rigurosas con el tipo de pruebas que hacen a los modelos. Por esta razón, y en aras de mantener un mejor control sobre las condiciones de ambiente,

en esta investigación se construirán 2 bases de datos orientadas al reconocimiento de emociones: la primera, en condiciones controladas y actuadas; y la segunda, en condiciones controladas y semi-naturales.

4.2.1. Bases de datos en condiciones controladas y actuadas

Para realizar la construcción de esta base de datos se realizó la selección del conjunto de declaraciones por cada una de las emociones basándose en [46], las cuales fueron sometidas a 3 tipos de validación: validación de las declaraciones de forma textual, validación por parte de los participantes de la base de datos y validación por parte de un jurado de 4 personas.

La validación de las declaraciones de forma textual, se realizó mediante una encuesta en las cuales participaron 96 personas (66 hombres y 30 mujeres). A cada una de las personas se les presentó un conjunto de declaraciones por cada una de las emociones: ira, sorpresa, felicidad, tristeza, asco y miedo. La encuesta consistía en seleccionar aquellas declaraciones con las cuales expresarían cada una de las emociones. Las declaraciones con las cuales se sintieron identificados gran parte de los participantes de la encuesta fueron: ira, tristeza, felicidad, miedo y sorpresa. Las declaraciones utilizadas para expresar el asco fueron las menos seleccionadas. Adicionalmente, se les pidió a los participantes sugerir qué declaraciones utilizarían ellos para expresar cada emoción. Luego cada una de las declaraciones, junto con las sugerencias de los participantes fueron seleccionadas por 2 jueces y el conjunto de declaraciones resultante fue el siguiente:

Tabla 4.4: Conjunto de declaraciones de cada emoción

Ira	<ol style="list-style-type: none"> 1) ¿Qué te pasa? 2) ¡Eso a mi que me importa! 3) ¡O te vas o te boto! 4) ¿Me vas a atender o no? 5) ¿Sabes qué? ¡Déjalo así! 6) No me molestes!
Sorpresa	<ol style="list-style-type: none"> 1) ¡No puede ser! ¿En serio? 2) ¡Qué! ¡Yo no sabia eso! 3) ¡Jamás lo hubiera creído! 4) ¡No me lo esperaba! 5) ¡No te creo! ¿De verdad? 6) ¿De verdad? ¡No sabia! 7) ¿Es en serio?
Felicidad	<ol style="list-style-type: none"> 1) ¡Gané! 2) ¡Qué genial! Pase! 3) ¡No me lo creo! ¡qué suerte! 4) ¡Lo logre! ¡Al fin! 5) ¡No puede ser! ¡qué bien! 6) ¡No lo creo! ¡Funciona!
Miedo	<ol style="list-style-type: none"> 1) No, no me hagas daño 2) Ya no tengo más, no tengo nada. 3) No, no me robes 4) Aléjate, Aléjate 5) Aléjate por favor 6) no, por favor
Asco	<ol style="list-style-type: none"> 1) Esto si está feo! 2) ¿Qué hay en el plato? 3) ¡Qué repugnante! 4) ¿Qué asco? ¿Qué es esto? 5) ¡Un bicho! 6) ¿Qué es esto?
Tristeza	<ol style="list-style-type: none"> 1) Todo iba tan bien, no sé qué paso 2) Lo/La extraño pero se fue 3) Ya no será lo mismo 4) Dime que no es verdad 5) Aún sentía algo por ella 6) El/Ella fue parte de mi vida 7) No pude hacerlo

El proceso de grabación se realizó en una oficina, con poco ruido. Adicionalmente, todos los participantes fueron ubicados en un mismo sitio para grabar, a una distancia de 40 centímetros del micrófono. El proceso de grabación fue realizado de la siguiente manera:

- A cada uno de los participantes se les pidió sentarse en una silla ubicada a 40 centímetros del micrófono.
- A cada participante se le pidió leer el conjunto de declaraciones de cada emoción, luego, se le pidió reproducir (actuar) cada una de las declaraciones de cada emoción 4 veces de distintas maneras.
- En caso de no expresar correctamente alguna declaración o de que el participante no estuviese satisfecho con el resultado, se le pedía al participante repetir dicha emoción utilizando como ayuda la orientación del operador o muestras de participantes anteriores.
- Para realizar las grabaciones fue utilizado el software Audacity [53]. Se utilizó un solo canal de grabación y la frecuencia de muestreo fue de 48 kHz.
- Luego del proceso de grabación cada una de las declaraciones fue seleccionada por el operador, el cual descartó aquellas declaraciones contaminadas (ruidos de golpes de mesa, movimientos de sillas, entre otros) o de poca calidad (declaraciones incompletas o ambiguas).
- Luego de seleccionadas las muestras, se recortaron cuidadosamente y se transformaron a 16 kHz. Adicionalmente, cada una de las muestras fue etiquetada.

La validación por parte de los participantes de la base de datos, consistió en reproducir cada una de las declaraciones del participante y hacerle dos preguntas por cada una de ellas: ¿Considera que en esta declaración se expresó la emoción? y ¿Considera que esta declaración pudiera interpretarse de otra manera?. Si algunas de las dos preguntas anteriores eran respondidas de manera negativa, se descartaba la muestra. En el caso particular en donde se descartaban todas las muestras de una declaración, se repetía el proceso de grabación.

Tabla 4.5: Escala utilizada para la validación de las muestras de audio

Significado	Valor
El audio es entendido e interpretado inequívocamente de una única manera	3
Para algunas personas podría tener otro significado	2
El audio es susceptible de ser entendido en sentidos diversos	1
El audio definitivamente se presta para múltiples interpretaciones.	0

Finalmente, la validación por parte de un jurado consistió en cuantificar la validez del contenido mediante la “V” de Aiken. El número de jurados fue 4. A cada uno de los jurados se les pidió calificar cada una de las muestras previamente filtradas por las validaciones anteriores según las preguntas de la Tabla 4.5:

Luego de realizar el etiquetado en base a la escala anterior, se obtuvo el coeficiente “V” de Aiken para cada una de las muestras.

$$V = \frac{S}{(n(C - 1))} \quad (4.2.1.1)$$

En la ecuación 4.2.1.1, el valor S , representa la suma de los valores de cada jurado por cada muestra. El valor n , representa el número de personas en el jurado, y el valor C , el número de valores en la escala de valoración. Las muestras cuyo cálculo de validación fue mayor a 0.75 fueron aceptadas y formaron parte de la base de datos. El resto de muestras fue descartado.

La base de datos en condiciones controladas y actuadas, fue conformada por un total de 1351 muestras. La frecuencia de las muestras por emoción puede verse en la Figura 4.6.

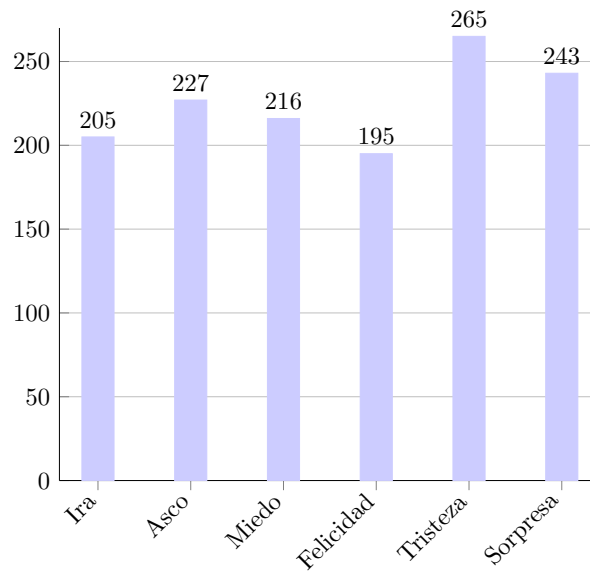


Figura 4.6: Frecuencia de las emociones de la base de datos en condiciones controladas y actuadas

4.2.2. Bases de datos en condiciones controladas y semi-natural

Para crear un conjunto que permitiera probar de manera rigurosa los modelos entrenados con la base de datos anterior, se realizó una base de datos en las mismas condiciones pero variando las palabras utilizadas inicialmente. Esta base de datos, de manera similar a la anterior, fue sometida a los 3 tipos de validación descritos en la sección anterior.

En aras de determinar la capacidad de los modelos para reconocer un amplio grupo de expresiones en las personas, esta base de datos consistió en expresar cada una de las declaraciones de cada emoción utilizando sus propias palabras. Es decir expresando el mismo significado y la emoción de la declaración original pero utilizando las palabras que utilizaría el participante en la vida cotidiana.

Todo el proceso de grabación fue similar al anterior, el único cambio que se realizó, se basó en que los participantes debían utilizar sus propias palabras para expresar 4 veces cada una de las declaraciones de cada emoción (ver Tabla 4.4).

La base de datos en condiciones controladas y semi-naturales, fue conformada por un total de 1163 muestras. La frecuencia de las muestras por emoción puede verse en

la Figura 4.7.

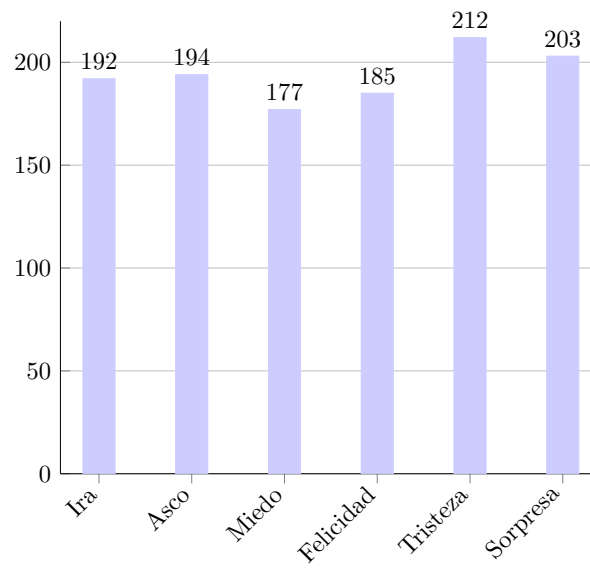


Figura 4.7: Frecuencia de las emociones de la base de datos en condiciones controladas y semi-naturales

4.3. Diseño de la base de datos para el reconocimiento de personas

Las bases de datos orientadas al reconocimiento de personas (identificación y verificación), siguen diversos criterios de construcción. Uno de los principales beneficios de diseñar y construir una base de datos propia es que permite la evaluación cualitativa de métodos y criterios para el reconocimiento de personas a través de la voz.

Las bases de datos orientadas al reconocimiento de personas mediante la voz, pueden ser clasificadas en 2 tipos según el número de sesiones utilizadas para obtener las muestras de audio:

- **Sesión única:** Las bases de sesión única, registran las muestras de audio de cada persona en una sola sesión, es decir, todas las muestras de cada participante son grabadas en un solo momento.
- **Sesión múltiple:** Las bases de datos de sesión múltiple, registran las muestras de audio de cada persona en múltiples sesiones, es decir, las muestras de cada participante son grabadas en diferentes momentos (días, semanas, meses). Estas bases de datos permiten capturar características intra-locutor, por lo que se consideran óptimas para el reconocimiento de personas.

Entre las bases de datos que se basan en única sesión se tiene la base de datos “TIMIT” [25]. Basadas en múltiples sesiones: la base de datos “YOHO” [12], la base de datos Polycost [32] y la base de datos “ELSDSR” [23]. Adicionalmente, existen bases de datos de sesión única y múltiple, donde se combinan usuarios cuyas muestras se grabaron en una única sesión y usuarios cuyas grabaciones se realizaron en múltiples sesiones [48].

En este trabajo de investigación se construirá 1 base de datos orientada al reconocimiento de personas de sesión única y múltiple. Esta base de datos estará dividida en dos conjuntos: entrenamiento y pruebas. Ambas en condiciones de ambiente controladas.

4.3.1. Construcción de la base de datos

Dentro de los objetivos principales en la creación de una base de datos en español para el reconocimiento de personas mediante la voz, es obtener muestras de voz representativas del español con respecto a la variabilidad inter-locutor e intra-locutor [23]. La población seleccionada para realizar las grabaciones la comprenden estudiantes de la Universidad de Los Andes, todos ellos venezolanos específicamente del estado Mérida.

4.3.2. Diseño y proceso de grabación

El conjunto frases y palabras ha sido diseñado para proveer muestras de audio para el desarrollo y evaluación de sistemas de reconocimiento de personas mediante la voz. El lenguaje del texto es Español, y fue leído por 8 participantes (7 hombres y 2 mujeres). Las grabaciones fueron realizadas en una oficina con poco ruido, donde a cada uno de los participantes se le otorgó una hoja que contenía cada una de las frases y palabras a leer. Se utilizó un micrófono convencional el cual se ubicó a 40 cm del participante. El proceso de grabación de la primera sesión fue realizado de la siguiente manera:

- A cada uno de los participantes se les pidió ubicarse en una silla, ubicada a 40 centímetros del micrófono.
- A cada participante se le pidió leer el conjunto de frases y palabras de cada una de las hojas (entrenamiento y pruebas) en un tono natural o neutral.
- Para realizar las grabaciones fue utilizado el software Audacity [53]. Se utilizó un solo canal de grabación y la frecuencia de muestreo fue de 48 kHz.
- Luego del proceso de grabación, cada una de las declaraciones fue seleccionada por el operador, el cual descartó aquellas declaraciones contaminadas (ruidos de golpes de mesa, movimiento de silla, entre otros). En algunos casos los participantes se equivocaban leyendo alguna declaración, en esos casos se repetía la declaración y se registraban ambas.

- Luego de seleccionadas las muestras, se recortaron cuidadosamente y se transformaron a 16 kHz. Adicionalmente cada una de las muestras fue etiquetada por participante.

Una vez finalizada la primera sesión se les pidió a los participantes repetir el proceso de grabación utilizando solamente el conjunto de entrenamiento, la segunda sesión fue realizada aproximadamente dos semanas después. Finalmente, se obtuvo un registro de 5 participantes de sesión múltiple y 3 de sesión única.

4.3.3. Diseño del conjunto de frases y palabras de entrenamiento y prueba

En aras de obtener un conjunto representativo de frases y palabras en español, se tomó la decisión de seleccionar un conjunto representativo de frases y palabras articuladas según el español venezolano [54]. Para realizar el conjunto de prueba se decidió utilizar un conjunto más general del español, por lo que se seleccionaron palabras diferentes basándose en libros de fonética y fonología del español [33, 60]. El conjunto de entrenamiento está conformado por frases y palabras cortas al igual que la base de datos TIMIT [25], y adicionalmente, esta base de datos está diseñada para capturar características intra-locutor. A continuación se presenta la selección de las frases utilizadas para la construcción del conjunto entrenamiento de la base de datos (Ver Tablas 4.6 – 4.25):

Tabla 4.6: Frases y palabras del conjunto de entrenamiento basadas en los fonemas del sistema vocálico del español

Fono	Frases y palabras	Fono	Frases y palabras
[i]	lima, país, sismo	[j]	bien, rabia, miau, aire, buey
[e]	semana, embuste, lento, puente	[ɛ]	eje, bueno, papel, peine, pesca
[a]	pasa, mar, pausa, alma	[a]	ahora, paja
[o]	comer, paso, lados, tosco	[ɔ]	roca, hoja, hoy, ahora
[u]	uña, tribu, baúl	[w]	causa, guau, cuatro, fui, gauira

Tabla 4.7: Frases y palabras de los fonemas del sistema vocálico del español: vocales nasalizadas

Realización	Frases y palabras
[ã][ẽ][ĩ][õ][ũ]	Insensato,entre, mamá, tango,mono, hombre

Tabla 4.8: Frases y palabras de los fonemas del sistema consonántico del español: realizaciones de los fonemas /p/ y /b/

Realización	Frases y palabras
[p]	puma, tapa, paño
[b]	bien, vamos, tumbar, un carbon, un velo
[β]	saber, volver

Tabla 4.9: Frases y palabras de los fonemas del sistema consonántico del español: neutralización de /p/ y /b/, realizaciones del archifonema /B/

Realización	Frases y palabras
[B], [β], [b], [p]	Eclipse, apto, ábside, obsceno

Tabla 4.10: Frases y palabras de los fonemas del sistema consonántico del español: Elisión de la d

Terminación	Frases y palabras
-ado	pescado, pescado
-ido	dormido, dormio
-udo	Peludo, peluo
-ita	Calientadita, calientaita

Tabla 4.11: Frases y palabras de los fonemas del sistema consonántico del español: realizaciones de los fonemas /t/ y /d/

Realización	Frases y palabras
[t]	lata, torre, bota
[d]	dámelo!, ¿Dónde?, mando, toldo, desde
[ð]	cuadro, nada

Tabla 4.12: Frases y palabras de los fonemas del sistema consonántico del español: neutralización de /t/ y /d/, realizaciones del archifonema /D/

Realización	Frases y palabras
[ð],[d],[t]	étnico, adjetivo, verdad, ciudad

Tabla 4.13: Frases y palabras de los fonemas del sistema consonántico del español: realizaciones de los fonemas /k/ y /g/

Realización	Frases y palabras
[k]	saco, querer, crater, kilómetro, kárate
[g]	gracias, gloton, tango, lengua, un gusano
[ɣ],[g̃],[k]	Agua, largo, el gusto, toga manga, gusto, gracias guerra, seguir

Tabla 4.14: Frases y palabras de los fonemas del sistema consonántico del español: neutralización de /k/ y /g/, realizaciones del archifonema /G/

Realización	Frases y palabras
[ɣ],[g̃],[k]	actúo, signo, zigzag, tic, zinc, frac

Tabla 4.15: Frases y palabras de los fonemas del sistema consonántico del español: neutralización de las oposiciones /p ~ b/, /t ~ d/ y /k ~ g/, realizaciones de los archifonemas /B/ /D/ /G/

Realización	Frases y palabras
[ʔ],[k], [g̃],[ɣ],[θ]	apto ['akto], objeto [ok'hɛto], étnico [e̞ɲniko] admirar [a̞m̩i'rar], acto ['akto], signo [si̞ɲno], pepsi [pe̞ʔsi]

Tabla 4.16: Frases y palabras de los fonemas del sistema consonántico del español: realizaciones de los fonema /tʃ/, /f/ y /s/

Realización	Frases y palabras
[tʃ]	chino, techo, mucho
[f]	felino, trifulca, afgano, naftalina
[s]	sabio, mesa, tres, encima, hacer, azul

Tabla 4.17: Frases y palabras de los fonemas del sistema consonántico del español: tipos de /s/

Realización	Frases y palabras
[h]	Los libros, ¿Quieres?
[θ]	Caracas

Tabla 4.18: Frases y palabras de los fonemas del sistema consonántico del español: la grafía X, archifonema /Gs/

Realización	Frases y palabras
[ks],[g̃],[ɣ̃]	taxi, examen
[s]	auxilio, exacto
[ks],[g̃],[ɣ̃],[h]	unisex, torax
[ks]	extranjero, extremo

Tabla 4.19: Frases y palabras de los fonemas del sistema consonántico del español: realizaciones del fonema /j/

Realización	Frases y palabras
[d͡ʒ]	cónyuge, el yeso, un llavero, yo!, llora!
[j]	mayo, ese yeso, la llave
[j]	playa, yunque, llover, calle

Tabla 4.20: Frases y palabras de los fonemas del sistema consonántico del español: realizaciones del fonema /h/

Realización	Frases y palabras
[h]	caja, gasté, reloj

Tabla 4.21: Frases y palabras de los fonemas del sistema consonántico del español:
realizaciones del fonema /m/, /n/ y /ɲ/

Realización	Frases y palabras
[m]	mesa, loma
[n]	nudo, cana
[ɲ]	ñame, niño

Tabla 4.22: Frases y palabras de los fonemas del sistema consonántico del español:
Asimilación general

Realización	Frases y palabras
[m]	cambio, un pie, álbum morado
[m̃]	enfermo, sin fe
[ñ]	once, un cero
[n̄]	canto, un diente
[n̄]	cansado, perenne, enlodar, honrado
[n̄ʲ]	ancho, conyugal, con ñame, sin llave
[ɲ]	tanque, tango, enjaular

Tabla 4.23: Frases y palabras de los fonemas del sistema consonántico del español:
Asimilación parcial

Realización	Frases y palabras
[m]	campo, samba, un pénom
[m̃]	enfermo
[n̄]	santo, donde
[n]	cansado, innovar, enredo
[nʲ]	concha, conyuge
[ŋ]	un alma, con interes, quiero pan

Tabla 4.24: Frases y palabras de los fonemas del sistema consonántico del español:
realizaciones del fonema /l/

Realización	Frases y palabras
[l]	luna, sala, el aire, dame sal, almirante
[l̄]	malta, aldea
[lʲ]	colchon, el yeso, el ñame

Tabla 4.25: Frases y palabras de los fonemas del sistema consonántico del español:
realizaciones de los fonemas /r/ y /r̄/, casos especiales

Fono	Frases y palabras	Fono	Frases y palabras
[r]	cero, caro, coro, foro	[r̄]	rosa, honrado, alrededor, israelí
[r̄]	cerro, carro, corral, forro	[r̃]	perro, burro, rama, enredo, pelar

Las palabras correspondientes a la neutralización de las oposiciones /p ~ b/, /t ~ d/ y /k ~ g/ (Ver Tabla 4.15) y la neutralización de los fonemas /t/ y /d/ (Ver Tabla 4.12), se grabaron 2 veces por sesión de manera concentrada y natural. En el caso de la grafía de la X (Ver Tabla 4.18), las palabras: “auxilio”, “exacto”, “extranjero” y “extremo”; se grabaron dos veces de maneras distintas utilizando los fonos [ks] y [s]. En el caso de los tipos de /s/ (Ver Tabla 4.17), se grabaron dos veces utilizando los fonemas [h] y [θ].

En total se recolectaron 484 muestras para 5 participantes de sesión múltiple (242 en cada sesión) y 242 para los otros 3 participantes de sesión única.

En la construcción del conjunto de prueba, se seleccionaron diversas frases y palabras equilibradas fonéticamente y seleccionadas en función de la clasificación clásica de vocales y consonantes. A continuación se presenta la selección de las frases y palabras utilizadas para la construcción del conjunto de entrenamiento de la base de datos.

Tabla 4.26: Frases y palabras del conjunto de prueba: vocales orales

Fonema	Frases y palabras	fonema	Frases y palabras
/i/	pito, tiza, ira, hijo, sismo.	/ĩ/	niño
/e/	ceda, semana, puente hecho, deshecho.	/ẽ/	nena
/a/	alto, cama, alameda, llama, haba.	/ã/	mano, ana
/o/	oso, color, hondo, alcohol.	/õ/	mono, hombre
/u/	lupa, luna, baul, humo, uña, tribu	/ũ/	nunca

Tabla 4.27: Frases y palabras del conjunto de prueba: semi-consonantes

fonema	Frases y palabras
/j/	hacia, diente, murió, ciudad
/w/	cuánto, cuerda, ruido, antiguo

Tabla 4.28: Frases y palabras del conjunto de prueba: semi-vocales

fonema	Frases y palabras
/i/	baile, veinte, hoy, aire, seis
/u/	causa, feudo, lo unió, aula, Europa, lo usó

Tabla 4.29: Frases y palabras del conjunto de prueba: consonantes oclusivas

fonema	Frases y palabras
/p/	ópera, copa, tapa, papá, paco
/t/	tan, lata, toro, tapa, metal
/k/	casa, queso, paquete, kilo, loco
/b/	vaso, bote, hombre, tumba, ambos
/d/	dedo, dolor, ronda, soldar
/g/	gasa, guiso, rango, hongo, guerra

Tabla 4.30: Frases y palabras del conjunto de prueba: fricativas

fonema	Frases y palabras
/f/	café, fe, ferrocarril, fama, trifulca
/s/	saco, passa, zapato, adición, decisión
/j/	hierba, mayo, cayado, yeso, hoyo
/h/	caja, gente , reloj

Tabla 4.31: Frases y palabras del conjunto de prueba: aproximantes

fonema	Frases y palabras
/d/	lo dan, los dan, arde, ese dedo, ese día
/b/	se van, desván, calvo, árbol, ese bote, las veces
/g/	hago, rasgo, algo, orgullo, los guisos, la guerra

Tabla 4.32: Frases y palabras del conjunto de prueba: africadas

fonema	Frases y palabras
/tʃ/	muchacho, chico, pecho, chicharrones, cuchara

Tabla 4.33: Frases y palabras del conjunto de prueba: consonantes nasales

fonema	Frases y palabras
/m/	mamá, cama, loma, marzo, asomar, mata
/n/	nudo, nuca, rana, lana, satén, común, ratón
/ɲ/	caña, leña, sueño, moño, año

Tabla 4.34: Frases y palabras del conjunto de prueba: alófonos nasales ante consonante

fonema	Frases y palabras
/m̩/	infame, un farol, confuso, énfasis
/n̩/	hongo, engranaje, un cuento, mango, hincó, ángel
/ɲ̩/	dónde, cuándo, duende, lento, canto, ando

Tabla 4.35: Frases y palabras del conjunto de prueba: consonantes líquidas laterales

fonema	Frases y palabras
/l/	balneario, balsa, colcha, el yeso, alto, caldo, calza

Tabla 4.36: Frases y palabras del conjunto de prueba: consonantes líquidas laterales

fonema	Frases y palabras
/r/	pero, pera, para, coro, quería, cero, vara
/r/	perro, parra, corro, querría, carro, cerro

Tabla 4.37: Frases y palabras del conjunto de prueba: hiatos

Frases y palabras
piano, ríe, río, continúa, actué, influir, cuota, maíz, ateísmo, oído, baúl, rehúso.

En total se recolectaron 186 muestras por cada participante, todas en una sola sesión.

4.4. Diseño del reconocedor de emociones mediante la voz

Las investigaciones relacionadas al reconocimiento de emociones mediante la voz, se basan en la extracción de características del audio para obtener una representación matemática. Esta representación, es utilizada para entrenar los algoritmos de aprendizaje de máquina y de esta manera realizar clasificaciones. Gran parte de las investigaciones se basan en la exactitud o tasa de reconocimiento de los clasificadores, enfocándose en 3 aspectos: algoritmos de aprendizaje de máquina utilizados, características extraídas del audio y bases de datos empleadas.

Tabla 4.38: Tasa de reconocimiento para diferentes algoritmos de aprendizaje de máquina

Referencia	Algoritmo	Número de clases	% Exactitud	Base de Datos
[41]	GSVM	4	67.1 %	Susas [31]
[41]	HMM	4	70.1 %	Susas [31]
[41]	HMM	2	96.3 %	Susas [31]
[41]	GSVM	5	42.3 %	Aibo [67]
[27]	Rand-SVM	7	55.89 %	Emo-DB [11]
[27]	RF	7	81.05 %	Emo-DB [11]
[27]	GB	7	65.23 %	Emo-DB [11]
[38]	MCP NN	2	85 %	Propia sin nombre
[14]	RBF-SVM	5	93.75 %	Emo-DB [11]
[17]	RBF-SVM	6	84 %	Polish-DB [30]
[34]	MLP	7	83.1 %	Emo-DB [11]
[34]	RF	7	77.19 %	Emo-DB [11]
[34]	PNN	7	94.1 %	Emo-DB [11]
[34]	SVM	7	83.1 %	Emo-DB [11]
[13]	RBF-SVM	7	86.6 %	Emo-DB [11]
[56]	SVM	3	91.30 %	Emo-DB [11]
[56]	SVM	3	95.09 %	SJTU-DB [56]
[61]	DT	6	96.21 %	eNTERFACE'05 [46]
[17]	RBF-SVM	6	84 %	Polish-DB [30]

Investigaciones han explorado el uso de diferentes clasificadores como: SVM con núcleo de función base radial (RBF-SVM, por sus siglas en inglés), SVM con núcleo gaussiano (GSV, por sus siglas en inglés), modelo oculto de Márkov (HMM, por sus siglas en inglés), Bosques Aleatorios (RF, por sus siglas en inglés), aumento del gradiente (GB, por sus siglas en inglés), modelo de red neuronal de McCulloch y Pits (MCP-NN, por sus siglas en inglés), perceptrón multicapa (MLP, por sus siglas en inglés), red neuronal probabilística (PNN, por sus siglas en inglés), entre otros. El número de clases y la base de datos varía según la investigación. En la Tabla 4.38 se muestran la tasas de reconocimiento, bases de datos, algoritmos de aprendizaje de máquina de las investigaciones descritas anteriormente, incluyendo los resultados de otras investigaciones relevantes.

4.4.1. Selección del algoritmo

Muchos son los algoritmos propuestos para realizar este tipo de tarea de reconocimiento (ver Tabla, 4.38). El algoritmo de reconocimiento más destacado para realizar el reconocimiento de emociones es el SVM; este algoritmo puede ser utilizado usando diferente núcleos, y entre los más populares en este tipo de investigaciones se encuentran: lineal, polinomial y radial. No obstante, en aras de realizar una comparativa entre el desempeño de diversos algoritmos como en [27, 34, 50, 61], se realizarán pruebas utilizando tres algoritmos de clasificación: Máquinas de Vectores de Soporte (SVM), Bosques Aleatorios (RF) y Aumento del Gradiente (GB).

4.4.2. Implementación

Para realizar la implementación de los algoritmos SVM, RF y GB se utilizará la biblioteca Scikit-Learn [57]. En la Figura 4.8, se puede observar un diagrama donde se muestran los procesos de entrenamiento y prueba. El proceso de entrenamiento consiste en la extracción de características de las muestras de audio, para luego obtener un conjunto de vectores correctamente etiquetados, con estadísticas de cada una de las características seleccionadas, para posteriormente entrenar los algoritmos de aprendizaje de máquina. De manera similar, el proceso de pruebas extrae el vector

de características de una muestra no etiquetada y utiliza los modelos entrenados para clasificar la muestra.

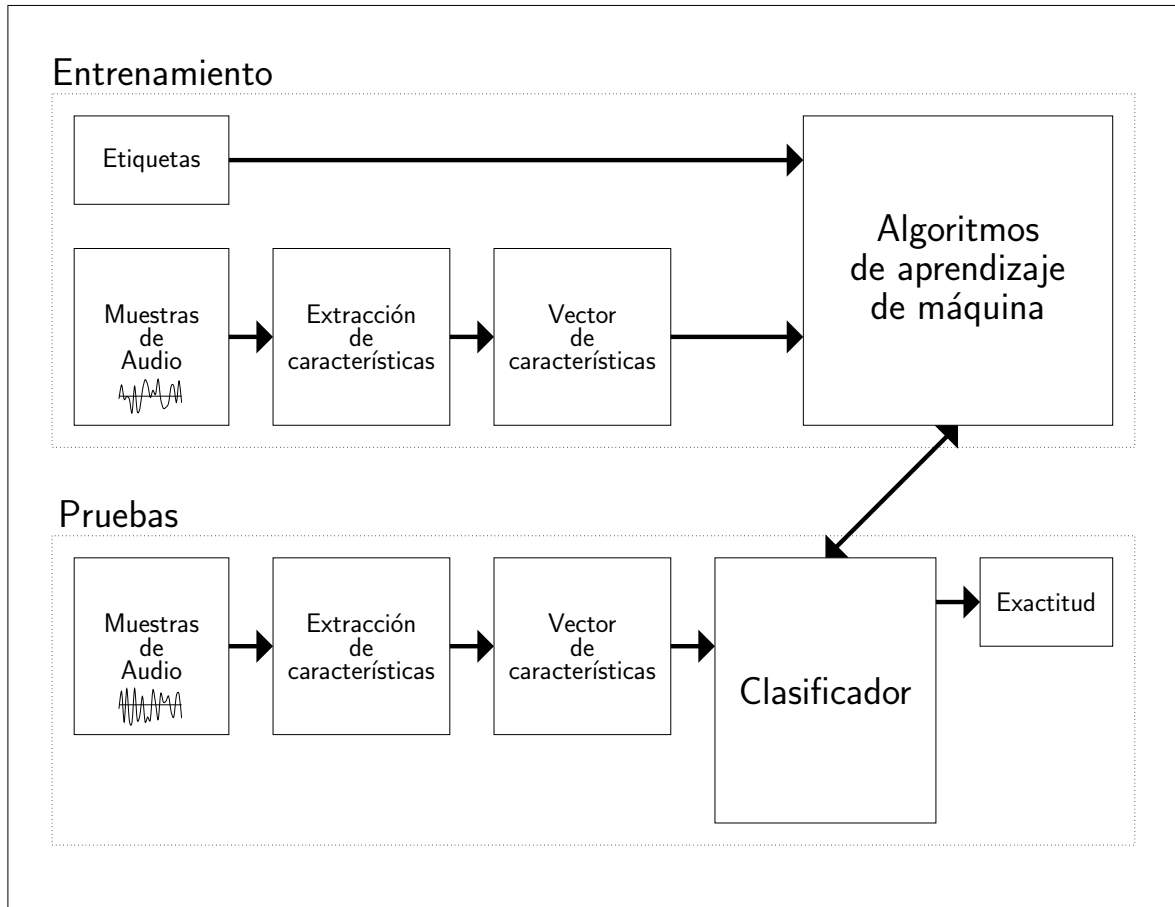


Figura 4.8: Diagrama de bloques: procesos de entrenamiento y pruebas

4.4.3. Optimización

Los algoritmos SVM, RF y GB, contienen diversos hiperpárametros. Los hiperpárametros son parámetros que no son directamente aprendidos dentro de un clasificador, por lo que pueden ser ajustados en aras de mejorar la tasa de reconocimiento de los clasificadores. A continuación se presentan los hiperpárametros más importantes de cada uno de estos clasificadores:

- **Máquinas de Vectores de Soporte:**

- **Núcleo:** lineal, polinomial o radial.
- **C:** El parámetro C compensa la correcta clasificación de las muestras de entrenamiento contra la maximización del margen de la función de decisión. Para valores mayores de C, se aceptará un margen menor si la función de decisión es mejor para clasificar correctamente todos los puntos de entrenamiento. Una C más baja fomentará un margen más grande, por lo tanto, una función de decisión más simple, a costa de la precisión del entrenamiento. En otras palabras, “C” se comporta como un parámetro de regularización en el SVM.
- **Grado:** Solo aplica en el núcleo polinomial. El núcleo polinomial de grado 1 genera una función de decisión lineal, mientras que núcleos polinomiales de mayor grado permiten una función de decisión más flexible.
- **Gamma:** Solo aplica en el núcleo de base radial. Este parámetro define hasta qué punto llega la influencia de una sola muestra de entrenamiento, los valores bajos que significan “lejos” y valores altos que significan “cerca”. El parámetro gamma puede verse como el inverso del radio de influencia de las muestras seleccionadas por el modelo como vectores de soporte.

- **Árboles Aleatorios:**

- **Número de estimadores:** El número de estimadores (`n_estimators`) representa el número de árboles en el bosque. Por lo general, cuanto mayor es el número de árboles, mejor para aprender los datos. Sin embargo,

usar muchos árboles puede ralentizar considerablemente el proceso de entrenamiento.

- **Profundidad Máxima:** La profundidad máxima (`max_depth`) representa la profundidad de cada árbol en el bosque. Cuanto más profundo es el árbol, más divisiones tiene y captura más información sobre los datos.
 - **Mínima división de muestras:** La mínima división de muestras (`min_samples_split`) representa el número mínimo de muestras necesarias para dividir un nodo interno. Esto puede variar entre considerar al menos una muestra en cada nodo a considerar todas las muestras en cada nodo. Cuando se aumenta este parámetro, cada árbol en el bosque se vuelve más restringido, ya que tiene que considerar más muestras en cada nodo.
 - **Muestras mínimas de hoja:** Las muestras mínimas de hoja (`min_samples_leaf`) es el número mínimo de muestras requeridas para estar en un nodo hoja.
 - **Características máximas:** Las características máximas (`max_features`), representa el número de características a considerar al buscar una mejor división.
- **Aumento del Gradiente:**
- **Mínima división de muestras:** La mínima división de muestras (`min_samples_split`), define el número mínimo de muestras que se requieren en un nodo para ser considerado para la división. Se utiliza para controlar el ajuste excesivo. Los valores altos impiden que un modelo aprenda relaciones que podrían ser altamente específicas para la muestra particular seleccionada para un árbol. Los valores demasiado altos pueden llevar a una adaptación insuficiente.
 - **Muestras mínimas de hoja:** Las muestras mínimas de hoja (`min_samples_leaf`), define las muestras mínimas (u observaciones) requeridas en un nodo terminal u hoja. Se utiliza para controlar el ajuste excesivo similar a `min_samples_split`. En general, se deben elegir valores

bajos para los problemas donde existe desbalance en las muestras de las clases, porque las regiones en las cuales la clase minoritaria será mayoritaria serán muy pequeñas.

- **Máxima profundidad:** La máxima profundidad (`max_depth`) representa la profundidad máxima de un árbol. Este parámetro es utilizado para controlar el ajuste excesivo ya que una mayor profundidad permitirá al modelo aprender relaciones muy específicas para una muestra en particular.
- **Características máximas:** Las características máximas (`max_depth`), representa el número de características a considerar al buscar una mejor división.
- **Tasa de aprendizaje:** La tasa de aprendizaje (`learning_rate`), determina el peso o impacto de cada árbol en el resultado final. El GB funciona comenzando con una estimación inicial que se actualiza utilizando la salida de cada árbol. El parámetro de aprendizaje controla la magnitud de este cambio en las estimaciones. Generalmente se prefieren los valores más bajos, ya que hacen que el modelo sea robusto a las características específicas del árbol y, por lo tanto, permite que se generalice bien.
- **Número de estimadores:** El número de estimadores (`n_estimators`), determina el número de árboles secuenciales a ser modelados. El GB es robusto incluso utilizando un gran número de árboles, no obstante, puede sobre entrenarse en algún punto.

Para optimizar los hiperparámetros de los modelos se realiza una búsqueda exhaustiva de candidatos, mediante la combinación de diferentes hiperparámetros y manteniendo los mejores resultados de cada combinación, utilizando pruebas de validación cruzada.

4.5. Diseño del reconocedor de personas mediante la voz

Las investigaciones relacionadas al reconocimiento de personas mediante la voz, pueden ser divididas en dependientes del texto e independientes del texto. A pesar de que ambas tareas de reconocimiento son similares, existe un mayor grado de complejidad en el reconocimiento independiente del texto, ya que realizar esta tarea involucra encontrar un conjunto representativo de frases y palabras que permitan reconocer a una persona. Estas investigaciones suelen basarse en la selección de un conjunto de oraciones o frases según algún criterio específico como: las palabras cotidianas o fonemas utilizados. Luego, estas frases son procesadas para entrenar los algoritmos de aprendizaje de máquina y posteriormente usarlos para identificar personas. Gran parte de estas investigaciones se basa en la exactitud o tasa de reconocimiento de los clasificadores, enfocándose en 3 aspectos: la tarea de reconocimiento (identificación o verificación), bases de datos, características extraídas del audio y algoritmos utilizados. En la Tabla 4.39, se muestran diferentes investigaciones que utilizan el algoritmo GMM para realizar identificación o verificación de personas.

Tabla 4.39: Tasa de reconocimiento en diferentes investigaciones utilizando el algoritmo GMM

Ref.	Tarea	Nº personas	% Exactitud	Base de Datos
[66]	Verificación	31 Hombres y 31 Mujeres	93.53 % Hombres 87.1 % Mujeres	Conversaciones naturales
[66]	Verificación	31 Hombres y 31 Mujeres	83.87 % Hombres 87.10 % Mujeres	Entrevistas
[43]	Identificación	15 personas	86.87 %	43 palabras del diccionario para entrenamiento y pruebas
[43]	Identificación	15 personas	84.18 %	43 palabras del diccionario para entrenamiento y palabras diferentes para las pruebas
[2]	Identificación	20 Hombres y 16 Mujeres	88 %	Base de datos CHAINS [16]
[2]	Identificación	20 Hombres o 16 Mujeres	91 %	Base de datos CHAINS [16]

4.5.1. Selección del algoritmo

En las investigaciones de reconocimiento de personas mediante la voz, son relevantes diversos algoritmos y técnicas de aprendizaje como: Modelo de Mezclas Gaussianas (GMM) [2, 43, 47, 66], Redes Neuronales Convolucionales (CNN) [44, 51], Cuantificación vectorial (VQ) [40], entre otros. Sin embargo, muchas investigaciones y aplicaciones actuales, utilizan el algoritmo GMM para realizar la identificación de personas y la técnica GMM-UBM para realizar la verificación personas. Esto se debe a diversos factores, como por ejemplo su alta precisión en comparación en otras técnicas como las CNN [44] o la poca diferencia en la precisión con otras técnicas en el estado del arte basadas en características “I-vector” [3, 20] .

Este algoritmo GMM permaneció en el estado del arte en las investigaciones de reconocimiento de personas durante mucho tiempo. Debido a esto, muchas técnicas se han desarrollado para mejorar su precisión y robustez. En este trabajo de investigación se implementará un identificador de personas basado en el algoritmo GMM, adicionalmente se implementará un verificador de personas basado en la técnica GMM-UBM.

4.5.2. Implementación

Para realizar la implementación del reconocedor de personas se utilizará la biblioteca Scikit-Learn [57]. El proceso de identificación se divide en dos: inscripción e identificación. La inscripción consiste en la extracción de características de las muestras de audio de cada una de las frases seleccionadas, y por cada usuario se genera un modelo estadístico utilizando los vectores de características de cada muestra de voz. La identificación consiste en la extracción de características de una nueva muestra, que se utilizará para generar un vector de características. Este vector, es probado en cada uno de los modelos estadísticos de los usuarios inscritos previamente. Posteriormente, se selecciona el modelo estadístico cuya probabilidad es mayor para la muestra dada.

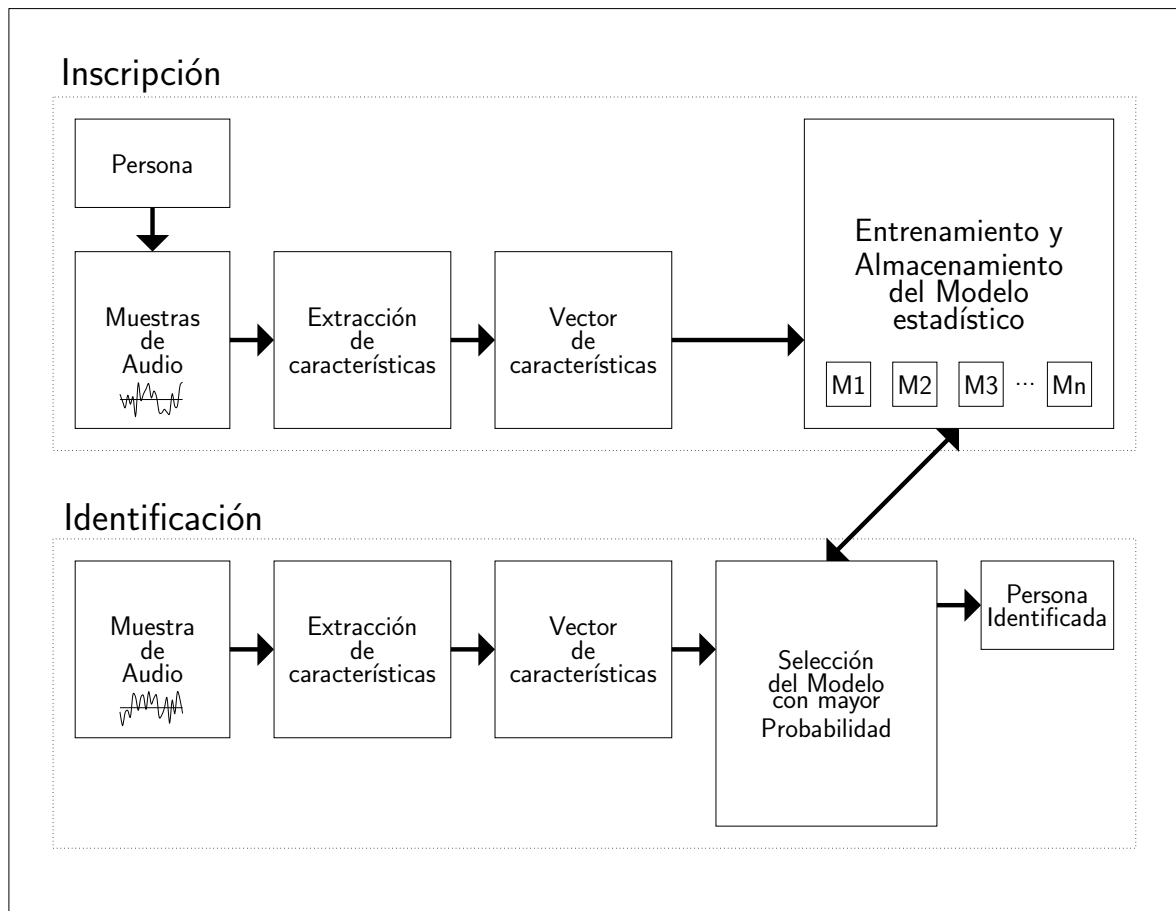


Figura 4.9: Diagrama de bloques: proceso de inscripción e identificación

Adicionalmente, un proceso de verificación es llevado a cabo utilizando la técnica GMM-UBM. El Modelo Universal (UBM, por sus siglas en inglés), es un modelo estadístico utilizado para representar las características generales de una población independiente de las personas inscritas [63]. El Modelo Universal es comparado con el modelo estadístico específico de una persona para realizar una verificación (aceptación o rechazo). Por ejemplo, en un sistema de verificación de personas mediante la voz, el UBM es un GMM entrenado con muestras de audio de un conjunto enorme de personas para representar las características generales de la voz de una población independiente. Utilizando un modelo estadístico específico de una persona inscrita y el UBM, es posible realizar una prueba basada en un radio de probabilidad para determinar si una muestra de audio pertenece al modelo estadístico dado (aceptación) o no (rechazo).

Dada una observación, O , y una persona, P . La tarea de verificación se puede

describir como una prueba de hipótesis:

$$\begin{cases} H_0 : \text{La observación } O \text{ es de la persona } P \\ H_1 : \text{La observación } O \text{ no es de la persona } P \end{cases} \quad (4.5.2.1)$$

Utilizando reconocimiento de patrones estadísticos, la prueba óptima para decidir entre las dos hipótesis, es un prueba basada en un radio de probabilidad. Esta prueba viene dada por:

$$\frac{P(O|H_0)}{P(O|H_1)} \begin{cases} \geq \theta \text{ Aceptar } H_0 \\ < \theta \text{ Rechazar } H_0 \end{cases} \quad (4.5.2.2)$$

Donde $P(O|H_i), i = 0, 1$ es la probabilidad de la función de densidad para la hipótesis H_i . El límite de decisión para aceptar o rechazar H_0 es θ . El objetivo principal es determinar los modelos estadísticos para representar $P(O|H_0)$ y $P(O|H_1)$.

Una vez encontrados los modelos estadísticos de las hipótesis $P(O|H_0)$ y $P(O|H_1)$ utilizando el algoritmo GMM, se determina si el modelo estadístico pertenece al usuario utilizando el radio de probabilidad para la toma de decisiones [8]. Este proceso, consiste en comparar el radio de probabilidad con un límite de decisión. Si el radio de probabilidad es mayor al límite se acepta, en caso contrario, se rechaza. Sea λ_{hyp} el modelo que representa a H_0 y $\lambda_{\bar{hyp}}$ el modelo que representa a H_1 , el radio de probabilidad viene dado por:

$$\Lambda(X) = \log(P(X|\lambda_{hyp})) - \log(P(X|\lambda_{\bar{hyp}})) \quad (4.5.2.3)$$

$$\Lambda(X) \begin{cases} \geq \theta \text{ Aceptar } H_0 \\ < \theta \text{ Rechazar } H_0 \end{cases} \quad (4.5.2.4)$$

Determinar de forma óptima límite de decisión θ , es un problema abierto [8]. No obstante, por lo general se utiliza un valor fijo determinado empíricamente. En la Figura 4.10, se puede observar el diagrama de bloques de un sistema de verificación que utiliza el radio de probabilidad para tomar la decisión, similar a [8].

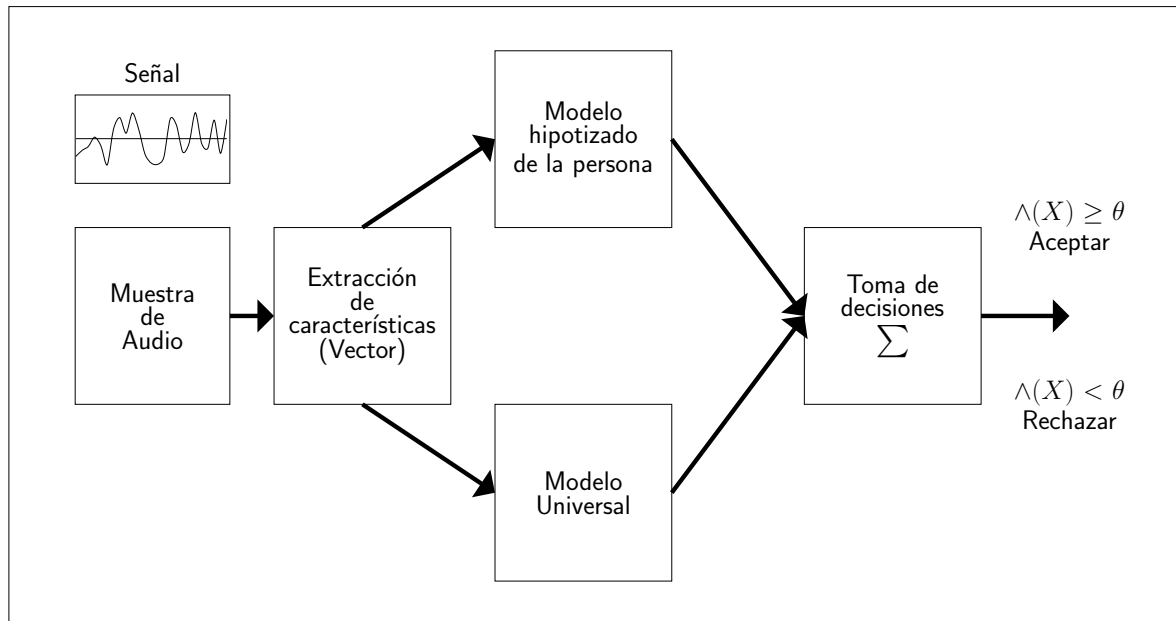


Figura 4.10: Diagrama de bloques: sistema de verificación basado en radio de probabilidad [8]

4.5.3. Optimización

El principal hiperparámetro del algoritmo GMM, es el número de componentes. Si se selecciona innecesariamente un gran número de componentes, el algoritmo genera probabilidades muy pequeñas. Por lo general, se realizan diferentes modelos con diferente número de componentes para seleccionar aquellos que permitan una mejor precisión en la identificación y verificación [2, 64].

Capítulo 5

Pruebas

En este capítulo se presentan las pruebas realizadas al Módulo Reconfigurable de Reconocimiento para la interacción humano-robot. Cada uno de los componentes principales del módulo: gestor local, gestor remoto y servicios de reconocimiento, se representan como nodos del grafo computacional de ROS [45], y adicionalmente, una vez que estos nodos se activan, se integran e interactúan en dicho grafo. Por esta razón se realizarán pruebas unitarias y de integración en cada uno de los componentes.

Los componentes principales del Módulo Reconfigurable de Reconocimiento, utilizan las tareas de reconocimiento descritas en el caso de estudio (reconocimiento de emociones y personas, mediante la voz). Estas tareas de reconocimiento son implementadas utilizando algoritmos de aprendizaje de máquina, es por esta razón que se realizarán diferentes tipos de pruebas para probar la exactitud de los modelos.

5.1. Pruebas sobre el Módulo Reconfigurable de Reconocimiento

El Módulo Reconfigurable de Reconocimiento, cuenta con diversos componentes como el gestor local y el gestor remoto, éstos, de manera individual e integrada, permiten ofrecer toda la funcionalidad relacionada con el módulo. Debido a esto, es necesario realizar diversos tipos de pruebas que permitan verificar el funcionamiento de los componentes del módulo.

5.1.1. Planificación de pruebas

El Módulo Reconfigurable de Reconocimiento, es implementando mediante el framework ROS [59]. En este framework son comunes dos tipos de pruebas: pruebas unitarias a nivel de código y pruebas de integración a nivel de nodos. A continuación se presenta de manera detallada toda la información relacionada con las pruebas realizadas al módulo.

5.1.1.1. Estrategias utilizadas en las pruebas

- **Pruebas unitarias:** Las pruebas unitarias son realizadas a nivel de código para capturar errores de programación específicos del lenguaje de programación como errores de lógica, funciones particulares, entre otros.

Estas pruebas serán aplicadas a cada uno de los gestores del Módulo Reconfigurable de Reconocimiento, esto se debe a que cada uno de estos gestores es representando como un nodo, cuyo objetivo es ofrecer una funcionalidad específica.

- **Pruebas de integración:** Las pruebas de integración son realizadas a nivel de funcionamiento del módulo reconfigurable para capturar errores de interacción entre los distintos componentes que lo constituyen.

Estas pruebas serán aplicadas sobre las distintas interacciones que realizan los nodos que se corresponden con el gestor local y el gestor remoto. Adicionalmente,

se realizarán pruebas de integración sobre las interacciones del robot y los servicios de reconocimiento que ofrece el módulo reconfigurable.

- **Pruebas del sistema:** Las pruebas del sistema son realizadas sobre el módulo reconfigurable con todos sus componentes integrados, y están basadas en pruebas funcionales donde solo se toma en consideración las entradas y salidas.

Estas pruebas serán realizadas sobre el módulo reconfigurable con todos sus componentes integrados. El objetivo principal es probar el funcionamiento a nivel de requisitos de aplicación.

5.1.1.2. Límites y suposiciones

- Las pruebas realizadas sobre el módulo reconfigurable se basan en pruebas funcionales de integración, cuyo objetivo es verificar el correcto ensamblaje entre los distintos componentes una vez que han sido probados de manera unitaria.
- Para realizar las pruebas unitarias sobre los nodos, estos deben encontrarse activos. Si la activación de un nodo depende de otro nodo ambos deben ser activados para realizar las correspondientes pruebas.
- Para realizar las pruebas de integración todos los nodos del módulo deben estar activos, el gestor local en el procesamiento interno y el gestor remoto en el procesamiento externo. Adicionalmente, debe estar instalado por lo menos un paquete de reconocimiento.
- No se contemplan pruebas no funcionales.
- No se contemplan pruebas sobre los paquetes de reconocimiento.

5.1.2. Especificación del diseño de las pruebas

El Módulo Reconfigurable de Reconocimiento está compuesto de 2 componentes principales: el gestor remoto y el gestor local, cuyo objetivo es gestionar las tareas de reconocimiento que se ofrecen como servicio al robot. A continuación se presenta de manera detallada el enfoque utilizado para probar cada una de las características de estos componentes.

5.1.2.1. Pruebas del gestor local

El gestor local es representado como un nodo del grafo computacional de ROS. Este nodo, ofrece diversos tipos de servicios que se encargan de gestionar las tareas de reconocimiento del robot.

Las pruebas unitarias realizadas sobre este nodo son de caja negra y caja blanca. Las pruebas de caja negra son realizadas a las interfaces de los servicios del nodo, mientras que las pruebas de caja blanca son realizadas a nivel de código en los procedimientos involucrados en el nodo.

Las pruebas de integración son de caja negra y son realizadas sobre las interacciones del gestor local y el robot.

5.1.2.2. Pruebas del gestor remoto

El gestor remoto también es representado como un del grafo computacional de ROS. Este nodo, ofrece diversos tipos de servicios que se encargan de gestionar las tareas de reconocimiento en el procesamiento externo del robot. Adicionalmente, ofrece servicios relacionados con el gestor remoto.

Las pruebas unitarias realizadas sobre este nodo son de caja negra y caja blanca. Las pruebas de caja negra son realizadas a las interfaces de los servicios del nodo, mientras que las pruebas de caja blanca son realizadas a nivel de código en los procedimientos involucrados en el nodo.

Las pruebas de integración son de caja negra y son realizadas sobre las interacciones del gestor local y el gestor remoto.

5.1.2.3. Pruebas sobre los servicios de reconocimiento

Ambos gestores activan o levantan procesos que permiten al robot solicitar servicios de reconocimiento. Estos servicios son implementados mediante un nodo genérico que utiliza los paquetes de reconocimiento para ofrecer los servicios mediante mensajes del grafo computacional de ROS.

Las pruebas unitarias realizadas sobre este nodo son de caja negra y caja blanca. Las pruebas de caja negra son realizadas a las interfaces de los servicios y mensajes del nodo, mientras que las pruebas de caja blanca son realizadas a nivel de código en los procedimientos involucrados en el nodo.

Las pruebas de integración son de caja negra y son realizadas sobre las interacciones del robot y los nodos genéricos.

5.1.2.4. Medición de memoria y procesamiento

Adicionalmente, se realizará la medición de memoria y procesamiento sobre uno de los gestores (local o remoto) y una de las tareas de reconocimiento. Para observar el consumo de memoria y procesamiento durante un conjunto de actividades.

5.1.2.5. Identificación de casos de prueba

Tabla 5.1: Casos de pruebas unitarias del gestor local

Id prueba	Nombre
P-GL-01	Activación de tarea de reconocimiento local.
P-GL-02	Desactivación de tareas de reconocimiento local.
P-GL-03	Activación de lista de tareas de reconocimiento local.
P-GL-04	Desactivación de lista de tareas de reconocimiento local.
P-GL-05	Solicitud de tareas de reconocimiento local.
P-GL-06	Desinstalación de paquetes de reconocimiento local.
P-GL-07	Desinstalación de lista de paquetes de reconocimiento local.

Tabla 5.2: Casos de pruebas de integración del gestor local

Id prueba	Nombre
P-GL-08	Instalación de paquetes de reconocimiento local.
P-GL-09	Instalación de lista de paquetes de reconocimiento local.
P-GL-10	Solicitud de direcciones de paquetes remotos.

Tabla 5.3: Casos de pruebas unitarias del gestor remoto

Id prueba	Nombre
P-GR-01	Activación de tareas de reconocimiento remoto.
P-GR-02	Desactivación de tareas de reconocimiento remoto.
P-GR-03	Activación de lista de tareas de reconocimiento remoto.
P-GR-04	Desactivación de lista de tareas de reconocimiento remoto.
P-GR-05	Instalación de paquetes de reconocimiento remoto.
P-GR-06	Desinstalación de paquete de reconocimiento remoto.
P-GR-07	Solicitud de tareas de reconocimiento remoto.

Tabla 5.4: Casos de pruebas de integración del gestor remoto

Id prueba	Nombre
P-GR-08	Envío de paquetes de reconocimiento al gestor local.

Tabla 5.5: Casos de pruebas de integración de los servicios de reconocimiento

Id prueba	Nombre
P-SR-01	Solicitar servicio de reconocimiento remoto.
P-SR-02	Solicitar servicio de reconocimiento local.

Tabla 5.6: Casos de pruebas sobre el sistema

Id prueba	Nombre
P-SIS-01	Activación, desactivación y solicitud de servicios de reconocimiento en el gestor local.
P-SIS-02	Activación, desactivación y solicitud de servicios de reconocimiento en el gestor remoto.
P-SIS-03	Instalación, activación, desactivación y solicitud de servicios de reconocimiento en el gestor local y remoto.

5.1.3. Especificación de los casos de prueba

Tabla 5.7: Caso de prueba: P-GL-01

Id de la prueba:	P-GL-01
Nombre del módulo:	Gestor Local
Nombre prueba:	Activación de tarea de reconocimiento
Descripción:	Pruebas de activación de tarea de reconocimiento local.
Dependencias:	Ninguna.
Pre-condiciones:	1. Gestor local activo. 2. Paquete: paquete01 instalado en el gestor local.
Post-condiciones:	1. Estado de la tarea de reconocimiento: paquete01, activado.
Pasos de la prueba:	1. Solicitar la activación de la tarea de reconocimiento: paquete01 al gestor local.
Datos de la prueba:	Identificador de la tarea: paquete01
Resultado:	True
Estado:	Aprobado

Tabla 5.8: Caso de prueba: P-GL-02

Id de la prueba:	P-GL-02
Nombre del módulo:	Gestor Local
Nombre prueba:	Desactivación de tarea de reconocimiento
Descripción:	Prueba de desactivación de tarea de reconocimiento local.
Dependencias:	Ninguna.
Pre-condiciones:	1. Gestor local activo. 2. Paquete: paquete01 instalado en el gestor local. 3. Estado de la tarea de reconocimiento: paquete01, activado.
Post-condiciones:	1. Estado de la tarea de reconocimiento: paquete01, desactivado.
Pasos de la prueba:	1. Solicitar la desactivación de la tarea de reconocimiento: paquete01 al gestor local.
Datos de la prueba:	Identificador de la tarea: paquete01
Resultado:	True
Estado:	Aprobado

Tabla 5.9: Caso de prueba: P-GL-03

Id de la prueba:	P-GL-03
Nombre del módulo:	Gestor Local
Nombre prueba:	Activación de lista de tareas de reconocimiento local
Descripción:	Pruebas de activación de lista de tareas de reconocimiento local.
Dependencias:	P-GL-01
Pre-condiciones:	<ol style="list-style-type: none"> 1. Gestor local activo. 2. Paquetes: paquete01, paquete02 y paquete03, instalados en el gestor local. 3. Estado de las tareas de reconocimiento: paquete01, paquete02 y paquete03, desactivado.
Post-condiciones:	<ol style="list-style-type: none"> 1. Estado de la tarea de reconocimiento: paquete01, paquete02 y paquete03, activado.
Pasos de la prueba:	<ol style="list-style-type: none"> 1. Solicitar la activación de la lista de tareas de reconocimiento: paquete01, paquete02 y paquete03 al gestor local.
Datos de la prueba:	Identificador de las tareas: paquete01, paquete02 y paquete03
Resultado:	paquete01, paquete02 y paquete03
Estado:	Aprobado

Tabla 5.10: Caso de prueba: P-GL-04

Id de la prueba:	P-GL-04
Nombre del módulo:	Gestor Local
Nombre prueba:	Desactivación de lista de tareas de reconocimiento local
Descripción:	Pruebas de desactivación de lista de tareas de reconocimiento local.
Dependencias:	P-GL-02
Pre-condiciones:	<ol style="list-style-type: none"> 1. Gestor local activo. 2. Paquetes: paquete01, paquete02 y paquete03, instalados en el gestor local. 3. Estado de las tareas de reconocimiento: paquete01, paquete02 y paquete03, activado.
Post-condiciones:	<ol style="list-style-type: none"> 1. Estado de la tarea de reconocimiento: paquete01, paquete02 y paquete03, desactivado.
Pasos de la prueba:	<ol style="list-style-type: none"> 1. Solicitar la desactivación de la lista de tareas de reconocimiento: paquete01, paquete02 y paquete03 al gestor local.
Datos de la prueba:	Identificador de las tareas: paquete01, paquete02 y paquete03
Resultado:	paquete01, paquete02 y paquete03
Estado:	Aprobado

Tabla 5.11: Caso de prueba: P-GL-05

Id de la prueba:	P-GL-05
Nombre del módulo:	Gestor Local
Nombre prueba:	Solicitud de tareas de reconocimiento local
Descripción:	Prueba de solicitud de tareas de reconocimiento local.
Dependencias:	Ninguna
Pre-condiciones:	<ol style="list-style-type: none"> 1. Gestor local activo. 2. Paquetes: paquete01, paquete02 y paquete03, instalados en el gestor local.
Post-condiciones:	<ol style="list-style-type: none"> 1. Gestor local activo. 2. Paquetes: paquete01, paquete02 y paquete03, instalados en el gestor local.
Pasos de la prueba:	<ol style="list-style-type: none"> 1. Solicitar la lista de tareas de reconocimiento local.
Datos de la prueba:	Ninguno
Resultado:	paquete01, paquete02 y paquete03
Estado:	Aprobado

Tabla 5.12: Caso de prueba: P-GL-06

Id de la prueba:	P-GL-06
Nombre del módulo:	Gestor Local
Nombre prueba:	Desinstalación de paquete de reconocimiento local.
Descripción:	Prueba de Desinstalación de paquete de reconocimiento local.
Dependencias:	P-GL-08, P-GL-09
Pre-condiciones:	<ol style="list-style-type: none"> 1. Gestor local activo. 2. Paquete: paquete01, instalado. 3. Estado de la tarea de reconocimiento: paquete01, desactivado.
Post-condiciones:	<ol style="list-style-type: none"> 1. Paquete: paquete01, desinstalado.
Pasos de la prueba:	<ol style="list-style-type: none"> 1. Solicitar la desinstalación del paquete: paquete01 en el gestor local.
Datos de la prueba:	Identificador de la tarea: paquete01.
Resultado:	True
Estado:	Aprobado

Tabla 5.13: Caso de prueba: P-GL-07

Id de la prueba:	P-GL-07
Nombre del módulo:	Gestor Local
Nombre prueba:	Desinstalación de lista de paquetes de reconocimiento local.
Descripción:	Prueba de desinstalación de lista de paquetes de reconocimiento local.
Dependencias:	P-GL-06
Pre-condiciones:	<ol style="list-style-type: none"> 1. Gestor local activo. 2. Paquetes: paquete01, paquete02 y paquete03, instalados en el gestor local.
Post-condiciones:	<ol style="list-style-type: none"> 1. Gestor local activo. 2. Paquetes: paquete01, paquete02 y paquete03, desinstalados del gestor local.
Pasos de la prueba:	<ol style="list-style-type: none"> 1. Solicitar la desinstalación de la lista de tareas de reconocimiento: paquete01, paquete02 y paquete03 al gestor local.
Datos de la prueba:	Identificador de las tareas: paquete01, paquete02 y paquete03
Resultado:	paquete01, paquete02 y paquete03
Estado:	Aprobado

Tabla 5.14: Caso de prueba: P-GL-08

Id de la prueba:	P-GL-08
Nombre del módulo:	Gestor Local
Nombre prueba:	Instalación de paquetes de reconocimiento local.
Descripción:	Prueba de instalación de paquete de reconocimiento local.
Dependencias:	P-GR-05
Pre-condiciones:	<ol style="list-style-type: none"> 1. Gestor local activo. 2. Gestor remoto activo. 3. Paquete: paquete01, instalado en el gestor remoto.
Post-condiciones:	<ol style="list-style-type: none"> 1. Paquete: paquete01, instalado en el gestor local.
Pasos de la prueba:	<ol style="list-style-type: none"> 1. Solicitar la instalación del paquete de reconocimiento: paquete01 en el gestor local.
Datos de la prueba:	Identificador de la tarea: paquete01.
Resultado:	True
Estado:	Aprobado

Tabla 5.15: Caso de prueba: P-GL-09

Id de la prueba:	P-GL-09
Nombre del módulo:	Gestor Local
Nombre prueba:	Instalación de lista de paquetes de reconocimiento local.
Descripción:	Prueba de instalación de lista de paquetes de reconocimiento local.
Dependencias:	P-GL-08
Pre-condiciones:	<ol style="list-style-type: none"> 1. Gestor local activo. 2. Gestor remoto activo. 3. Paquetes: paquete01, paquete02 y paquete03, instalados en el gestor remoto.
Post-condiciones:	1. Paquetes: paquete01,paquete02 y paquete03 instalados en el gestor local.
Pasos de la prueba:	1. Solicitar la instalación de la lista de paquetes de reconocimiento: paquete01, paquete02 y paquete03 en el gestor local.
Datos de la prueba:	Identificador de la tareas: paquete01, paquete02 y paquete03.
Resultado:	paquete01, paquete02 y paquete03
Estado:	Aprobado

Tabla 5.16: Caso de prueba: P-GL-10

Id de la prueba:	P-GL-10
Nombre del módulo:	Gestor Local
Nombre prueba:	Solicitud de direcciones de paquetes remotos.
Descripción:	Prueba de solicitud de direcciones de paquetes remotos.
Dependencias:	P-GR-05
Pre-condiciones:	<ol style="list-style-type: none"> 1. Gestor remoto activo. 2. Paquete: paquete01, instalado en el gestor remoto.
Post-condiciones:	1. Dirección del paquete: paquete01 obtenida en el gestor local.
Pasos de la prueba:	1. Solicitar la dirección de un paquete de reconocimiento remoto.
Datos de la prueba:	Identificador de la tarea: paquete01.
Resultado:	Dirección del paquete01
Estado:	Aprobado

Tabla 5.17: Caso de prueba: P-GR-01

Id de la prueba:	P-GR-01
Nombre del módulo:	Gestor Remoto
Nombre prueba:	Activación de tarea de reconocimiento remoto
Descripción:	Pruebas de activación de tarea de reconocimiento remoto.
Dependencias:	Ninguna.
Pre-condiciones:	1. Gestor remoto activo. 2. Paquete: paquete01 instalado en el gestor remoto.
Post-condiciones:	1. Estado de la tarea de reconocimiento: paquete01, activado.
Pasos de la prueba:	1. Solicitar la activación de la tarea de reconocimiento: paquete01 al gestor remoto.
Datos de la prueba:	Identificador de la tarea: paquete01
Resultado:	True
Estado:	Aprobado

Tabla 5.18: Caso de prueba: P-GR-02

Id de la prueba:	P-GR-02
Nombre del módulo:	Gestor Remoto
Nombre prueba:	Desactivación de tarea de reconocimiento remoto
Descripción:	Prueba de desactivación de tarea de reconocimiento remoto.
Dependencias:	Ninguna.
Pre-condiciones:	1. Gestor remoto activo. 2. Paquete: paquete01 instalado en el gestor remoto. 3. Estado de la tarea de reconocimiento: paquete01, activado.
Post-condiciones:	1. Estado de la tarea de reconocimiento: paquete01, desactivado.
Pasos de la prueba:	1. Solicitar la desactivación de la tarea de reconocimiento: paquete01 al gestor remoto.
Datos de la prueba:	Identificador de la tarea: paquete01
Resultado:	True
Estado:	Aprobado

Tabla 5.19: Caso de prueba: P-GR-03

Id de la prueba:	P-GR-03
Nombre del módulo:	Gestor Remoto
Nombre prueba:	Activación de lista de tareas de reconocimiento remoto
Descripción:	Pruebas de activación de lista de tareas de reconocimiento remoto.
Dependencias:	P-GR-01
Pre-condiciones:	<ol style="list-style-type: none"> 1. Gestor remoto activo. 2. Paquetes: paquete01, paquete02 y paquete03, instalados en el gestor remoto. 3. Estado de las tareas de reconocimiento: paquete01, paquete02 y paquete03, desactivado.
Post-condiciones:	<ol style="list-style-type: none"> 1. Estado de la tarea de reconocimiento: paquete01, paquete02 y paquete03, activado.
Pasos de la prueba:	<ol style="list-style-type: none"> 1. Solicitar la activación de la lista de tareas de reconocimiento: paquete01, paquete02 y paquete03 al gestor remoto.
Datos de la prueba:	Identificador de las tareas: paquete01, paquete02 y paquete03
Resultado:	paquete01, paquete02 y paquete03
Estado:	Aprobado

Tabla 5.20: Caso de prueba: P-GR-04

Id de la prueba:	P-GR-04
Nombre del módulo:	Gestor Remoto
Nombre prueba:	Desactivación de lista de tareas de reconocimiento remoto
Descripción:	Pruebas de desactivación de lista de tareas de reconocimiento remoto
Dependencias:	P-GR-02
Pre-condiciones:	<ol style="list-style-type: none"> 1. Gestor remoto activo. 2. Paquetes: paquete01, paquete02 y paquete03, instalados en el gestor remoto. 3. Estado de las tareas de reconocimiento: paquete01, paquete02 y paquete03, activado.
Post-condiciones:	<ol style="list-style-type: none"> 1. Estado de las tareas de reconocimiento: paquete01, paquete02 y paquete03, desactivado.
Pasos de la prueba:	<ol style="list-style-type: none"> 1. Solicitar la desactivación de la lista de tareas de reconocimiento: paquete01, paquete02 y paquete03 al gestor remoto.
Datos de la prueba:	Identificador de las tareas: paquete01, paquete02 y paquete03
Resultado:	paquete01, paquete02 y paquete03
Estado:	Aprobado

Tabla 5.21: Caso de prueba: P-GR-05

Id de la prueba:	P-GR-05
Nombre del módulo:	Gestor Remoto
Nombre prueba:	Instalación de paquete de reconocimiento remoto.
Descripción:	Prueba de instalación de paquete de reconocimiento remoto.
Dependencias:	Ninguno
Pre-condiciones:	1. Gestor remoto activo.
Post-condiciones:	1. Paquete: paquete01, instalado en el gestor remoto.
Pasos de la prueba:	1. Solicitar la instalación del paquete de reconocimiento: paquete01 en el gestor remoto.
Datos de la prueba:	Dirección del paquete de reconocimiento: paquete01
Resultado:	True
Estado:	Aprobado

Tabla 5.22: Caso de prueba: P-GR-06

Id de la prueba:	P-GR-06
Nombre del módulo:	Gestor Remoto
Nombre prueba:	Desinstalación de paquete de reconocimiento remoto.
Descripción:	Prueba de Desinstalación de paquete de reconocimiento remoto.
Dependencias:	P-GR-05
Pre-condiciones:	1. Gestor remoto activo. 2. Paquete: paquete01, instalado. 3. Estado de la tarea de reconocimiento: paquete01, desactivado.
Post-condiciones:	1. Paquete: paquete01, desinstalado.
Pasos de la prueba:	1. Solicitar la desinstalación del paquete: paquete01 en el gestor remoto.
Datos de la prueba:	Identificador de la tarea: paquete01.
Resultado:	True
Estado:	Aprobado

Tabla 5.23: Caso de prueba: P-GR-07

Id de la prueba:	P-GR-07
Nombre del módulo:	Gestor Remoto
Nombre prueba:	Solicitud de tareas de reconocimiento remoto
Descripción:	Prueba de solicitud de tareas de reconocimiento remoto.
Dependencias:	Ninguna
Pre-condiciones:	<ol style="list-style-type: none"> 1. Gestor remoto activo. 2. Paquetes: paquete01, paquete02 y paquete03, instalados en el gestor remoto.
Post-condiciones:	<ol style="list-style-type: none"> 1. Gestor local activo. 2. Paquetes: paquete01, paquete02 y paquete03, instalados en el gestor remoto.
Pasos de la prueba:	<ol style="list-style-type: none"> 1. Solicitar la lista de tareas de reconocimiento remoto.
Datos de la prueba:	Ninguno
Resultado:	paquete01, paquete02 y paquete03
Estado:	Aprobado

Tabla 5.24: Caso de prueba: P-SR-01

Id de la prueba:	P-SR-01
Nombre del módulo:	Servicios de reconocimiento
Nombre prueba:	Solicitar servicio de reconocimiento remoto
Descripción:	Prueba de solicitud de servicios de reconocimiento remoto
Dependencias:	P-GR-01,P-GR-02,P-GR-05,P-GL-07
Pre-condiciones:	<ol style="list-style-type: none"> 1. Gestor remoto activo. 2. Paquete: paquete01, instalado en el gestor remoto. 3. Estado de la tarea de reconocimiento: paquete01, activado.
Post-condiciones:	<ol style="list-style-type: none"> 1. Gestor remoto activo. 2. Estado de la tarea de reconocimiento: paquete01, activado. 3. Solicitud de reconocimiento enviada.
Pasos de la prueba:	<ol style="list-style-type: none"> 1. Enviar solicitud de reconocimiento utilizando la dirección del archivo o una representación binaria del mismo. 2. Esperar respuesta.
Datos de la prueba:	Dirección del archivo a reconocer y nombre del tópico correspondiente a la tarea de reconocimiento.
Resultado:	Respuesta del reconocimiento.
Estado:	Aprobado

Tabla 5.25: Caso de prueba: P-SR-02

Id de la prueba:	P-SR-02
Nombre del módulo:	Servicios de reconocimiento
Nombre prueba:	Solicitar servicio de reconocimiento local
Descripción:	Prueba de solicitud de servicios de reconocimiento local
Dependencias:	P-GL-01,P-GL-02,P-GL-05,P-GL-08
Pre-condiciones:	<ol style="list-style-type: none"> 1. Gestor local activo. 2. Paquete: paquete01, instalado en el gestor local. 3. Estado de la tarea de reconocimiento: paquete01, activado.
Post-condiciones:	<ol style="list-style-type: none"> 1. Gestor local activo. 2. Estado de la tarea de reconocimiento: paquete01, activado. 3. Solicitud de reconocimiento enviada.
Pasos de la prueba:	<ol style="list-style-type: none"> 1. Enviar solicitud de reconocimiento utilizando la dirección del archivo o una representación binaria del mismo. 2. Esperar respuesta.
Datos de la prueba:	Dirección del archivo a reconocer y nombre del tópico correspondiente a la tarea de reconocimiento.
Resultado:	Respuesta del reconocimiento.
Estado:	Aprobado

Tabla 5.26: Caso de prueba: P-SIS-01

Id de la prueba:	P-SIS-01
Nombre del módulo:	Módulo Reconfigurable de Reconocimiento
Nombre prueba:	Activación, desactivación y solicitud de servicios de reconocimiento en el gestor local.
Descripción:	Prueba sobre la gestión de tareas y solicitud de servicios de reconocimiento sobre el gestor local
Dependencias:	Desde P-GL-01 hasta P-GL-10
Pre-condiciones:	<ol style="list-style-type: none"> 1. Gestor local activo. 2. Paquetes: <code>SpeechEmotionRecognition</code> y <code>SpeechSpeakerIdentification</code> instalados en el gestor local. 3. Estado de las tareas de reconocimiento: <code>SpeechEmotionRecognition</code> y <code>SpeechSpeakerIdentification</code>, desactivado.
Post-condiciones:	<ol style="list-style-type: none"> 1. Gestor local activo. 2. Paquetes: <code>SpeechEmotionRecognition</code> y <code>SpeechSpeakerIdentification</code> instalados en el gestor local. 3. Estado de las tareas de reconocimiento: <code>SpeechEmotionRecognition</code> y <code>SpeechSpeakerIdentification</code>, desactivado.
Pasos de la prueba:	<ol style="list-style-type: none"> 1. Solicitar la activación de la tarea <code>SpeechEmotionRecognition</code>. 2. Solicitar la activación de la tarea <code>SpeechSpeakerIdentification</code>. 3. Realizar solicitudes de servicios de reconocimiento a la tarea <code>SpeechEmotionRecognition</code>. 4. Realizar solicitudes de servicios de reconocimiento a la tarea <code>SpeechSpeakerIdentification</code>. 5. Solicitar la desactivación de la tarea <code>SpeechEmotionRecognition</code>. 6. Solicitar la desactivación de la tarea <code>SpeechSpeakerIdentification</code>.
Datos de la prueba:	Direcciones de los archivos a reconocer y nombre de los tópicos correspondientes a las tareas <code>SpeechEmotionRecognition</code> y <code>SpeechSpeakerIdentification</code> del gestor local.
Resultado:	Respuesta de las solicitudes de reconocimiento.
Estado:	Aprobado

A continuación se presentan diversos diagramas de nodos y tópicos utilizando el paquete de ROS `rqt_graph`. Cada uno de los diagramas se corresponde con uno o más pasos de la prueba P-SIS-01, especificada en la Tabla 5.25.



Figura 5.1: Diagrama de nodos y tópicos de la prueba P-SIS-01: Nodo inicial, gestor local.

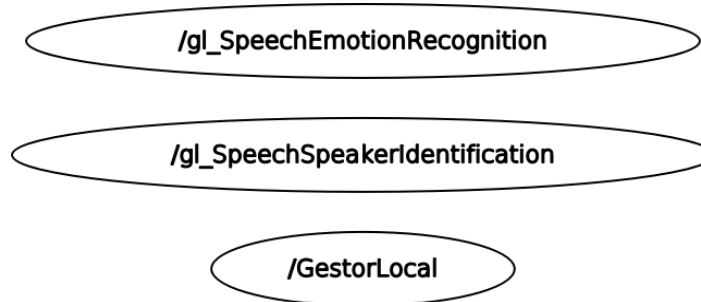


Figura 5.2: Diagrama de nodos y tópicos de la prueba P-SIS-01: Activación de las tareas SpeechEmotionRecognition y SpeechSpeakerIdentification del gestor local.

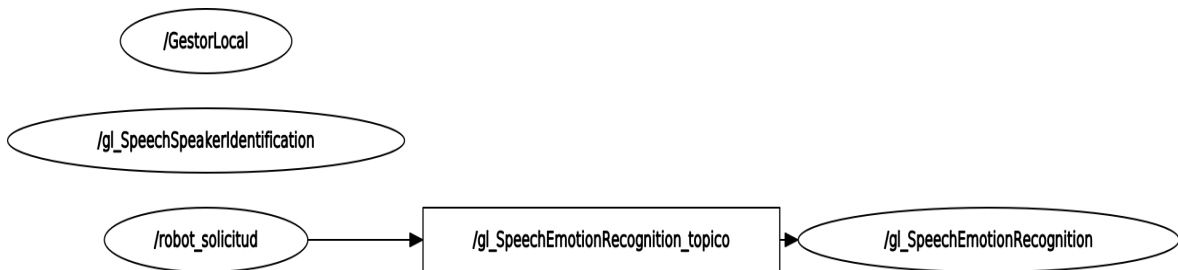


Figura 5.3: Diagrama de nodos y tópicos de la prueba P-SIS-01: Solicitudes de servicios reconocimiento a la tarea de reconocimiento SpeechEmotionRecognition.

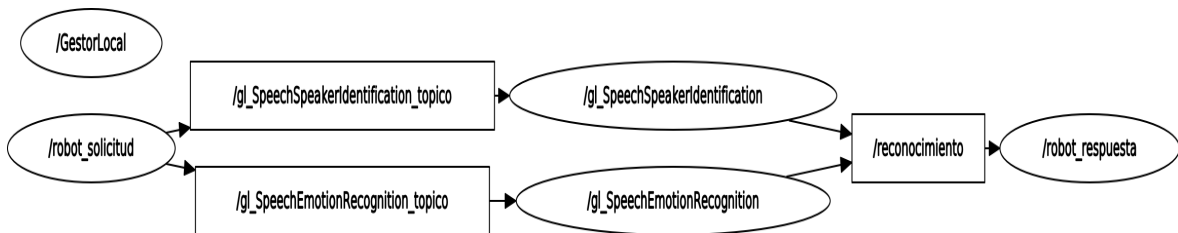


Figura 5.4: Diagrama de nodos y tópicos de la prueba P-SIS-01: Solicitudes y respuestas de reconocimiento de las tareas SpeechEmotionRecognition y SpeechSpeakerIdentification

Tabla 5.27: Caso de prueba: P-SIS-02

Id de la prueba:	P-SIS-02
Nombre del módulo:	Módulo Reconfigurable de Reconocimiento
Nombre prueba:	Activación, desactivación y solicitud de servicios de reconocimiento en el gestor remoto.
Descripción:	Prueba sobre la gestión de tareas y solicitud de servicios de reconocimiento sobre el gestor remoto
Dependencias:	Desde P-GR-01 hasta P-GR-08
Pre-condiciones:	<ol style="list-style-type: none"> 1. Gestor remoto activo. 2. Paquetes: <code>SpeechEmotionRecognition</code> y <code>SpeechSpeakerIdentification</code> instalados en el gestor remoto. 3. Estado de las tareas de reconocimiento: <code>SpeechEmotionRecognition</code> y <code>SpeechSpeakerIdentification</code>, desactivado.
Post-condiciones:	<ol style="list-style-type: none"> 1. Gestor local activo. 2. Paquetes: <code>SpeechEmotionRecognition</code> y <code>SpeechSpeakerIdentification</code> instalados en el gestor remoto. 3. Estado de las tareas de reconocimiento: <code>SpeechEmotionRecognition</code> y <code>SpeechSpeakerIdentification</code>, desactivado.
Pasos de la prueba:	<ol style="list-style-type: none"> 1. Solicitar la activación de la tarea <code>SpeechEmotionRecognition</code>. 2. Solicitar la activación de la tarea <code>SpeechSpeakerIdentification</code>. 3. Realizar solicitudes de servicios de reconocimiento a la tarea <code>SpeechEmotionRecognition</code>. 4. Realizar solicitudes de servicios de reconocimiento a la tarea <code>SpeechSpeakerIdentification</code>. 5. Solicitar la desactivación de la tarea <code>SpeechEmotionRecognition</code>. 6. Solicitar la desactivación de la tarea <code>SpeechSpeakerIdentification</code>.
Datos de la prueba:	Direcciones de los archivos a reconocer y nombre de los tópicos correspondientes a las tareas <code>SpeechEmotionRecognition</code> y <code>SpeechSpeakerIdentification</code> del gestor remoto.
Resultado:	Respuesta de las solicitudes de reconocimiento.
Estado:	Aprobado

A continuación se presentan diversos diagramas de nodos y tópicos utilizando el paquete de ROS `rqt_graph`. Cada uno de los diagramas se corresponde con uno o más pasos de la prueba P-SIS-02, especificada en la Tabla 5.27.



Figura 5.5: Diagrama de nodos y tópicos de la prueba P-SIS-02: Nodo inicial, gestor remoto.

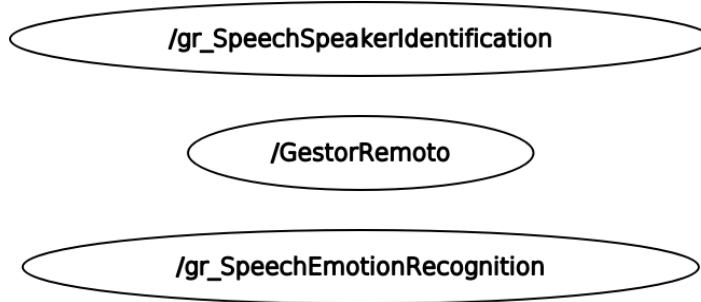


Figura 5.6: Diagrama de nodos y tópicos de la prueba P-SIS-02: Activación de las tareas SpeechEmotionRecognition y SpeechSpeakerIdentification del gestor remoto.

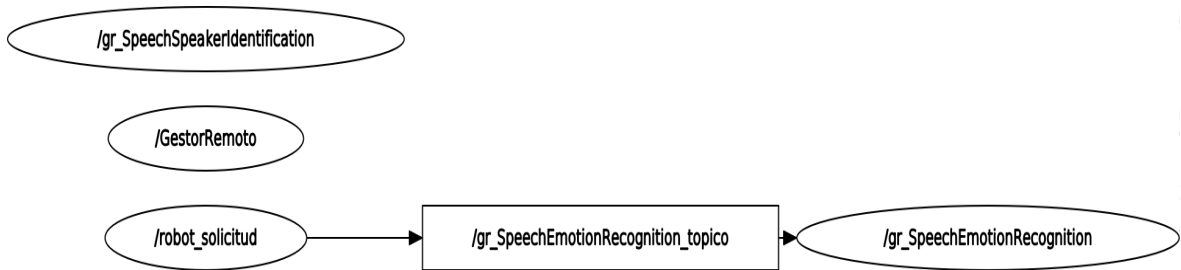


Figura 5.7: Diagrama de nodos y tópicos de la prueba P-SIS-02: Solicitudes de servicios de reconocimiento a la tarea de reconocimiento SpeechEmotionRecognition.

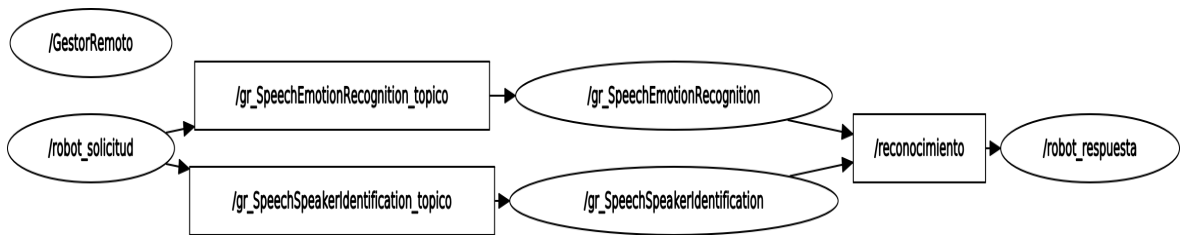


Figura 5.8: Diagrama de nodos y tópicos de la prueba P-SIS-02: Solicitudes y respuestas de reconocimiento de las tareas SpeechEmotionRecognition y SpeechSpeakerIdetification

Tabla 5.28: Caso de prueba: P-SIS-03

Id de la prueba:	P-SIS-03
Nombre del módulo:	Módulo Reconfigurable de Reconocimiento
Nombre prueba:	Instalación, activación, desactivación y reconocimiento en el gestor local y remoto.
Descripción:	Prueba sobre la gestión de tareas, gestión de paquetes y solicitudes de servicios de reconocimiento sobre el gestor local y remoto.
Dependencias:	P-SIS-01, P-SIS-02
Pre-condiciones:	<ol style="list-style-type: none"> 1. Gestor local y remoto activos. 2. Paquetes: <code>SpeechEmotionRecognition</code>, <code>SpeechSpeakerIdentification</code>, <code>SpeechGenreRecognition</code> instalados en el gestor remoto.
Post-condiciones:	<ol style="list-style-type: none"> 1. Gestor local y remoto activos. 2. Paquetes: <code>SpeechEmotionRecognition</code>, <code>SpeechSpeakerIdentification</code>, <code>SpeechGenreRecognition</code> instalados en el gestor remoto. 3. Paquete: <code>SpeechGenreRecognition</code> instalado en el gestor local.
Pasos de la prueba:	<ol style="list-style-type: none"> 1. Solicitar la activación de las tareas <code>SpeechEmotionRecognition</code> y <code>SpeechSpeakerIdentification</code>. 2. Solicitar la activación de la tarea <code>SpeechGenreRecognition</code>. 3. Realizar solicitudes de servicios de reconocimiento a la tarea <code>SpeechEmotionRecognition</code> del gestor remoto. 4. Realizar solicitudes de servicios de reconocimiento a la tarea <code>SpeechSpeakerIdentification</code> del gestor remoto. 5. Solicitar la instalación de paquete de reconocimiento: <code>SpeechGenreRecognition</code> en el gestor local. 6. Realizar solicitudes de servicios de reconocimiento a la tarea <code>SpeechGenreRecognition</code> del gestor local. 7. Solicitar la desactivación de las tareas <code>SpeechEmotionRecognition</code> y <code>SpeechSpeakerIdentification</code> del gestor remoto. 8. Solicitar la desactivación de la tarea <code>SpeechGenreRecognition</code> del gestor local. 9. Solicitar la desinstalación de paquete de reconocimiento: <code>SpeechGenreRecognition</code> en el gestor local.
Datos de la prueba:	Direcciones de los archivos a reconocer y nombre de los tópicos correspondientes a las tareas de reconocimiento.
Resultado:	Respuesta de las solicitudes de reconocimiento.
Estado:	Aprobado

A continuación se presentan diversos diagramas de nodos y tópicos utilizando el paquete de ROS `rqt_graph`. Cada uno de los diagramas se corresponde con uno o más pasos de la prueba P-SIS-03, especificada en la Tabla 5.28.



Figura 5.9: Diagrama de nodos y tópicos de la prueba P-SIS-03: Nodos iniciales, gestor local y gestor remoto.

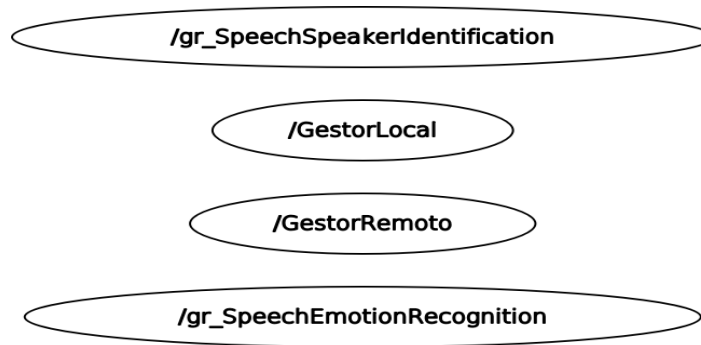


Figura 5.10: Diagrama de nodos y tópicos de la prueba P-SIS-03: Activación de las tareas `SpeechEmotionRecognition` y `SpeechSpeakerIdentification` del gestor remoto.

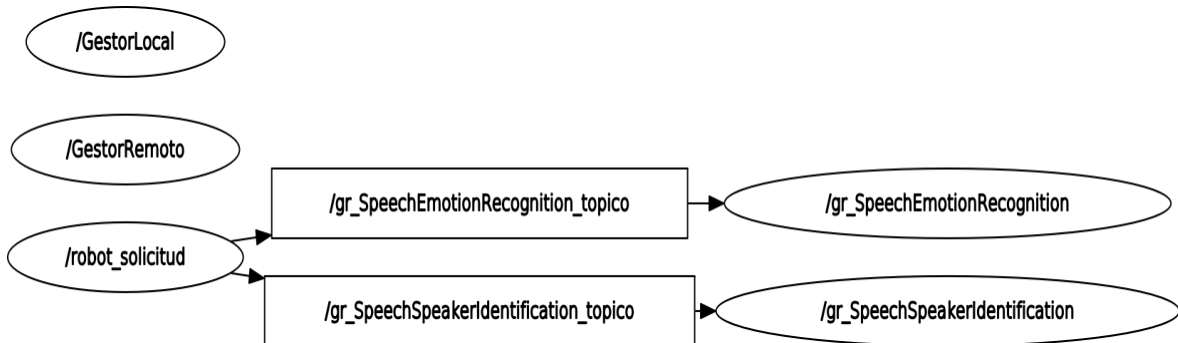


Figura 5.11: Diagrama de nodos y tópicos de la prueba P-SIS-03: Solicitudes de servicios de reconocimiento a las tareas de reconocimiento `SpeechEmotionRecognition` y `SpeechSpeakerIdentification`.

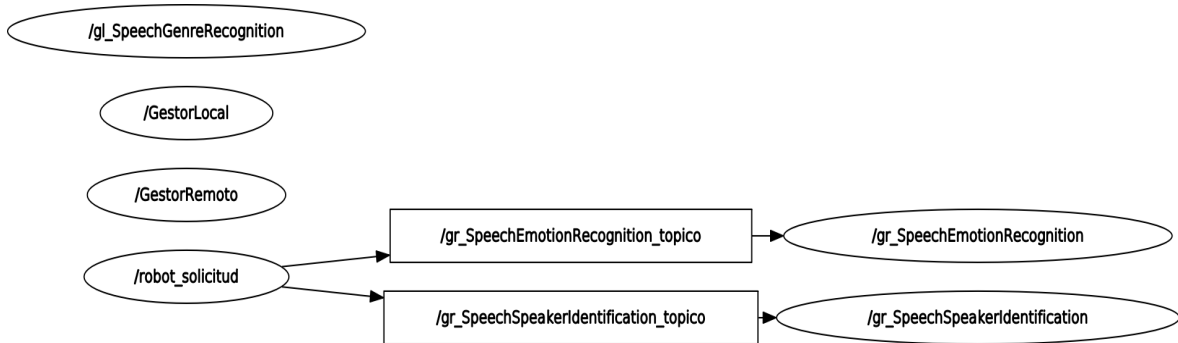


Figura 5.12: Diagrama de nodos y tópicos de la prueba P-SIS-03: Instalación y activación de la tarea de reconocimiento SpeechGenreRecognition.

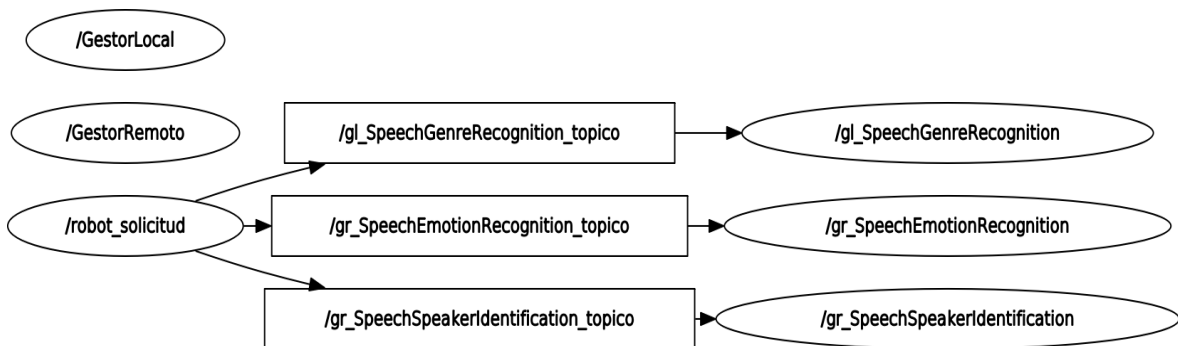


Figura 5.13: Diagrama de nodos y tópicos de la prueba P-SIS-03: Solicitudes de reconocimiento de las tareas SpeechEmotionRecognition, SpeechSpeakerIdetification y SpeechGenreRecognition.

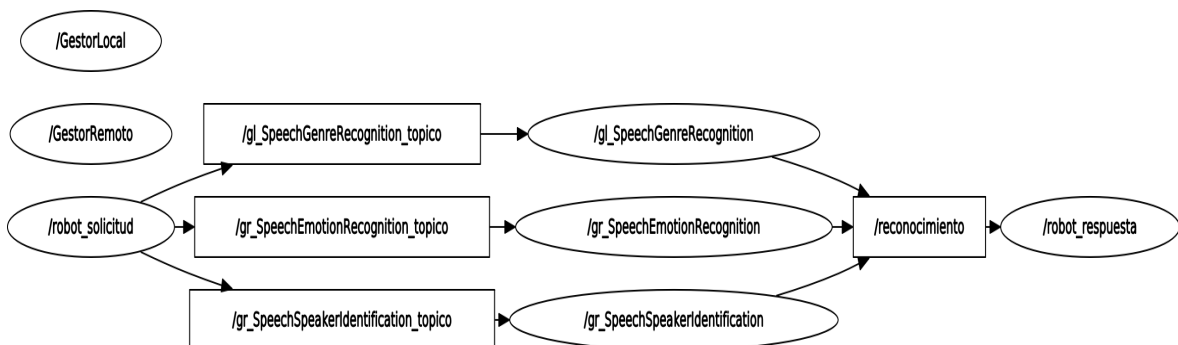


Figura 5.14: Diagrama de nodos y tópicos de la prueba P-SIS-03: Solicitudes y respuestas de reconocimiento de las tareas SpeechEmotionRecognition, SpeechSpeakerIdetification y SpeechGenreRecognition.

5.1.4. Medición de memoria y procesamiento de uno de los gestores

A continuación se muestra la medición del consumo de procesamiento y memoria de manera representativa de uno de los gestores (local o remoto). Estas pruebas de medición se corresponden con la especificación de los casos de prueba de la Tabla 5.25.

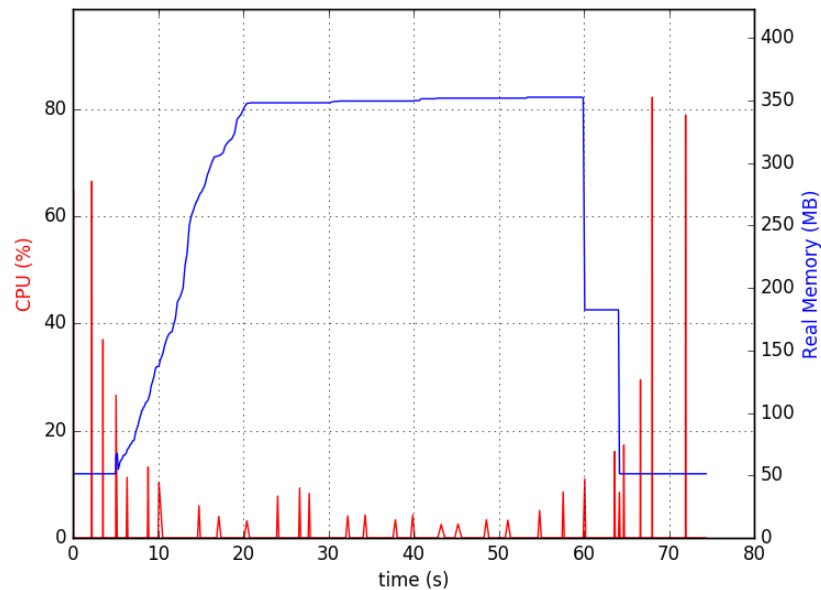


Figura 5.15: Consumo de procesamiento y memoria sobre la prueba P-SIS-01 utilizando el software psrecord

En la Figura 5.15, se puede observar el proceso de activación y desactivación de tareas de reconocimiento. Estas actividades son parte de la funcionalidad de ambos gestores. Entre los aspectos más relevantes de esta Figura, se puede observar un incremento exponencial en el consumo de memoria en la primera etapa del caso de prueba (activación de tareas de reconocimiento); luego, el consumo de memoria se mantiene estable durante un periodo de tiempo; y finalmente, la memoria se libera en la etapa final del caso de prueba (desactivación de tareas de reconocimiento). Por otro lado, los momentos de mayor consumo de procesamiento se encuentran en la inicialización y finalización del gestor. Esto se debe a que en el inicio del gestor se realiza una inicialización de diversos parámetros por cada una de las tareas instaladas,

de manera similar, al finalizar el gestor se realizan verificaciones para desactivar todas las tareas que se encuentren activas y eliminar diversos parámetros.

5.1.5. Medición de memoria y procesamiento de una de las tareas de reconocimiento

A continuación se muestra la medición del consumo de procesamiento y memoria de manera representativa de una de las tareas de reconocimiento. Estas pruebas de medición se corresponden con la especificación de los casos de prueba de la Tabla 5.25.

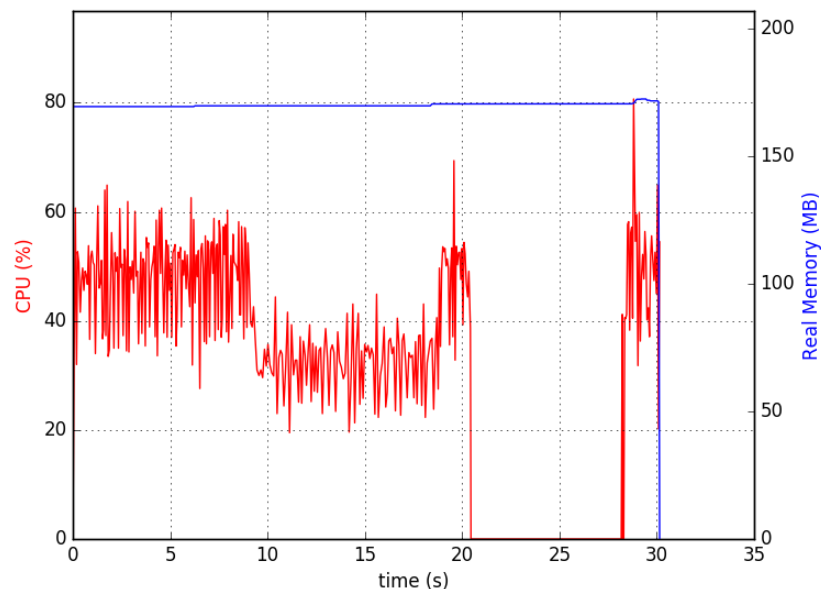


Figura 5.16: Consumo de procesamiento y memoria sobre la prueba P-SIS-01 utilizando el software psrecord

En la Figura 5.16, se puede observar el proceso de respuesta a solicitudes en las tareas de reconocimiento. Estas actividades son parte de la funcionalidad de las tareas de reconocimiento. Entre los aspectos más relevantes de esta Figura, se puede observar que el consumo de memoria es constante durante todo el proceso hasta que éste termina. Por otro lado, el consumo en procesamiento se muestra en los periodos en los cuales son realizadas las solicitudes de reconocimiento a esa tarea y en el momento de su finalización (desactivación).

5.2. Pruebas sobre el reconocedor de emociones

Cuando se trata de probar algoritmos de aprendizaje supervisado, una tarea frecuente es entrenar el modelo con datos de entrenamiento, y luego probar este modelo con datos diferentes a los de entrenamiento; esto con la finalidad de medir su capacidad para predecir datos no antes vistos. La validación cruzada es un procedimiento que permite estimar la capacidad o rendimiento de generalización en este contexto [62].

Existen dos objetivos principales en la validación cruzada:

- Estimar el rendimiento del modelo (capacidad de generalización) entrenado a partir de los datos disponibles, utilizando un algoritmo.
- Estimar el rendimiento de uno o más algoritmos diferentes y encontrar el mejor algoritmo, para los datos disponibles.

Para seguir estos objetivos diferentes técnicas han sido propuestas [62]:

- Validación por resubstitución.
- Validación cruzada de K iteraciones.
- Método de retención (Hold-out, inglés).
- Validación cruzada dejando uno afuera.

No obstante, si el objetivo es determinar la exactitud de un algoritmo de aprendizaje de máquina A, entrenado con un conjunto de datos N. El promedio de la validación cruzada puede ser utilizado como un estimador de la exactitud de A para datos no vistos, cuando A es entrenado con todos los ejemplos. Por otro lado, si el objetivo es comparar dos algoritmos de aprendizaje de máquina, los resultados de rendimiento obtenidos a través de la validación cruzada puede ser utilizados para realizar una prueba estadística de hipótesis.

A continuación se presentarán cada una de las pruebas que se realizarán a cada uno de los algoritmos de aprendizaje de máquina utilizados:

- **P1:** Pruebas de validación cruzada utilizando 10 iteraciones sobre la base de datos en condiciones controladas y actuadas.

- **P2**: Pruebas utilizando la base de datos en condiciones controladas y actuadas como entrenamiento y la base de datos en condiciones controladas y semi-natural como pruebas.
- **P3**: Cálculo de las métricas de evaluación: exactitud, precisión, sensibilidad, puntaje f1 y curva de la característica operativa del receptor (mayormente conocida como curva ROC, por sus siglas en inglés).

La prueba 1 (**P1**), permitirá evaluar los resultados y garantizar que son independientes de la partición entre datos de entrenamiento y pruebas. Adicionalmente, esta prueba tiene como ventaja una estimación precisa de la exactitud sobre datos no antes vistos.

Para realizar la prueba 2 (**P2**), se utilizará el método de retención para evaluar los modelos entrenados en la prueba 1. Este método tiene como ventaja evitar el solapamiento entre los datos de entrenamiento y validación, además, permite realizar una estimación mucho más precisa de la capacidad de generalización del algoritmo.

En la prueba 3 (**P3**), se obtendrán diferentes métricas que permitirán evaluar en profundidad las características de los modelos con respecto al desempeño obtenido en la prueba **P2**. La tasa de reconocimiento o exactitud permite estimar en promedio la calidad del clasificador; la precisión, permite estimar el costo de los falsos positivos en la clasificación; la sensibilidad, permite estimar el costo de los falsos negativos en la clasificación; por último, el puntaje F1, permite evaluar la exactitud en función de la precisión y de la sensibilidad, en otras palabras, evaluar la exactitud en función el costo de los falsos positivos y falsos negativos. Adicionalmente, se realizarán las curvas ROC de cada modelo, que permitirán representar la relación de beneficio entre la tasa de verdaderos positivos y el coste de los falsos positivos.

Inicialmente se seleccionaron 3 algoritmos para ser probados: SVM, RF y GB. No obstante, en base a los resultados obtenidos en investigaciones previas como en [50], donde se mostró que algunos algoritmos permiten reconocer ciertas emociones con una mejor capacidad. Se tomó la decisión de probar un método de aprendizaje combinado mediante votación, en la cual el resultado es elegido en función de las contribuciones individuales de cada algoritmo [9].

Los algoritmos de aprendizaje de máquina que se utilizarán para realizar cada una de las pruebas sobre el reconocedor de emociones son:

- Máquina de Vectores de Soporte (**SVM**).
- Bosques Aleatorios (**RF**).
- Aumento del Gradiente (**GB**).
- Aprendizaje combinado (EL, por sus siglas en inglés), basado en votación, utilizando los algoritmos descritos anteriormente (**EL**).

5.2.1. Prueba 1: validación cruzada

Los hiperpárametros de cada uno de los algoritmos de aprendizaje fueron optimizados utilizando 5 iteraciones. A continuación se presentan las especificaciones de cada hiperpárametro de cada uno de los algoritmos de aprendizaje de máquina utilizados:

- **SVM**: Núcleo: radial, $C = 10$, $\gamma = 0.2$.
- **RF**: Criterion: “entropy”, máxima profundidad = 13, características máximas = auto, número de estimadores = 165.
- **GB**: número de estimadores = 100, tasa de aprendizaje = 0.26, máxima profundidad = 4, mínimo número de muestras por hoja = 12, muestras mínimas de hoja = 12, criterio = “friedman_mse”.

Tabla 5.29: Exactitud de cada uno de los segmentos de prueba de la validación cruzada, $K = 10$

Algoritmo	K=1	K=2	K=3	K=4	K=5	K=6	K=7	K=8	K=9	K=10
SVM	66 %	81 %	70 %	64 %	73 %	67 %	66 %	67 %	61 %	65 %
RF	73 %	72 %	66 %	58 %	67 %	58 %	75 %	64 %	61 %	59 %
GB	67 %	73 %	66 %	60 %	72 %	62 %	75 %	62 %	62 %	58 %
EL	76 %	77 %	66 %	64 %	72 %	66 %	76 %	69 %	62 %	62 %

Tabla 5.30: Promedio y desviación estándar de la exactitud de cada uno de los segmentos de prueba de la validación cruzada, $K = 10$

Algoritmo	Promedio	Desviación estándar
SVM	68 %	5 %
RF	65 %	6 %
GB	66 %	5 %
EL	69 %	5 %

5.2.1.1. Comparativa de los algoritmos: Prueba estadística

Una de las formas de comparar el rendimiento de los algoritmos de aprendizaje de máquina según [62], es mediante el cálculo del rendimiento obtenido mediante la validación cruzada. Realizar una comparación directa utilizando el promedio de los puntajes obtenidos mediante la validación cruzada no siempre permite determinar si existe una diferencia real entre el rendimiento de dos algoritmos que se entrenan utilizando el mismo conjunto de datos. Por esta razón se utilizará una prueba no paramétrica conocida como prueba de los rangos con signo de Wilcoxon [70], la cual permitirá determinar si existe una diferencia real entre realizar aprendizaje combinado o utilizar algunos de los algoritmos seleccionados.

A continuación se presentan las siguientes pruebas estadísticas para determinar si el aprendizaje combinado presenta una mejoría real en comparación a los algoritmos seleccionados, basándose en la pruebas de validación cruzada de 10 iteraciones de la Tabla 5.29. Se utiliza como valor de significación $\alpha = 0.05$. Si el valor de la prueba (pv) es menor al valor de significación α , se rechaza la hipótesis nula a favor de la hipótesis alternativa, en caso contrario, se acepta la hipótesis nula.

$$\begin{cases} H_0 : \text{No existe diferencia entre ambos algoritmos} & pv < \alpha \\ H_1 : \text{Existe una diferencia entre ambos algoritmos} & pv \geq \alpha \end{cases} \quad (5.2.1.1)$$

En la Tabla 5.31, se pueden observar los diferentes valores del valor del prueba (pv), utilizando la prueba no paramétrica de Wilcoxon. Los resultados obtenidos muestran que no existe una diferencia entre las tasas de reconocimiento o exactitud utilizando

Tabla 5.31: Pruebas de hipótesis de Wilcoxon: Valores de las pruebas (pv) y conclusiones

Algoritmos	pv	Condición	Conclusión
EL-SVM	0.96	$pv > \alpha$	No existe diferencias entre los puntajes de exactitud de ambos algoritmos.
EL-RF	0.005	$pv < \alpha$	El algoritmo EL presenta una mejora con respecto a los resultados del algoritmo RF.
EL-GB	0.01	$pv < \alpha$	El algoritmo EL presenta una mejora con respecto a los resultados del algoritmo GB.

el algoritmo Máquina de Vectores de Soporte (SVM) y el aprendizaje combinado (EL) presentes la prueba de validación cruzada (ver Tabla 5.29). No obstante, los resultados restantes mostraron que utilizando el algoritmo basado en aprendizaje combinado (EL), se presenta una mejora con respecto a los algoritmos Bosques Aleatorios (RF) y Aumento del Gradiente (GB).

5.2.2. Prueba 2: pruebas de retención

A continuación se realizarán pruebas de retención. Estas pruebas están orientadas a realizar una estimación de la capacidad de generalización de los algoritmos [62]. Se probarán las muestras de la base de datos en condiciones controladas y semi-natural, utilizando los modelos correspondientes a cada uno de los algoritmos seleccionados (SVM, RF, GB, EL), entrenados con las muestras de la base de datos en condiciones controladas y actuadas. Adicionalmente, se realizará una comparativa con los resultados obtenidos en la Tabla 5.31, basado en las tasas de reconocimiento o valores de exactitud obtenidos en las pruebas realizadas sobre cada uno de los modelos.

5.2.2.1. Resultados: algoritmo SVM

Actual	ira	57.1% 109/191	0.5% 1	7.3% 14	7.3% 14	27.2% 52	0.5% 1
	tristeza		89.1% 188/211	3.8% 8			7.1% 15
	asco	3.1% 6	6.7% 13	66.8% 129/193	1.6% 3	21.2% 41	0.5% 1
	felicidad	14.7% 27		5.4% 10	57.1% 105/184	20.7% 38	2.2% 4
	sorpresa	17.3% 35	0.5% 1	18.8% 38	8.4% 17	54.0% 109/202	1.0% 2
	miedo	0.6% 1	9.7% 17	6.8% 12	1.1% 2	5.1% 9	76.7% 135/176
		ira	tristeza	asco	felicidad	sorpresa	miedo
		Predicción					

Figura 5.17: Prueba de retención utilizando el algoritmo SVM: tasa de reconocimiento: 67%

En la Figura 5.17, se puede observar los porcentajes de las tasas de reconocimiento o exactitud, de cada una de las emociones presentes en la base de datos controlada y semi-natural. Adicionalmente, se puede observar que utilizando el modelo entrenado con el algoritmo SVM, la tristeza es la emoción que mejor tasa de reconocimiento tiene. Por otro lado, la sorpresa fue la emoción que menor tasa de reconocimiento obtuvo.

5.2.2.2. Resultados: algoritmo RF

Actual	ira	61.8% 118/191	1.0% 2	8.4% 16	8.9% 17	18.8% 36	1.0% 2
	tristeza		91.0% 192/211	2.4% 5		0.9% 2	5.7% 12
	asco	2.6% 5	6.2% 12	74.1% 143/193	3.1% 6	10.9% 21	3.1% 6
	felicidad	13.6% 25	1.6% 3	3.3% 6	64.1% 118/184	14.7% 27	2.7% 5
	sorpresa	15.8% 32	1.5% 3	11.4% 23	10.9% 22	56.9% 115/202	3.5% 7
	miedo	0.6% 1	17.6% 31	4.5% 8		1.1% 2	76.1% 134/176
		ira	tristeza	asco	felicidad	sorpresa	miedo
		Predicción					

Figura 5.18: Prueba de retención utilizando el algoritmo RF: tasa de reconocimiento: 70%

En la Figura 5.18, se pueden observar los porcentajes de las tasas de reconocimiento o exactitud, de cada una de las emociones presentes en la base de datos controlada y semi-natural. Adicionalmente, se puede observar que utilizando el modelo entrenado con el algoritmo RF, de manera similar a las pruebas utilizando el algoritmo SVM (ver Figura 5.17), la tristeza es la emoción que mayor tasa de reconocimiento tiene y la sorpresa es la emoción que menor tasa de reconocimiento tiene.

5.2.2.3. Resultados: algoritmo GB

Actual	ira	63.9% 122/191	0.5% 1	8.9% 17	5.2% 10	20.4% 39	1.0% 2
	tristeza	0.9% 2	87.2% 184/211	3.8% 8			8.1% 17
	asco	4.1% 8	6.2% 12	71.0% 137/193	3.1% 6	10.9% 21	4.7% 9
	felicidad	14.7% 27	0.5% 1	3.8% 7	60.3% 111/184	20.1% 37	0.5% 1
	sorpresa	18.3% 37	1.5% 3	6.4% 13	11.9% 24	60.9% 123/202	1.0% 2
	miedo	1.1% 2	15.3% 27	4.0% 7	2.8% 5	1.1% 2	75.6% 133/176
	Predicción	ira	tristeza	asco	felicidad	sorpresa	miedo

Figura 5.19: Prueba de retención utilizando el algoritmo GB: tasa de reconocimiento: 70%

En la Figura 5.19, se puede observar los porcentajes de las tasas de reconocimiento o exactitud, de cada una de las emociones presentes en la base de datos controlada y semi-natural. Adicionalmente, se puede observar que utilizando el modelo entrenado con el algoritmo GB, la tristeza es la emoción que mayor tasa de reconocimiento obtuvo. No obstante, a diferencia de las pruebas realizadas utilizando los algoritmos SVM y RF (ver Figuras 5.17 y 5.18), la felicidad fue la emoción que menor tasa de reconocimiento obtuvo.

5.2.2.4. Resultados: algoritmo EL

Actual	ira	66.0% 126/191	0.5% 1	8.4% 16	6.8% 13	17.3% 33	1.0% 2
	tristeza		91.5% 193/211	2.8% 6			5.7% 12
	asco	2.6% 5	6.2% 12	74.1% 143/193	2.6% 5	11.9% 23	2.6% 5
	felicidad	14.7% 27	0.5% 1	4.9% 9	62.5% 115/184	16.8% 31	0.5% 1
	sorpresa	16.3% 33	1.0% 2	9.4% 19	8.9% 18	63.9% 129/202	0.5% 1
	miedo	0.6% 1	12.5% 22	4.5% 8	1.7% 3	1.7% 3	79.0% 139/176
		ira	tristeza	asco	felicidad	sorpresa	miedo
		Predicción					

Figura 5.20: Prueba de retención utilizando el algoritmo EL: tasa de reconocimiento: 73%

En la Figura 5.20, se puede observar los porcentajes de las tasas de reconocimiento o exactitud, de cada una de las emociones presentes en la base de datos controlada y semi-natural. Adicionalmente, se puede observar que utilizando el modelo entrenado basando en el aprendizaje combinado de los algoritmos SVM, RF y GB; la tristeza es la emoción que mayor tasa de reconocimiento obtuvo. La felicidad y la sorpresa fueron las emociones que menor tasa de reconocimiento obtuvieron.

5.2.3. Prueba 3: Cálculo de métricas

5.2.3.1. Precisión, sensibilidad y puntaje F1

A continuación se presentan las métricas: exactitud, precisión, sensibilidad, puntaje f1 y curva ROC de cada uno de los algoritmos y técnicas seleccionados (SVM, RF, GB, EL). Los resultados de estas pruebas, están basados en los resultados de la prueba 2, en la cual se prueban los modelos que son entrenados con las muestras de la base de datos en condiciones controladas y actuadas, y son probadas con las muestras de la base de datos en condiciones controladas y semi-naturales.

Tabla 5.32: Exactitud o tasa de reconocimiento de cada uno de los modelos.

Algoritmo	Exactitud
SVM	66 %
RF	70 %
GB	70 %
EL	73 %

En la Tabla 5.32, se pueden observar las tasas de reconocimiento o exactitud de cada uno de los modelos. Los resultados obtenidos muestran que la técnica basada en aprendizaje combinado obtiene los mejores resultados. La matriz de confusión correspondiente a esta técnica se encuentra en la Figura 5.20.

Tabla 5.33: Métricas obtenidas utilizando el algoritmo SVM

Emoción	Precisión	Sensibilidad	Puntaje F1
Ira	61 %	57 %	59 %
Tristeza	85 %	89 %	87 %
Asco	61 %	64 %	64 %
Felicidad	74 %	65 %	65 %
Sorpresa	44 %	48 %	48 %
Miedo	85 %	81 %	81 %
Promedio	68 %	67 %	67 %

En la Tabla 5.33, se pueden encontrar las métricas de precisión, sensibilidad y puntaje F1 para del modelo utilizando el algoritmo SVM. Las emociones tristeza y miedo tienen la mayor precisión, sensibilidad y puntaje F1, mientras que la sorpresa es la emoción que menor puntaje tiene en todas las métricas.

Tabla 5.34: Métricas obtenidas utilizando el algoritmo RF

Emoción	Precisión	Sensibilidad	Puntaje F1
Ira	65 %	62 %	63 %
Tristeza	79 %	91 %	85 %
Asco	71 %	74 %	73 %
Felicidad	72 %	64 %	68 %
Sorpresa	57 %	57 %	57 %
Miedo	81 %	76 %	78 %
Promedio	71 %	71 %	71 %

En la Tabla 5.34, se pueden encontrar las métricas de precisión, sensibilidad y puntaje F1 para del modelo utilizando el algoritmo RF. De manera similar a la Tabla 5.33, las emociones tristeza y miedo tienen la mayor precisión, sensibilidad y puntaje F1, mientras que la sorpresa es la emoción que menor puntaje tiene en todas las métricas. No obstante, en promedio las métricas obtenidas por este modelo son superiores a las métricas obtenidas utilizando el modelo SVM.

Tabla 5.35: Métricas obtenidas utilizando el algoritmo GB

Emoción	Precisión	Sensibilidad	Puntaje F1
Ira	62 %	64 %	63 %
Tristeza	81 %	87 %	84 %
Asco	72 %	71 %	72 %
Felicidad	71 %	60 %	65 %
Sorpresa	55 %	61 %	58 %
Miedo	81 %	76 %	78 %
Promedio	70 %	70 %	70 %

En la Tabla 5.35, se pueden encontrar las métricas de precisión, sensibilidad y puntaje F1 para del modelo utilizando el algoritmo RF. De manera similar a las Tablas 5.33 y 5.34, las emociones tristeza y miedo tienen la mayor precisión, sensibilidad y puntaje F1, mientras que la sorpresa es la emoción que menor puntaje tiene en todas las métricas. En promedio, las métricas obtenidas son superiores a las métricas del modelo SVM (ver Tabla 5.33), no obstante inferiores a las de las métricas del modelo RF (ver Tabla 5.34).

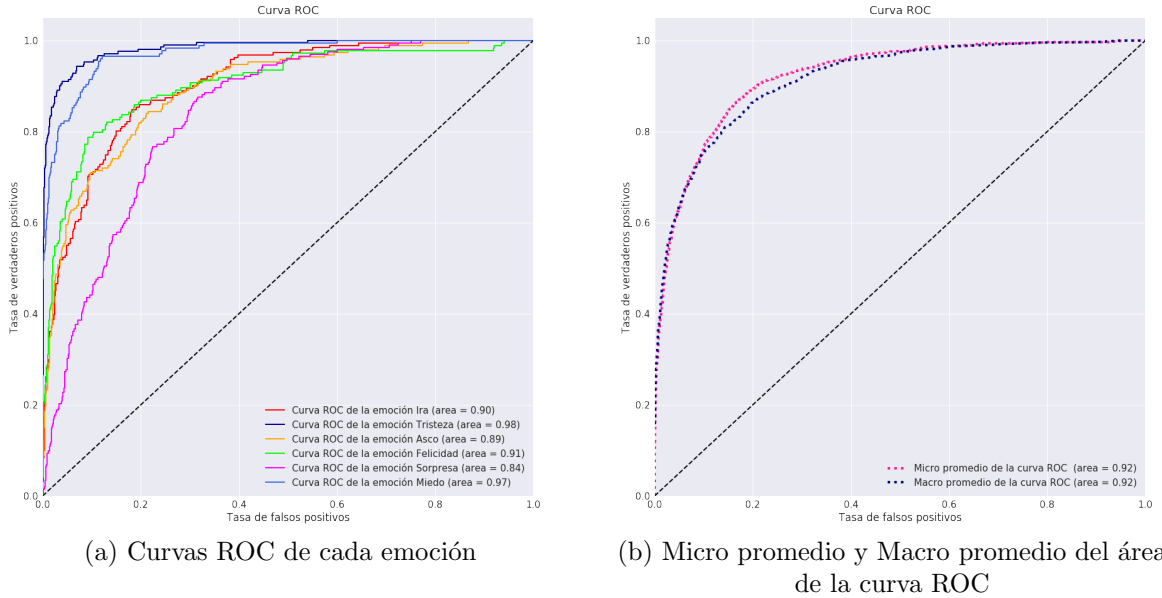
En la Tabla 5.36, se pueden encontrar las métricas de precisión, sensibilidad y puntaje F1 para del modelo utilizando el algoritmo RF. De manera similar a las Tablas

Tabla 5.36: Métricas obtenidas utilizando la técnica basada en aprendizaje combinado (EL)

Emoción	Precisión	Sensibilidad	Puntaje F1
Ira	65 %	66 %	66 %
Tristeza	83 %	91 %	87 %
Asco	71 %	74 %	72 %
Felicidad	75 %	62 %	68 %
Sorpresa	59 %	63 %	61 %
Miedo	87 %	80 %	83 %
Promedio	73 %	73 %	73 %

5.33, 5.34 y 5.36, las emociones tristeza y miedo tienen la mayor precisión, sensibilidad y puntaje F1, mientras que la sorpresa es la emoción que menor puntaje tiene en todas las métricas. En promedio, las métricas obtenidas son superiores a todos los modelos anteriores. Adicionalmente, es importante resaltar que existe un incremento notorio en las métricas obtenidas con respecto a la emoción sorpresa.

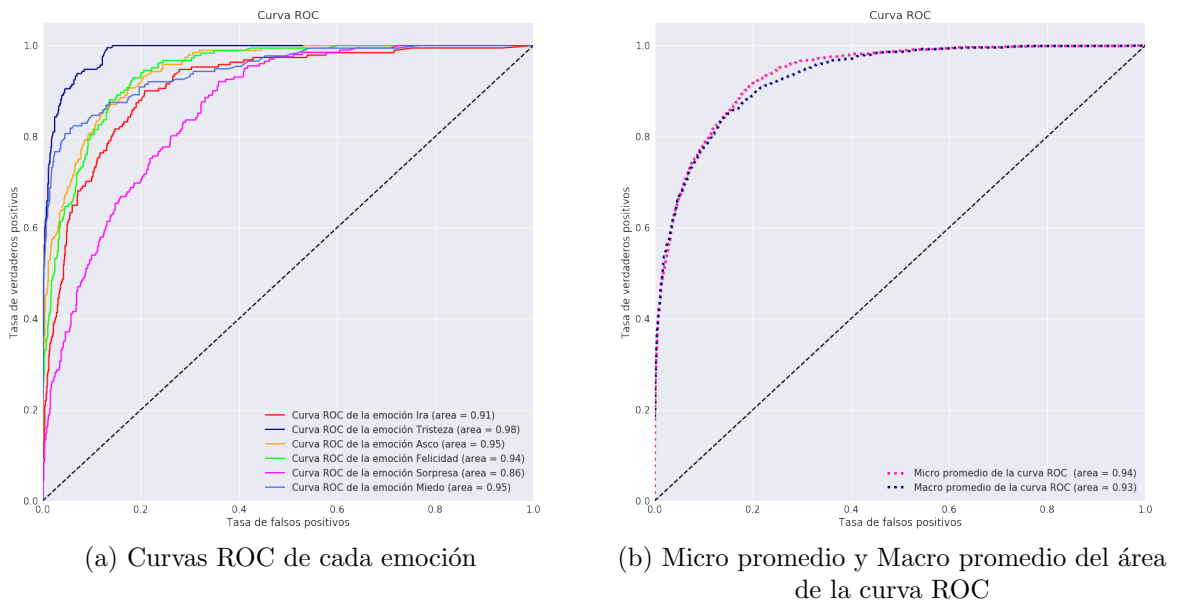
5.2.3.2. Curvas ROC



(a) Curvas ROC de cada emoción

(b) Micro promedio y Macro promedio del área de la curva ROC

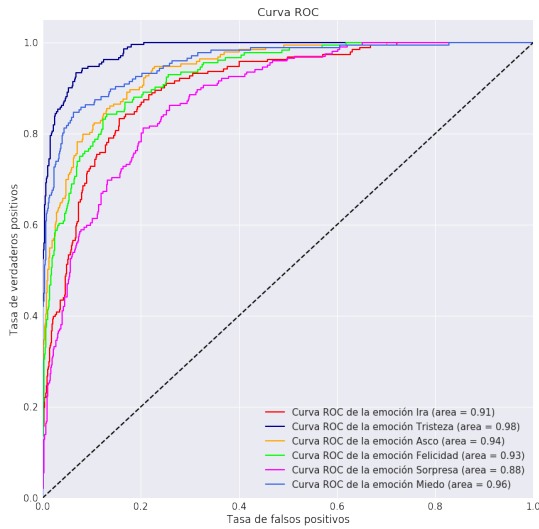
Figura 5.21: Áreas bajo la curva ROC de cada emoción utilizando el algoritmo SVM



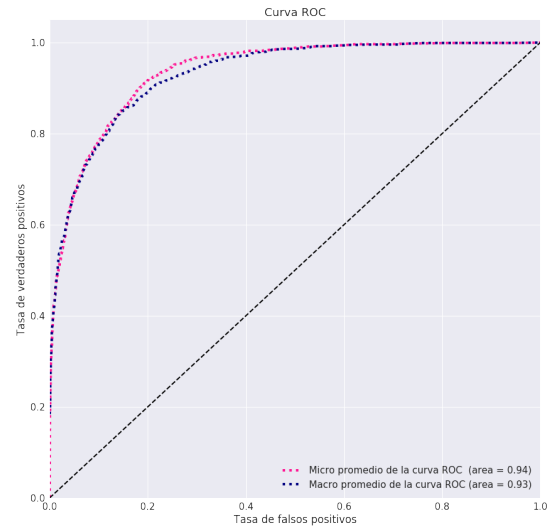
(a) Curvas ROC de cada emoción

(b) Micro promedio y Macro promedio del área de la curva ROC

Figura 5.22: Áreas bajo la curva ROC de cada emoción utilizando el algoritmo RF

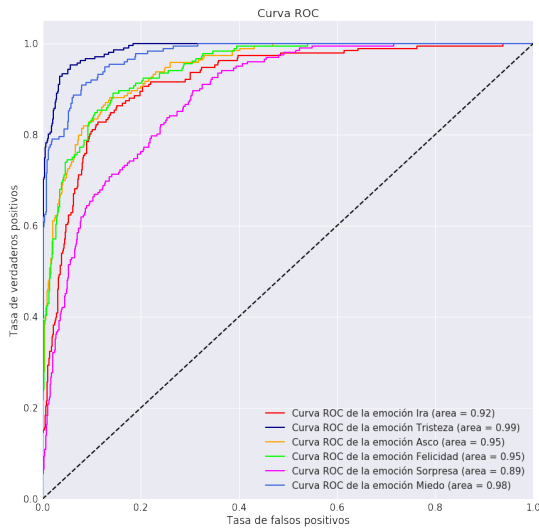


(a) Curvas ROC de cada emoción

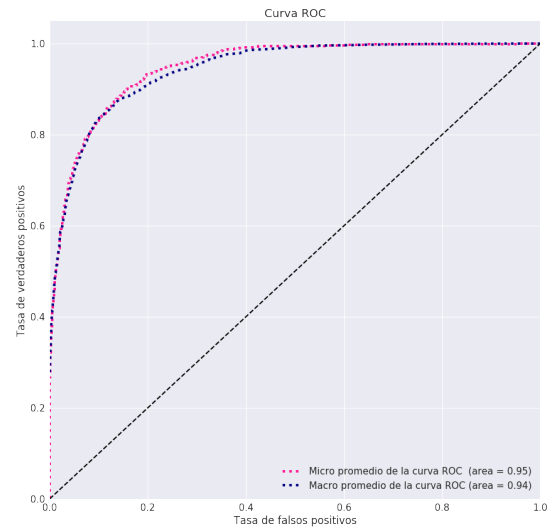


(b) Micro promedio y Macro promedio del área de la curva ROC

Figura 5.23: Áreas bajo la curva ROC de cada emoción utilizando el algoritmo GB



(a) Curvas ROC de cada emoción



(b) Micro promedio y Macro promedio del área de la curva ROC

Figura 5.24: Áreas bajo la curva ROC de cada emoción utilizando la técnica de aprendizaje combinado EL

Tabla 5.37: Macro promedio y micro promedio del área bajo la curva ROC

Emoción	SVM	RF	GB	EL
Ira	0.90	0.91	0.91	0.92
Tristeza	0.98	0.98	0.98	0.99
Asco	0.89	0.95	0.94	0.95
Felicidad	0.91	0.94	0.93	0.95
Sorpresa	0.84	0.86	0.88	0.89
Miedo	0.97	0.95	0.96	0.98
Micro promedio	0.92	0.94	0.94	0.95
Macro promedio	0.92	0.93	0.94	0.94

En la Tabla 5.37, se puede encontrar de manera resumida el área bajo la curva ROC de cada uno de los algoritmos y técnicas utilizadas (ver Figuras 5.21, 5.22, 5.23, 5.24), adicionalmente, se encuentra el micro promedio y macro promedio. Los resultados obtenidos en la prueba P3, son consistentes con los resultados obtenidos en las pruebas P1 y P2 (ver Secciones 5.2.1 y 5.2.2), debido a que en estas pruebas las conclusiones han sido similares: existe una evidente mejora cuando se utiliza la técnica basada en aprendizaje combinado.

Los resultados de las diferentes métricas obtenidas (ver Tablas 5.33, 5.33, 5.33 y 5.33), muestran una evidente mejora con respecto al coste de los falsos positivos (precisión) y falsos negativos (sensibilidad), cuando se realizan las pruebas con el modelo basado en aprendizaje combinado (EL). Adicionalmente, se observa una mejora en cada una de las emociones con respecto al puntaje F1.

Una de las características principales del área bajo la curva ROC, es que a medida de que ésta se acerca hacia a 1, el grado de separabilidad entre las clases es mayor. En los resultados obtenidos en el cálculo del área bajo la curva de cada uno de los modelos (ver Tabla 5.37), se pudo observar que la tristeza y el miedo son las clases con mayor grado de separabilidad en todas las pruebas, adicionalmente, la sorpresa es la emoción con menor grado de separabilidad.

5.2.4. Análisis de los resultados

Basándose en los resultados obtenidos en las pruebas de validación cruzada, las pruebas de retención y las pruebas basadas en el cálculo de métricas (ver Secciones 5.2.1, 5.2.2 y 5.2.3); realizadas sobre los modelos entrenados con la base de datos en condiciones controladas y actuadas; y utilizando los algoritmos y técnicas propuestas: SVM, RF, GB y EL. Se pueden realizar las siguientes observaciones:

- Las pruebas de validación cruzada en $K=10$ iteraciones (ver Sección 5.2.1), permitieron estimar la exactitud de los algoritmos sobre datos no vistos. En estos resultados se observó que utilizando la técnica de aprendizaje combinado (EL), se presentaba una mejora con respecto a los algoritmos RF y GB. Considerando las pruebas realizadas sobre la base de datos en condiciones controladas y actuadas, se pudo observar que esta técnica obtuvo la mayor tasa de reconocimiento.
- Las pruebas de retención (ver Sección 5.2.2), permitieron estimar la capacidad de generalización de cada uno de los algoritmos. En estos resultados se observó que la tristeza fue la emoción que mejor tasa de reconocimiento obtuvo en cada uno de los algoritmos. Adicionalmente, la felicidad, la ira y la sorpresa, fueron las emociones que menor tasa de reconocimiento obtuvieron en las pruebas realizadas. La mayor tasa de reconocimiento se obtuvo mediante la técnica de aprendizaje combinado (EL).
- Las emociones que obtuvieron menor tasa de reconocimiento (ira, sorpresa y felicidad), en las pruebas de retención, se caracterizan por contener un alto nivel de energía. A pesar de que existe una fuerte distinción entre las muestras de estas emociones, los resultados obtenidos pueden sugerir que utilizando únicamente información paraverbal, no es suficiente para discriminar de manera eficiente estas emociones.

- Las pruebas basadas en el cálculo de métricas (ver Sección 5.2.3), permitieron estimar el costo de los falsos negativos y falsos positivos en la clasificación de cada uno de los modelos. Adicionalmente, se realizó el cálculo de la curva de ROC, la cual permitió estimar el beneficio de los verdaderos positivos con respecto al costo de los falsos positivos en cada una de las emociones de cada uno de los modelos seleccionados. En estos resultados se observó que utilizando la técnica de aprendizaje combinado, el área bajo la curva de cada una de las emociones incrementó notablemente; adicionalmente esta técnica obtuvo el micro-promedio de área más cercano a 1 (ver Tabla 5.37).

5.3. Pruebas sobre el reconocedor de personas

En el área de reconocimiento de personas, existen dos tipos de tareas frecuentes: Identificación y verificación. En la tarea de identificación, el estimador utilizado para evaluar la calidad del identificador es la tasa de exactitud o tasa de correcta identificación (CIR, por sus siglas en inglés), la cual viene dada por:

$$CIR = \frac{\text{Número de peticiones correctamente identificadas}}{\text{Número total de identificaciones}} \times 100 \quad (5.3.0.1)$$

En la tarea de verificación, son comunes dos tipos de errores para evaluar la calidad de la verificación [8], entre estos se encuentran: la tasa de error de falso rechazo (FRR, por sus siglas en inglés) y la tasa de error de falsa aceptación (FAR, por sus siglas en inglés). El error de falso rechazo ocurre cuando la persona correcta es rechazada por el sistema, mientras que el error de falsa aceptación ocurre cuando la persona incorrecta es aceptada por el sistema, adicionalmente, se utilizará la tasa de aceptación correcta (SAR, por sus siglas en inglés). Estas tres métricas vienen dadas por:

$$FRR = \frac{\text{Número de falsos rechazos}}{\text{Número total de peticiones}} \times 100 \quad (5.3.0.2)$$

$$FAR = \frac{\text{Número de falsas aceptaciones}}{\text{Número total de peticiones}} \times 100 \quad (5.3.0.3)$$

$$SAR = \frac{\text{Número de peticiones correctamente aceptadas}}{\text{Número total de peticiones}} \times 100 \quad (5.3.0.4)$$

5.3.1. Pruebas de identificación de personas

A continuación se presentan las pruebas que se realizarán sobre el identificador de personas:

- **P1-A:** Tasa de identificación correcta (CIR), utilizando 16 componentes para el entrenamiento de los modelos estadísticos de cada usuario. Para los modelos estadísticos, se utilizarán como datos de entrenamiento las muestras tomadas en la primera sesión.
- **P1-B:** Tasa de identificación correcta (CIR), utilizando 16 componentes para el entrenamiento de los modelos estadísticos de cada usuario. Para los modelos estadísticos, se utilizarán como datos de entrenamiento las muestras tomadas en la primera y segunda sesión.
- **P2-A:** Tasa de identificación correcta (CIR), utilizando 32 componentes para el entrenamiento de los modelos estadísticos de cada usuario. Se utilizarán como datos de entrenamiento las muestras tomadas en la primera sesión.
- **P2-B:** Tasa de identificación correcta (CIR), utilizando 32 componentes para el entrenamiento de los modelos estadísticos de cada usuario. Para los modelos estadísticos, se utilizarán como datos de entrenamiento las muestras tomadas en la primera y segunda sesión.

El cambio de los componentes para el cálculo de la tasa de identificación correcta (CIR), permite determinar el número de componentes que mejor representa las muestras de audio de cada usuario. Adicionalmente, para observar el beneficio de la variabilidad intra-locutor, se realizarán pruebas utilizando las muestras de los usuarios en la primera y segunda sesión.

Las pruebas realizadas para la identificación de personas se realizarán con 5 usuarios cuyas muestras se tomaron en dos sesiones.

5.3.1.1. Prueba: P1-A

En la Figura 5.25, se encuentra la matriz de confusión, en donde se puede observar la tasa de correcta identificación en la diagonal principal y los errores de identificación para cada uno de los usuarios. Este valor representa el número de veces que cada usuario fue identificado correctamente con respecto al número de peticiones que se realizaron.

Actual \ Predicción	U01_M	U02_F	U03_M	U04_M	U05_F
U01_M	72.9% 137/188	1.6% 3	11.2% 21	13.3% 25	1.1% 2
U02_F	1.1% 2	93.0% 172/185			5.9% 11
U03_M	18.3% 34	0.5% 1	73.1% 136/186	8.1% 15	
U04_M	9.2% 17		6.5% 12	84.3% 156/185	
U05_F		0.6% 1	0.6% 1		98.9% 178/180

Figura 5.25: Matriz de confusión de la identificación de usuarios: tasa de identificación correcta utilizando 16 componentes y una sola sesión.

En la Figura 5.25, se pueden observar las tasas de correcta identificación de cada uno de los usuarios, y adicionalmente se puede observar que el usuario 05_F y el usuario 02_F tienen el CIR más alto y por consiguiente, los errores de identificación son menores con respecto a otros usuarios. Los usuarios 01_M, 03_M y 04_M, son los que más errores de identificación obtuvieron.

5.3.1.2. Prueba: P1-B

En la Figura 5.26, se encuentra la matriz de confusión, en donde se puede observar la tasa de correcta identificación en la diagonal principal y los errores de identificación para cada uno de los usuarios. En este caso cada uno de los modelos estadísticos utiliza 16 componentes y es entrenado con las muestras de la primera y segunda sesión.

Actual	U01_M	77.1% 145/188	1.6% 3	11.2% 21	9.6% 18	0.5% 1
	U02_F	0.5% 1	97.3% 180/185			2.2% 4
	U03_M	9.1% 17	0.5% 1	82.3% 153/186	8.1% 15	
	U04_M	8.6% 16		5.9% 11	85.4% 158/185	
	U05_F		3.9% 7			96.1% 173/180
		U01_M	U02_F	U03_M	U04_M	U05_F
		Predicción				

Figura 5.26: Matriz de confusión de la identificación de usuarios: tasa de identificación correcta utilizando 16 componentes y multisesión

En la Figura 5.26, se pueden observar las tasas de correcta identificación de cada uno de los usuarios, cuando los modelos estadísticos (GMMs), son entrenados con muestras de la primera y segunda sesión. Dentro de los cambios relevantes con respecto a los resultados de la Figura 5.25, se encuentran: el usuario 05_F disminuyó su tasa de correcta identificación, mientras que las tasas de los demás usuarios aumentaron notablemente.

5.3.1.3. Prueba: P2-A

En la Figura 5.27, se encuentra la matriz de confusión, en donde se puede observar la tasa de correcta identificación en la diagonal principal y los errores de identificación para cada uno de los usuarios. Este valor representa el número de veces que cada usuario fue identificado correctamente con respecto al número de peticiones que se realizaron.

Actual \ Predicción	U01_M	U02_F	U03_M	U04_M	U05_F
U01_M	73.4% 138/188	3.7% 7	9.6% 18	12.8% 24	0.5% 1
U02_F	0.5% 1	89.2% 165/185			10.3% 19
U03_M	14.0% 26	1.1% 2	75.8% 141/186	8.1% 15	1.1% 2
U04_M	2.2% 4		11.4% 21	86.5% 160/185	
U05_F		2.8% 5			97.2% 175/180

Figura 5.27: Matriz de confusión de la identificación de usuarios: tasa de identificación correcta utilizando 32 componentes y una sola sesión

En la Figura 5.27, se pueden observar las tasas de correcta identificación de cada uno de los usuarios. Para el entrenamiento de los modelos estadísticos (GMMs) se utilizaron 32 componentes con muestras de una única sesión. Con respecto a los resultados utilizando 16 componentes y una única sesión (ver Figura 5.25), se puede observar que el promedio de las tasas de correcta identificación de cada uno de los usuarios son iguales (ver Tabla 5.39).

5.3.1.4. Prueba: P2-B

En la Figura 5.28, se encuentra la matriz de confusión, en donde se puede observar la tasa de correcta identificación en la diagonal principal y los errores de identificación para cada uno de los usuarios. Este valor representa el número de veces que cada usuario fue identificado correctamente con respecto al número de peticiones que se realizaron.

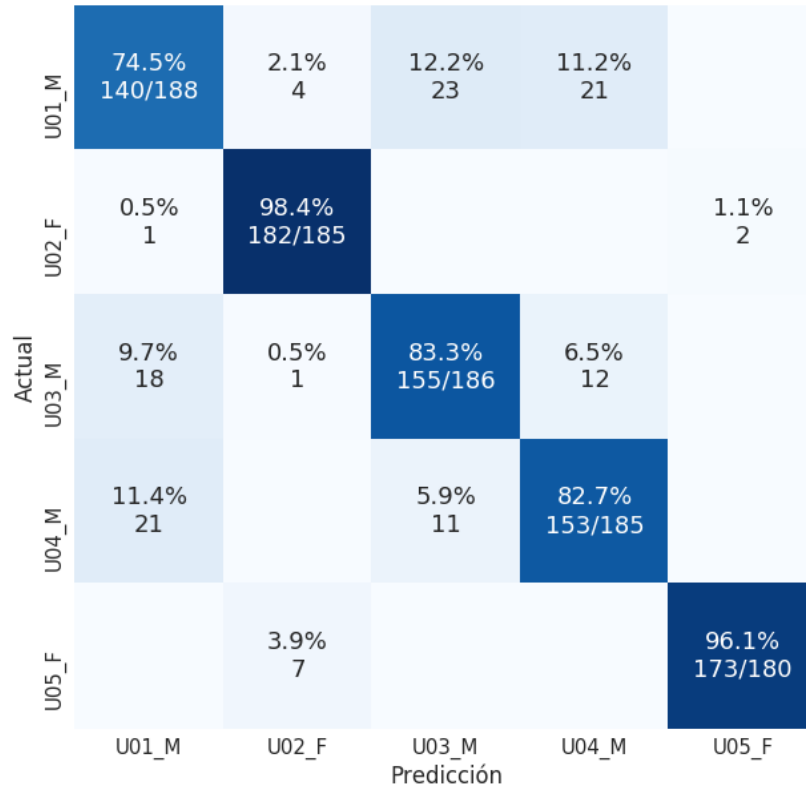


Figura 5.28: Matriz de confusión de la identificación de usuarios: tasa de identificación correcta utilizando 32 componentes y multisesión

En la Figura 5.28, se pueden observar las tasas de correcta identificación de cada uno de los usuarios. Para el entrenamiento de los modelos estadísticos (GMMs) se utilizaron 32 componentes con muestras de la primera y segunda sesión. Con respecto a los resultados utilizando 32 componentes y una única sesión (ver Figura 5.27), se puede observar que para los usuarios 04_M y 05_F existe una pequeña disminución en la tasa de correcta identificación, no obstante, en el resto de usuarios la tasa incrementa de manera notoria.

Tabla 5.38: Tasa de correcta identificación, utilizando 16 componentes con sesión única y sesión múltiple para los usuarios 01_M, 02_F, 03_M, 04_M, 05_F

Usuario	CIR, sesión única	Usuario	CIR, sesión múltiple
01_M	72.9 %	01_M	77.1 %
02_F	93.0 %	02_F	97.3 %
03_M	73.1 %	03_M	82.3 %
04_M	84.3 %	04_M	85.4 %
05_F	98.9 %	05_F	96.1 %
Promedio	84.4 %	Promedio	87.6 %

Tabla 5.39: Tasa de correcta identificación, utilizando 32 componentes con sesión única y sesión múltiple para los usuarios 01_M, 02_F, 03_M, 04_M, 05_F

Usuario	CIR, sesión única	Usuario	CIR, sesión múltiple
01_M	73.4 %	01_M	74.5 %
02_F	89.2 %	02_F	98.4 %
03_M	75.8 %	03_M	83.3 %
04_M	86.5 %	04_M	82.7 %
05_F	97.2 %	05_F	96.1 %
Promedio	84.4 %	Promedio	87 %

En las Tablas 5.38 y 5.39, se pueden encontrar las tasas de correcta identificación de cada uno de los usuarios para 16 y 32 componentes, utilizando una única sesión y sesión múltiple. En estas tablas es importante resaltar dos cosas: entrenar los modelos con muestras de la primera y segunda sesión, incrementa la tasa de correcta identificación en cada una de las pruebas realizadas (16 componentes y 32 componentes). Adicionalmente, entrenando los modelos con 16 componentes y muestras de la primera y segunda sesión se obtienen los mejores resultados de identificación 87.6 % (Ver Tabla 5.38).

5.3.2. Pruebas de verificación de personas

A continuación se presentan las pruebas que se realizarán sobre el verificador de personas:

- **P3-A:** Cálculo de la tasa del error de falsa aceptación (FAR), la tasa del error de falso rechazo (FRR) y la tasa de correcta aceptación (SAR). Utilizando 16 componentes en los modelos estadísticos y las muestras de una sola sesión.
- **P3-B:** Cálculo de la tasa del error de falsa aceptación (FAR), la tasa del error de falso rechazo (FRR) y la tasa de correcta aceptación (SAR). Utilizando 16 componentes en los modelos estadísticos y las muestras de la primera y segunda sesión.
- **P4-A:** Cálculo de la tasa del error de falsa aceptación (FAR), la tasa del error de falso rechazo (FRR) y la tasa de correcta aceptación (SAR). Utilizando 32 componentes en los modelos estadísticos y las muestras de una sola sesión.
- **P4-B:** Cálculo de la tasa del error de falsa aceptación (FAR), la tasa del error de falso rechazo (FRR) y la tasa de correcta aceptación (SAR). Utilizando 32 componentes en los modelos estadísticos y las muestras de la primera y segunda sesión.

Para todas las pruebas anteriores, el modelo universal, es entrenado con las muestras de 3 personas que realizaron una única sesión en la construcción de la base de datos para el reconocimiento de personas, adicionalmente, se utilizaron diversas muestras de la base de datos para el reconocimiento de emociones para equilibrar las muestras según el genero y aumentar la variedad de posibles expresiones. En total se consideraron 1000 muestras aproximadamente.

De manera similar a las pruebas realizadas con el identificador de personas, se utilizarán 16 y 32 componentes para determinar el número de componentes que mejor representa las muestras de audio de cada usuario. Adicionalmente, se realizan pruebas para observar el beneficio de añadir la variabilidad intra-locutor.

5.3.2.1. Prueba: P3-A y P3-B

Tabla 5.40: Tasas (FAR), (FRR) y (SAR) de los usuarios, utilizando 16 componentes sesión única y sesión múltiple para los usuarios 01_M, 02_F, 03_M, 04_M, 05_F.

Usuario	FAR	FRR	SAR	Usuario	FAR	FRR	SAR
01_M	13 %	2 %	63 %	01_M	24 %	1 %	71 %
02_F	5 %	0.5 %	94 %	02_F	16 %	0.05 %	95 %
03_M	7 %	1 %	75 %	03_M	14 %	0 %	83 %
04_M	7 %	0 %	89 %	04_M	12 %	0 %	94 %
05_F	11 %	0 %	97 %	05_F	14 %	0 %	99 %
Promedio	8.6 %	0.7 %	83.6 %	Promedio	16 %	0.21 %	88.4 %

5.3.2.2. Prueba: P4-A y P4-B

Tabla 5.41: Tasas (FAR), (FRR) y (SAR) de los usuarios, utilizando 32 componentes sesión única y sesión múltiple para los usuarios 01_M, 02_F, 03_M, 04_M, 05_F.

Usuario	FAR	FRR	SAR	Usuario	FAR	FRR	SAR
01_M	12 %	1 %	41 %	01_M	19 %	1 %	59 %
02_F	6 %	0 %	89 %	02_F	15 %	0 %	96 %
03_M	4 %	2 %	69 %	03_M	10 %	0.5 %	80 %
04_M	5 %	1 %	73 %	04_M	6 %	0.5 %	90 %
05_F	9 %	0 %	96 %	05_F	13 %	0 %	97 %
Promedio	7.2 %	0.8 %	73.6 %	Promedio	12.6 %	0.4 %	84.4 %

En las tablas 5.40 y 5.41 se encuentran las tablas correspondientes a las pruebas P3-A, P3-B, P4-A, P4B. Inicialmente, se puede notar, que de manera similar a las pruebas realizadas en la identificación de personas, al utilizar muestras de la primera y segunda sesión para entrenar a los modelos estadísticos, los resultados de la tasa de aceptación correcta aumentan para las diferentes pruebas realizadas (16 y 32 componentes).

Por otro lado, de la Tabla 5.40, se puede notar que utilizando 16 componentes y las muestras de la primera y segunda sesión, se obtienen los mejores resultados en la tasa de correcta aceptación (88,4%), se obtiene los mejores resultados con respecto a la tasa de falso rechazo(0.21%) y se obtienen los peores resultados en la tasa de falsa aceptación (16%).

5.3.3. Análisis de los resultados

Basándose en los resultados obtenidos en las pruebas realizadas sobre el identificador de personas y el verificador de personas mediante la voz (ver Secciones 5.3.1, 5.3.2). Se pueden realizar las siguientes observaciones:

- En las pruebas realizadas sobre el identificador de personas (ver Sección 5.3.1), se observó que la tasa de correcta aceptación (CIR), fue mayor cuando se entrenaron los modelos estadísticos (GMMs) con 16 componentes y utilizaron muestras de la primera y segunda sesión (ver Tabla 5.38). Adicionalmente, las pruebas realizadas sobre los usuarios 01_M y 03_M fueron las que menos tasa de correcta identificación obtuvieron.
- En prueba P1-B correspondiente a la Figura 5.26, se puede encontrar la matriz de confusión correspondiente a los resultados obtenidos en el cálculo de la tasa de correcta identificación para los modelos entrenados utilizando 16 componentes y muestras de la primera y segunda sesión. En esta Figura se puede observar que un porcentaje considerable de las muestras de las pruebas de los usuarios masculinos, 01_M, 03_M y 04_M, fueron clasificadas de manera incorrecta entre ellos. De manera similar, las muestras de prueba de los usuarios femeninos 02_F y 05_F, fueron clasificadas de manera errónea entre ellas. Esto se debe a las características inter-locutor que se corresponden con la variabilidad que existen entre las características particulares de la voz de cada usuario. A pesar de los casos de baja clasificación los resultados fueron aceptables.

- Debido a los errores de clasificación presentes en el identificador de personas, es necesario establecer un mecanismo de verificación. En las pruebas realizadas sobre el verificador de personas (ver Sección 5.3.2), se obtuvieron diferentes tipos de estimadores, que permiten determinar la calidad del verificador. De manera similar al identificador de personas, los mejores resultados se obtuvieron cuando los los modelos estadísticos (GMMs), fueron entrenados utilizando 16 componentes y muestras de la primera y segunda sesión. Esta configuración obtuvo una tasa de correcta identificación en promedio de 88.4 %, sin embargo, la tasa de falsa aceptación es mayor en comparación a las demás pruebas 16 %. No obstante, es importante resaltar que a diferencia de las demás pruebas, la tasa de falsa aceptación fue probada utilizando las muestras de todos los usuarios a excepción del usuario del modelo estadístico, es decir, se utilizaron muestras de otros usuarios para determinar la cantidad de veces en la cual accidentalmente se aceptaban cuando realmente no debían ser aceptados. Debido a esto, el número de muestras utilizadas para esta prueba (744 aproximadamente), es mayor con respecto a las demás pruebas.

Capítulo 6

Conclusiones y recomendaciones

En el presente trabajo de investigación se desarrollaron 3 productos principales: un Módulo Reconfigurable de Reconocimiento, un reconocedor de emociones y un reconocedor de personas mediante la voz. Este Módulo se diseñó con el objetivo de gestionar tareas de reconocimiento que manejan los robots sociales en la actualidad, permitiendo configurar en tiempo de ejecución qué tareas se ejecutan en el procesamiento interno y qué tareas se ejecutan en el procesamiento externo.

Para cada una de las tareas de reconocimiento involucradas en el caso de estudio, se realizó una investigación profunda para establecer criterios de diseño e implementación. Adicionalmente, se realizaron diferentes implementaciones y pruebas, que sirvieron para realizar comparaciones y evaluar compensaciones en ambas tareas.

Finalmente, la integración de estos 3 productos se llevó a cabo, permitiendo realizar una gestión completa de tareas y paquetes; y a su vez, ofreciendo a cualquier robot en particular, la capacidad de solicitar servicios de reconocimiento de forma local y remota.

6.1. Conclusiones

En este trabajo de investigación se desarrolló un Módulo Reconfigurable de Reconocimiento, con el objetivo de gestionar tareas y paquetes de manera local y remota, que además ofrece servicios de reconocimiento a un robot en particular. Este Módulo nace de la necesidad de mejorar las interacciones de los robots sociales actuales, permitiendo hacer un mejor uso de su procesamiento (interno y externo) mediante la configuración en tiempo de ejecución de sus tareas de reconocimiento. Al momento de implementar el Módulo Reconfigurable, se investigaron diferentes robots sociales actuales con la finalidad de encontrar las tecnológicas usadas en el área; tomando en consideración el diseño de los módulos que permiten el reconocimiento en estos robots. El framework ROS fue utilizado para realizar la implementación, no sólo por ser uno de los frameworks más utilizados actualmente para la programación de robots, sino además, porque permite la programación de aplicaciones robóticas sobre clusters heterogéneos de computadoras. Diferentes tipos de pruebas fueron realizadas para comprobar el funcionamiento del módulo, adicionalmente, se realizó la medición de memoria y procesamiento sobre uno de los gestores y una de las tareas de reconocimiento, obteniendo resultados acordes a lo esperado.

El reconocimiento de emociones es una de las principales tareas de reconocimiento encontradas en los robots sociales actuales, no obstante, para llevar a cabo esta tarea es necesario una base de datos para entrenar los algoritmos de aprendizaje de máquina. En esta investigación se llevó a cabo la construcción de una base de datos para el reconocimiento de emociones en español, para la cual se realizó una investigación de bases de datos de propósito similar, con el objetivo de establecer criterios propios de construcción y de diseño. Parte de su diseño involucró la realización de una encuesta, que permitió seleccionar las frases con las cuales las personas se sentían más a gusto al momento de expresarse según la emoción. El diseño y la construcción de la base de datos se realizó con la intención de dividirla en dos: una base de datos en condiciones controladas y actuadas, y una base de datos en condiciones controladas y semi-natural. Esto con el objetivo de realizar pruebas que permitan simular un ambiente más natural, y cada una de las muestras obtenidas fue sometida a un proceso de validación mediante un jurado. Finalmente, las muestras aprobadas fueron utilizadas como parte

del entrenamiento y prueba de diversos algoritmos de aprendizaje de máquina.

En esta investigación se probaron 3 algoritmos de aprendizaje de máquina para el reconocimiento de emociones mediante la voz: SVM, RF y GB; seleccionados según investigaciones previas en esta área. Adicionalmente, bajo la sugerencia de diversas investigaciones se probó un algoritmo basado en aprendizaje combinado, conformado por los modelos de los algoritmos seleccionados inicialmente. La selección del conjunto de características utilizado para representar las muestras de voz, se basó en investigaciones previas de esta área. Para cada uno de los modelos entrenados se realizaron diferentes pruebas para seleccionar el mejor y evaluar compensaciones entre los mismos. El algoritmo basado en aprendizaje combinado obtuvo los mejores resultados logrando una tasa de reconocimiento en promedio del 73 % en las pruebas de retención sobre la base de datos en condiciones controladas y semi-naturales. Finalmente, se realizó un análisis de los resultados para describir las particularidades de los resultados obtenidos.

Una de las tareas de reconocimiento fundamentales para la personalización de las interacciones en los robots sociales actuales, es la identificación. De manera similar al reconocimiento de emociones, para llevar a cabo la identificación de personas mediante la voz, es necesario la creación de una base de datos que permita capturar las características particulares de la voz de cada individuo de tal forma que pueda reconocerlo. En esta investigación se llevó a cabo la construcción de una base de datos orientada al reconocimiento independiente de texto de las personas mediante la voz. Para realizar esta base de datos se realizó una revisión de bases de datos de propósito similar, con la intención de establecer criterios propios de construcción y diseño. Parte de su diseño involucró la selección de un conjunto (frases y palabras) representativo del español venezolano. Esto se realizó con la intención de abarcar todos los fonemas del español y capturar las diferencias sutiles de las características inter-locutor. Otro conjunto de frases y palabras representativas del español fue seleccionado para la construcción de un conjunto de pruebas basándose en libros de fonética del español. Finalmente, la base de datos construida es multi-sesión, porque las muestras del conjunto de entrenamiento fueron grabadas dos veces en días distintos para capturar las características intra-locutor de los participantes.

Para realizar la implementación del identificador de personas se utilizó el algoritmo modelo de mezclas Gaussianas (GMM). Su selección fue basada en diversas investigaciones relacionadas con la identificación y verificación de personas independiente de texto. La selección del conjunto de características se basó en diversas investigaciones de propósito similar. Para realizar la identificación se debe entrenar un modelo por participante, no obstante, para optimizar la tasa de correcta identificación, se entrenaron varios modelos utilizando las muestras de una sola sesión y múltiples sesiones y variando el número de componentes del GMM. Los resultados obtenidos mostraron que utilizando 16 componentes y muestras de la primera y segunda sesión se obtuvieron los mejores resultados, alcanzando en promedio una tasa de reconocimiento del 87.6%. Adicionalmente, bajo sugerencia de diversas investigaciones se realizó la implementación de un verificador de personas, con la finalidad de establecer un filtro y verificar si se conoce o no la persona que habla.

Finalmente, se puede concluir que los objetivos planteados en el capítulo 1 fueron alcanzados tras haber realizado la integración del Módulo Reconfigurable de Reconocimiento con las tareas de reconocimiento involucradas en el caso de estudio.

6.2. Aportes

Las principales contribuciones de esta investigación son las siguientes:

- Se realizó una revisión profunda sobre las arquitecturas y tareas de reconocimiento de los robots sociales actuales.
- Se realizó una revisión profunda sobre algoritmos de aprendizaje de máquina orientados al reconocimiento de emociones mediante la voz.
- Se realizó una revisión profunda sobre algoritmos de aprendizaje de máquina orientados al reconocimiento de personas mediante la voz, de manera independiente al texto.
- Se realizó la construcción de una base datos para el reconocimiento de 6 emociones en español mediante la voz.

- Se realizó la construcción de una base de datos para el reconocimiento de personas mediante la voz, multi-sesión y orientada a la captura de características inter-locutor e intra-locutor.
- Se desarrollo un Módulo Reconfigurable de Reconocimiento mediante el framework ROS, que permite la configuración de nodos en tiempo de ejecución (gestión de tareas y paquetes), y ofrece servicios de reconocimiento a un robot en particular.

6.3. Recomendaciones

A continuación se presentan algunas recomendaciones sobre el Módulo Reconfigurable de Reconocimiento y las tareas involucradas en el caso de estudio.

- Los seres humanos pueden expresar emociones a distintos niveles de intensidad. De manera paraverbal, una misma emoción puede ser expresada de distintas maneras; es por esto que se debe tomar en cuenta que la emociones que se reconocen mediante la tarea del caso de estudio siguen ciertos patrones y no abarcan todo el espectro de cada una de las emociones.
- Una de las principales desventajas en el reconocimiento de patrones mediante el audio, es que éste es muy susceptible a las condiciones de ambiente. Para obtener los mejores resultados las condiciones entrenamiento y de pruebas deben ser lo más similares posibles.
- Por lo general, las condiciones de ambiente en las cuales interactúa un robot son bastante controladas. Las tareas de reconocimiento involucradas en el caso de estudio fueron implementadas tomando en consideración este hecho. Por lo que se recomienda su uso en condiciones similares (poco ruido y a una distancia de 30 centímetros del micrófono).
- Es recomendable añadir al conjunto de entrenamiento para el reconocedor de personas frases comunes y frecuentes del participante. Esto permite mejorar la captura de características inter-locutor del participante.

- El Módulo Reconfigurable de Reconocimiento fue diseñado para versión de ROS Kinectic Kame, por lo que no se garantiza su funcionamiento para cualquier otra versión de ROS. Se recomienda utilizar los sistema operativos soportados por esta versión: Ubuntu GNOME 16.04.5 LTS, Ubuntu Wily 15.10 y Debian 8.11 Jessie.
- En esta investigación se estilan ciertas responsabilidades a cada uno de los usuarios del Módulo Reconfigurable de Reconocimiento, no obstante, el framework ROS expone los servicios de cada uno de sus nodos activos, permitiendo hacer uso de ciertas funcionalidades a cualquier usuario. Se recomienda utilizar las funcionalidades basándose en las responsabilidades que se estilan de cada usuario.

6.4. Trabajos Futuros

A continuación se presentan trabajos futuros que surgieron a partir de este trabajo de investigación:

- Realizar un análisis exploratorio de las emociones en español, utilizando la base de datos construida en esta investigación.
- Realizar ingeniería de características para el reconocimiento de emociones en español, utilizando la base de datos construida en esta investigación.
- Realizar un análisis cualitativo de técnicas para el reconocimiento de personas de manera independiente al texto, utilizando la base de datos para el reconocimiento de personas mediante la voz.
- Desarrollar un sistema detector de voz (VAD, por sus siglas en ingles) orientado a la interacción humano-robot.
- Desarrollar de un Módulo autónomo para la gestión de tareas y paquetes de reconocimiento orientado a la interacción humano-robot.

Anexo A

Documento de Especificación de Requisitos

A.1. Introducción

La finalidad de esta sección es proveer una descripción y revisión de todos los requisitos del Módulo Reconfigurable de Reconocimiento para la interacción humano-robot. La estructura de este documento es realizada mediante la especificación de requisitos software ANSI/IEEE 830 [1].

A.1.1. Objetivo

El objetivo de este documento es especificar y definir de manera clara y precisa las funcionalidades y restricciones que tendrá el Módulo Reconfigurable de Reconocimiento para la interacción humano-robot. Éste, ilustrará el propósito y una declaración completa para el desarrollo del módulo.

A.1.2. Alcance

El Módulo Reconfigurable de Reconocimiento es un subcomponente orientado a la interacción humano-robot, específicamente donde participan robots sociales. Este subcomponente, permite la gestión de tareas de reconocimiento de forma local y remota,

permitiendo cambiar la configuración del robot en tiempo de ejecución. El robot puede delegar las tareas de reconocimiento en el gestor local o en el gestor remoto según el escenario de interacción o cualquier otro factor.

Como caso de estudio, el Módulo Reconfigurable de Reconocimiento, cuenta con dos tareas: reconocimiento de emociones y de personas mediante la voz. No obstante, este módulo permite instalar múltiples tareas de reconocimiento.

Para realizar las pruebas sobre el módulo se simulará un módulo de emisión y recepción que permitirá enviar solicitudes de reconocimiento y obtener las respuestas de las mismas. Además, este módulo utilizará las operaciones disponibles para el robot.

El gestor local y el gestor remoto se encuentran en dos computadoras distintas.

A.1.3. Definiciones, acrónimos y abreviaciones

- **TCP:** Protocolo de Transmisión de Control (TCP, por sus siglas en inglés).
- **Tarea de reconocimiento:** Elemento del sistema de software que ofrece servicios de reconocimiento.
- **Paquete de reconocimiento:** Paquete de datos que contiene una tarea de reconocimiento.

A.1.4. Referencias

- IEEE Template for System Requirement Specification Documents [1].
- UML [24]
- Wiki Ubuntu GNOME 16.04.5 LTS [37]
- Jupyter Notebook [39]

A.1.5. Visión general

El resto de este capítulo lo conforman 3 secciones. La segunda sección contiene una descripción de la funcionalidad del módulo de reconocimiento, la interacción con

otros submódulos y usuarios. Finalmente, la tercera sección contiene la especificación de requisitos en términos detallados y una descripción de la diferentes interfaces del módulo.

A.2. Descripción general

Esta sección contiene una visión general de todo el Módulo Reconfigurable de Reconocimiento. El módulo será explicado en su contexto para mostrar cómo interactúa con otros submódulos e introducir la funcionalidad básica del mismo. También se describirá cuáles son los usuarios que utilizarán el módulo y qué funcionalidad está disponible para cada uno de ellos. Finalmente, se presentarán las restricciones y suposiciones para el módulo.

A.2.1. Perspectiva del producto

El módulo consiste de dos partes principales: un gestor local (procesamiento interno) y un gestor remoto (procesamiento externo). El gestor local se encarga de gestionar las tareas y paquetes de reconocimiento de forma local, de tal manera que permite al robot solicitar servicios de configuración y reconocimiento en su procesamiento interno. De manera similar, el gestor remoto gestiona las tareas y paquetes de reconocimiento de forma remota, permitiendo al robot solicitar servicios como la instalación de paquetes remotos en su gestor local y la solicitud de servicios de reconocimiento locales y remotos.

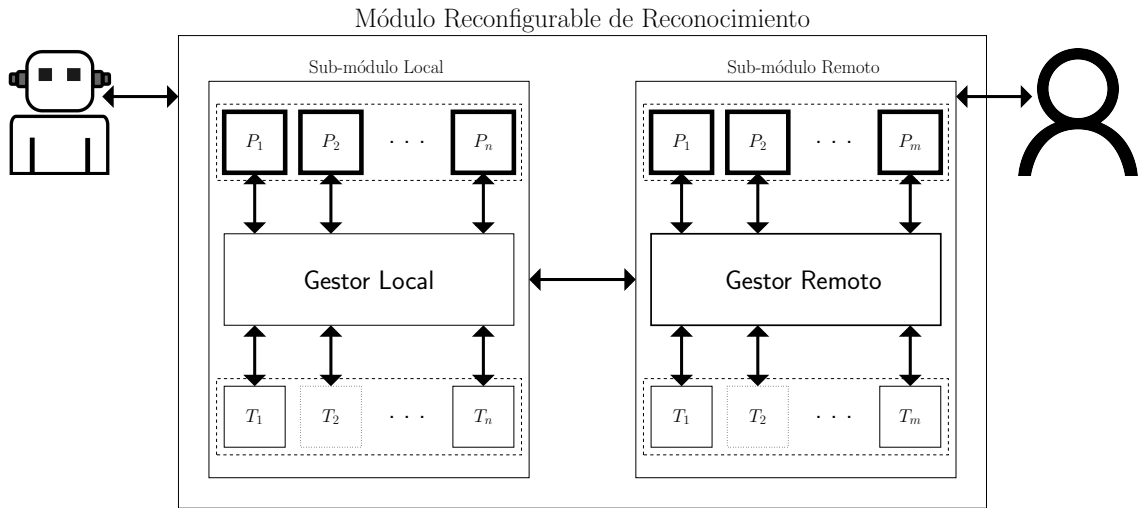


Figura A.1: Diagrama de bloques del Módulo Reconfigurable de Reconocimiento

El sub-módulo local es parte del procesamiento interno del robot y está formado por un gestor local y n tareas de reconocimiento, y el Sub-módulo remoto se ejecuta en el procesamiento externo del robot y está formado por un gestor remoto y m tareas de reconocimiento. De manera que, las tareas del gestor local son un subconjunto de las tareas del gestor remoto ($n \leq m$). El atributo reconfigurable de la arquitectura se refleja en la posibilidad de cambiar la configuración del robot en tiempo de ejecución.

A.2.2. Funciones del producto

Utilizando el Módulo Reconfigurable de Reconocimiento, el robot podrá realizar solicitudes de gestión de tareas y paquetes al gestor local, y solicitudes de reconocimiento locales o remotas. Por otro lado, el gestor remoto permitirá al operador gestionar las tareas y paquetes de reconocimiento disponibles para el robot.

Las funciones del producto se dividen en 3 áreas principales: gestor local, gestor remoto y servicios de reconocimiento.

- **Gestor Local:** Se encarga de responder las solicitudes realizadas del robot, en general ofrece dos grandes servicios:
 1. **Gestión de tareas de reconocimiento local:** Se encarga de ofrecer servicios para el cambio de estado (activación y desactivación) de las tareas

de reconocimiento disponibles en el gestor local. El gestor local provee una lista de todas las tareas de reconocimiento instaladas junto con información sobre su estado, permitiendo al robot cambiar el estado de cualquier tarea en particular.

2. **Gestión de paquetes de reconocimiento local:** Se encarga de ofrecer servicios para la instalación y eliminación de paquetes en el gestor local. El gestor remoto provee una lista de todas las tareas de reconocimiento disponibles para el robot, permitiéndole solicitar la instalación de un paquete específico desde el gestor remoto al gestor local. Por otro lado, el robot puede eliminar un paquete de reconocimiento del gestor local utilizando como referencia la lista de tareas que se han instalado.

- **Gestor remoto:** Se encarga de responder las solicitudes del gestor local y del operador. En general ofrece 2 grandes servicios:

1. **Gestión de tareas de reconocimiento remoto:** Se encarga de ofrecer servicios para el cambio de estado (activación y desactivación) de las tareas de reconocimiento disponibles en el gestor remoto. El gestor remoto provee una lista de todas las tareas de reconocimiento instaladas junto con información de su estado, permitiendo al operador cambiar el estado de cualquier tarea en particular y por ende, extendiendo o reduciendo las capacidades de reconocimiento del robot.
2. **Gestión de paquetes de reconocimiento remoto:** Se encarga de ofrecer servicios para la instalación y eliminación de paquetes en el gestor remoto. El gestor remoto ofrece servicios para que el operador pueda instalar paquetes de reconocimiento nuevos y además pueda eliminar un paquete de reconocimiento utilizando como referencia la lista de tareas que se han instalado. Otro servicio incluido en la gestión de paquetes, es proveer la dirección de un paquete en específico, como parte del mecanismo de instalación de paquetes en el gestor local.

- **Servicios de reconocimiento:** Se encarga de ofrecer al robot la posibilidad de

realizar solicitudes de reconocimiento locales o remotas.

1. **Solicitud de reconocimiento local:** Se encarga de proveer al robot con la capacidad de realizar solicitudes de reconocimiento local. El robot, utiliza la lista de tareas de reconocimiento disponible del gestor local para solicitar un servicio específico de reconocimiento local.
2. **Solicitud de reconocimiento remoto:** Se encarga de proveer al robot con la capacidad de realizar solicitudes de reconocimiento remoto. El robot, utiliza la lista de tareas de reconocimiento disponible del gestor remoto para solicitar un servicio específico de reconocimiento remoto.

A.2.3. Características de los usuarios

Existen dos tipos de usuarios finales que interactúan con el módulo reconfigurable: el robot y el operador. Cada uno de éstos utiliza el módulo de manera distinta por lo que cada uno tiene sus propios requerimientos.

El robot utiliza el gestor local y el gestor remoto para gestionar las tareas y paquetes de reconocimiento locales y remotos. Además solicita servicios de reconocimiento local y remoto.

El investigador no interactuá con el gestor local, en su lugar, utiliza el gestor remoto para gestionar las tareas y paquetes de reconocimiento.

A.2.4. Limitaciones generales

- El gestor local y el gestor remoto están limitados por sus recursos de hardware, como la memoria y procesamiento. Dado que existen diferentes configuraciones de hardware disponibles para robots, la cantidad de componentes de reconocimiento y la capacidad de procesamiento pueden variar.
- La conexión de la red es una limitante para el robot, ya que el tiempo de respuesta de las solicitudes al gestor remoto se ve afectado por la latencia de la red.

A.2.5. Supuestos y dependencias

- El robot tiene acceso a la red del gestor remoto.
- La conexión entre los gestores siempre se mantiene.
- Los paquetes apropiados son instalados en ambos gestores.

A.3. Requisitos específicos

Esta sección contiene todos los requisitos funcionales y de calidad del módulo de reconocimiento. Da detalles de descripción del módulo y todas sus características.

A.3.1. Interfaces de usuario

- **Interfaz de comunicación con el operador:** El operador utiliza el terminal de comandos, para conectarse mediante shell seguro (SSH, por sus siglas en inglés) al gestor remoto y de esta forma utilizar los servicios del gestor remoto para gestionar las tareas y paquetes de reconocimiento.
- **Interfaz de comunicación con el robot:** El robot utiliza las interfaces de programación de aplicaciones (APIs, por sus siglas en inglés) de ROS, para comunicarse con todos los componentes del Módulo Reconfigurable de Reconocimiento.

A.3.2. Interfaces de hardware

- El operador requiere una computadora que tenga conexión con el gestor remoto. Esa conexión se proporciona mediante una conexión remota (SSH).

A.3.3. Interfaces de software

1. Sistema operativo Ubuntu Xenial Xerus.
 - **Mnemotécnico:** Ubuntu Xenial Xerus

- **Número de versión:** 16.04 LTS
- **Fuente:** <http://releases.ubuntu.com/16.04>
- **Propósito:** Compatibilidad con el sistema operativo ROS.

2. Sistema Operativo Robótico.

- **Mnemotécnico:** ROS
- **Versión:** Kinetic Kame
- **Fuente:** <http://wiki.ros.org/kinetic>
- **Propósito:** Implementación de la arquitectura del módulo reconfigurable.

A.3.4. Interfaces de comunicación

- El robot se comunica con los gestores interno y externo mediante el protocolo TCP/IP.
- El operador utiliza SSH, por lo que el protocolo utilizado es TCP/IP.

A.3.5. Características del módulo

A.3.5.1. Gestión de tareas de reconocimiento local

La gestión de tareas de reconocimiento local comprende todas las actividades que se encargan de activar o desactivar las tareas de reconocimiento en el procesamiento interno del robot. Para realizar cualquier acción de activación o desactivación es necesario conocer la lista de tareas disponibles en el procesamiento interno.

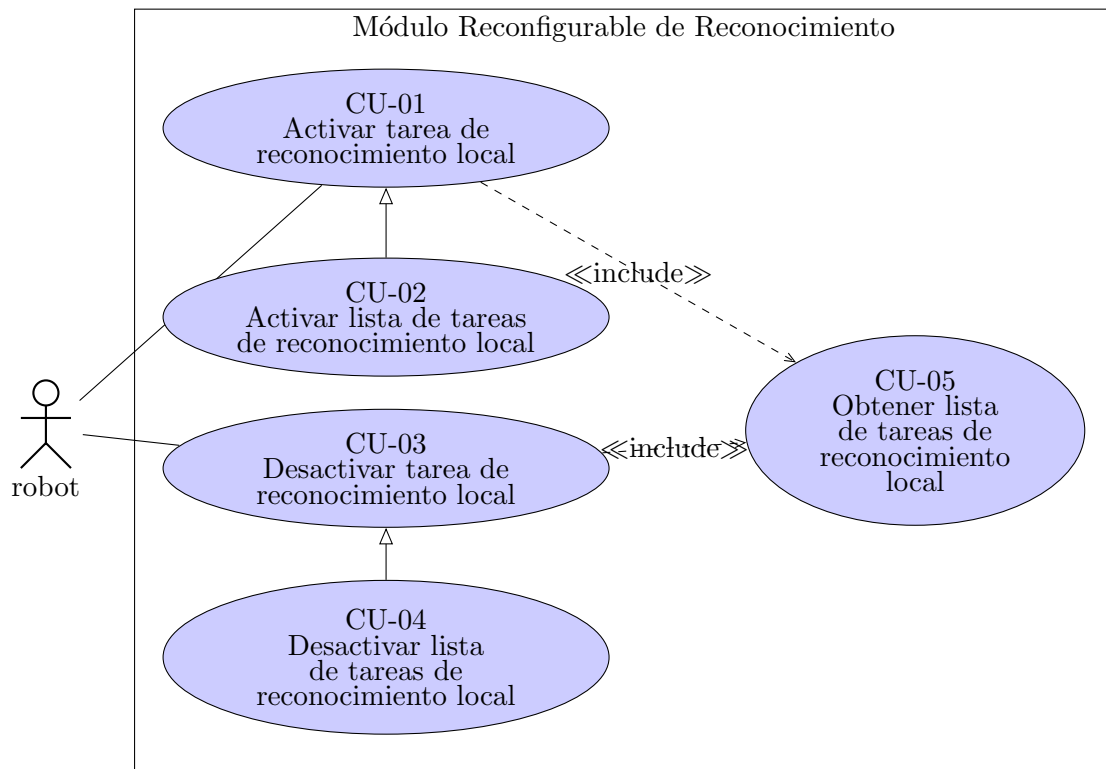


Figura A.2: Diagrama de caso de uso: Gestión de tareas de reconocimiento local

A.3.5.1.1. CU-01: Activar tarea de reconocimiento local

Tabla A.1: Descripción del Caso de Uso CU-01

Caso de Uso:	01
Nombre:	Activar tarea de reconocimiento local
Evento disparador:	El robot requiere activar una tarea de reconocimiento.
Actores:	Robot
Dependencias:	CU-05: Obtener lista de tareas de reconocimiento local
Descripción:	Permite al robot solicitar la activación de una tarea de reconocimiento local, para ser usada posteriormente.
Pre-condiciones:	<ol style="list-style-type: none"> 1. El robot tiene la lista de las tareas disponibles de reconocimiento local. 2. Existe por lo menos un paquete de reconocimiento instalado en el gestor local.
Post-condiciones:	<ol style="list-style-type: none"> 1. La tarea específica de reconocimiento está activada y disponible para ser usada.
Flujo Normal:	<ol style="list-style-type: none"> 1. El robot selecciona una tarea específica para su activación, utilizando como referencia la lista de tareas disponibles en el gestor local. 2. El robot solicita al gestor local la activación de una tarea de reconocimiento específica. 3. El gestor local activa la tarea de reconocimiento. 4. Se actualiza el estado de la tarea de reconocimiento.
Flujo Alternativo:	<ol style="list-style-type: none"> 2.A La tarea de reconocimiento solicitada ya se encuentra activada. El caso de uso termina.

A.3.5.1.2. CU-02: Activar lista de tareas de reconocimiento local

Tabla A.2: Descripción del Caso de Uso CU-02

Caso de Uso:	02
Nombre:	Activar lista de tareas de reconocimiento local
Evento disparador:	El robot requiere activar una lista tarea de reconocimiento.
Actores:	Robot
Dependencias:	CU-05: Obtener lista de tareas de reconocimiento local
Descripción:	Permite al robot solicitar la activación de una lista de tareas de reconocimiento local, para ser usadas posteriormente.
Pre-condiciones:	<ol style="list-style-type: none"> 1. El robot tiene la lista de las tareas disponibles de reconocimiento local. 2. Existe por lo menos un paquete de reconocimiento instalado en el gestor local.
Post-condiciones:	<ol style="list-style-type: none"> 1. La taras específicas de reconocimiento están activadas y disponibles para ser usadas.
Flujo Normal:	<ol style="list-style-type: none"> 1. El robot selecciona un conjunto de tareas específicas para su activación, utilizando como referencia la lista de tareas disponibles en el gestor local. 2. El robot solicita al gestor local la activación de una lista de tareas de reconocimiento. 3. El gestor local activa las tareas de reconocimiento. 4. Se actualiza el estado de las tareas de reconocimiento.
Flujo Alternativo:	<ol style="list-style-type: none"> 2.A Una o más tareas de reconocimiento ya se encuentran activadas. Se activan solo aquellas que se encuentran desactivadas.

A.3.5.1.3. CU-03: Desactivar tarea de reconocimiento local

Tabla A.3: Descripción del Caso de Uso CU-03

Caso de Uso:	03
Nombre:	Desactivar tarea de reconocimiento local
Evento disparador:	El robot requiere desactivar una tarea de reconocimiento.
Actores:	Robot
Dependencias:	CU-05: Obtener lista de tareas de reconocimiento local
Descripción:	Permite al robot solicitar la desactivación de una tarea de reconocimiento local.
Pre-condiciones:	<ol style="list-style-type: none"> 1. El robot tiene la lista de las tareas disponibles de reconocimiento local. 2. Existe por lo menos un paquete de reconocimiento instalado en el gestor local.
Post-condiciones:	<ol style="list-style-type: none"> 1. La tarea específica de reconocimiento está desactivada.
Flujo Normal:	<ol style="list-style-type: none"> 1. El robot selecciona una tarea específica para su desactivación, utilizando como referencia la lista de tareas disponibles en el gestor local. 2. El robot solicita al gestor local la desactivación de una tarea de reconocimiento específica. 3. El gestor local desactiva la tarea de reconocimiento. 4. Se actualiza el estado de la tarea de reconocimiento.
Flujo Alternativo:	<ol style="list-style-type: none"> 2.A La tarea de reconocimiento solicitada ya se encuentra desactivada. El caso de uso termina.

A.3.5.1.4. CU-04: Desactivar lista tareas de reconocimiento local

Tabla A.4: Descripción del Caso de Uso CU-04

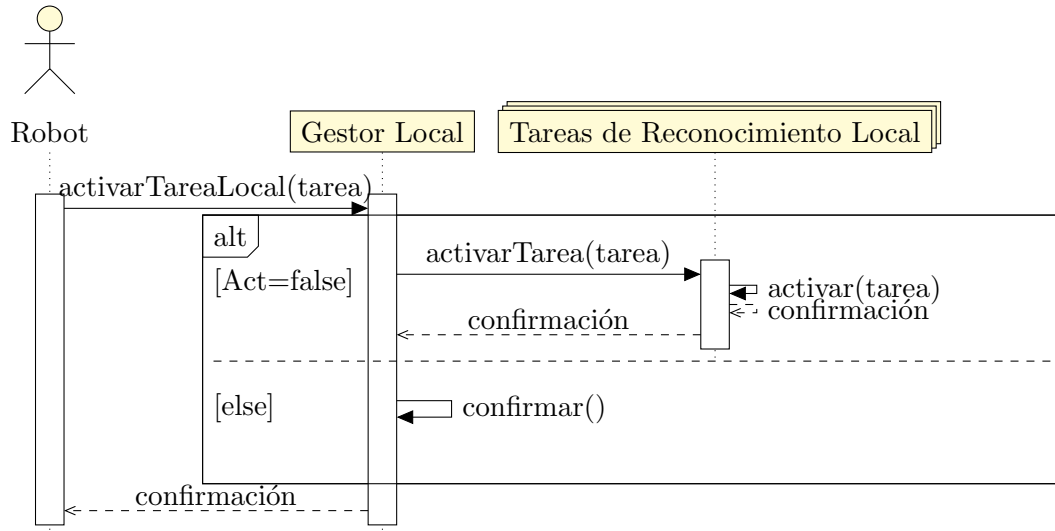
Caso de Uso:	04
Nombre:	Desactivar lista de tareas de reconocimiento local
Evento disparador:	El robot requiere desactivar una lista de tarea de reconocimiento.
Actores:	Robot
Dependencias:	CU-05: Obtener lista de tareas de reconocimiento local
Descripción:	Permite al robot solicitar la desactivación de una lista de tareas de reconocimiento local.
Pre-condiciones:	<ol style="list-style-type: none"> 1. El robot tiene la lista de las tareas disponibles de reconocimiento local. 2. Existe por lo menos un paquete de reconocimiento instalado en el gestor local.
Post-condiciones:	<ol style="list-style-type: none"> 1. La tareas de reconocimiento están desactivadas.
Flujo Normal:	<ol style="list-style-type: none"> 1. El robot selecciona un conjunto de tareas específicas para su desactivación, utilizando como referencia la lista de tareas disponibles en el gestor local. 2. El robot solicita al gestor local la desactivación de una lista de tareas de reconocimiento. 3. El gestor local desactiva las tareas de reconocimiento. 4. Se actualiza el estado de las tareas de reconocimiento.
Flujo Alternativo:	<ol style="list-style-type: none"> 2.A Una o más tareas de reconocimiento ya se encuentran desactivadas. Se desactivan solo aquellas que se encuentran activadas.

A.3.5.1.5. CU-05: Obtener lista de tareas de reconocimiento local

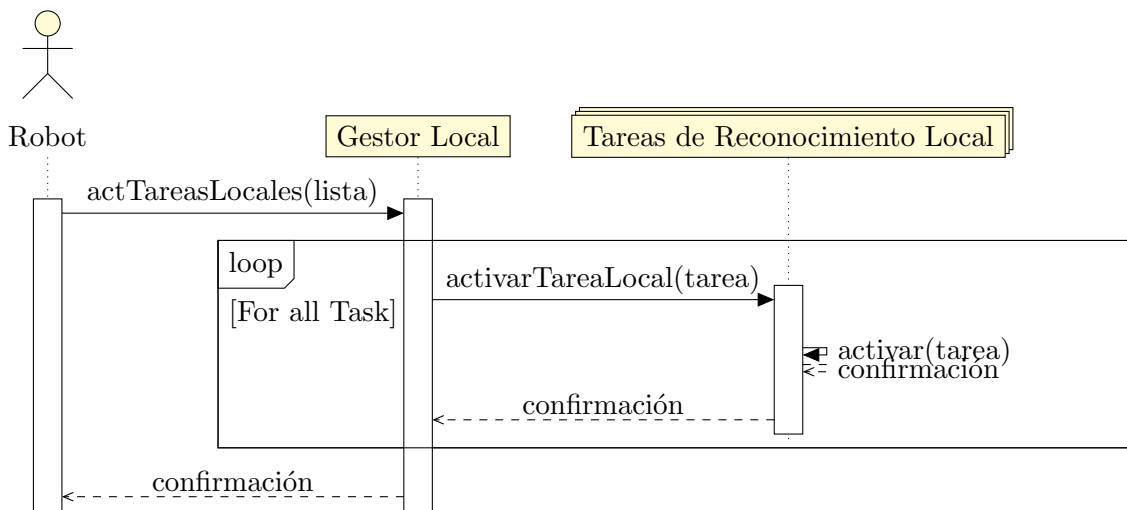
Tabla A.5: Descripción del Caso de Uso CU-05

Caso de Uso:	05
Nombre:	Obtener lista de tareas de reconocimiento local
Evento disparador:	El robot requiere conocer la lista de tareas de reconocimiento local.
Actores:	Robot
Dependencias:	CU-01: Activar tarea de reconocimiento local , CU-03: Desactivar tarea de reconocimiento local.
Descripción:	Permite al robot obtener la lista de tareas de reconocimiento disponibles en el gestor local.
Pre-condiciones:	
Post-condiciones:	1. El robot obtiene una lista de todas las tareas de reconocimiento disponibles en el gestor local con información de sus estados.
Flujo Normal:	<ol style="list-style-type: none"> 1. El robot solicita al gestor local la lista de tareas de reconocimiento. 2. El gestor local retorna una lista de todas las tareas de reconocimiento disponibles junto con su estado (activado/desactivado). 3. El robot obtiene la lista de tareas de reconocimiento local.
Flujo Alternativo:	2.A El gestor local no tiene ninguna tarea de reconocimiento disponible. Se retorna una lista vacía y el caso de uso termina.

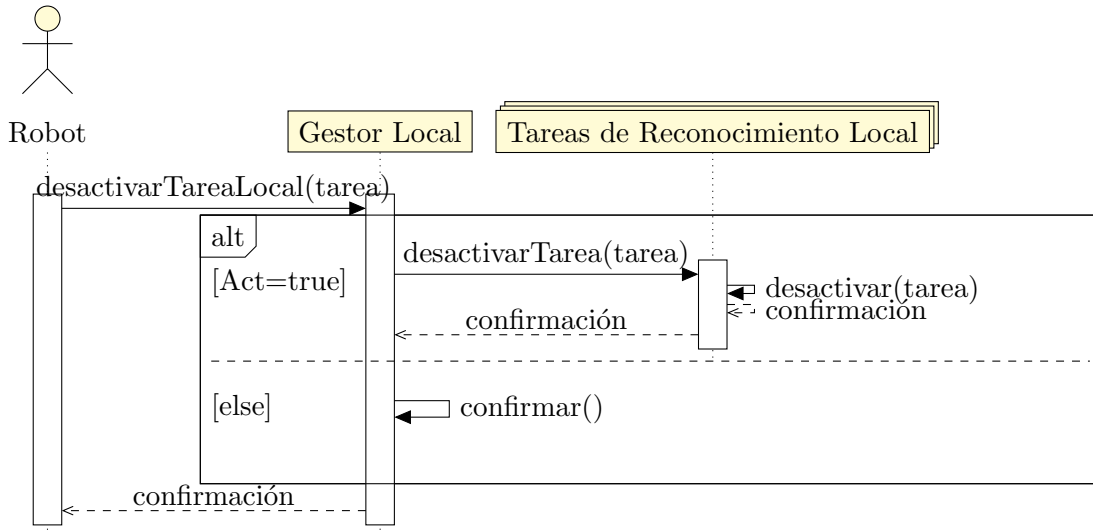
A.3.5.1.6. Diagrama de secuencia CU-01: Activar tarea de reconocimiento local



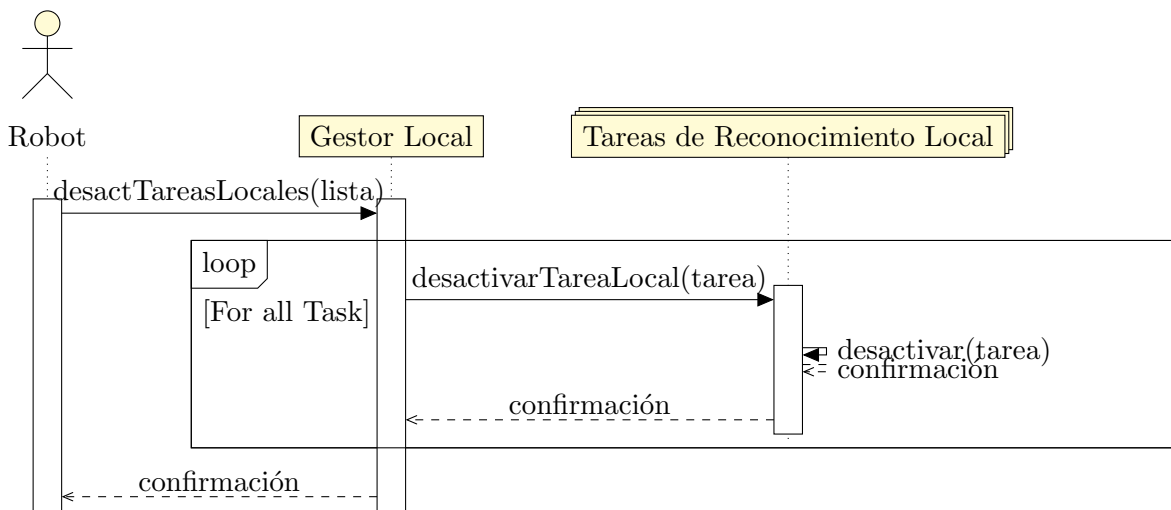
A.3.5.1.7. Diagrama de secuencia CU-02: Activar lista de tareas de reconocimiento local.



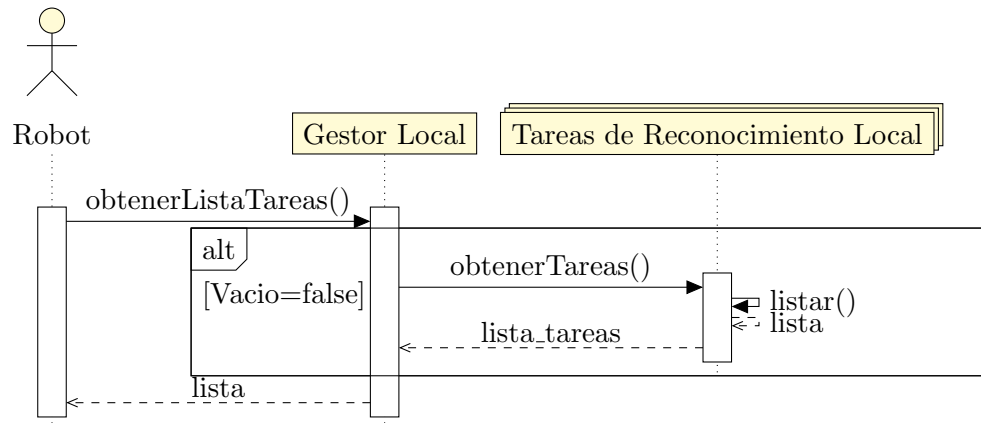
A.3.5.1.8. Diagrama de secuencia CU-03: Desactivar tarea de reconocimiento local



A.3.5.1.9. Diagrama de secuencia CU-04: Desactivar lista de tareas de reconocimiento local.



A.3.5.1.10. Diagrama de secuencia CU-05: Obtener lista de tareas de reconocimiento local



A.3.5.2. Gestión de paquetes de reconocimiento local

La gestión de paquetes de reconocimiento local comprende todas las actividades que se encargan de la instalación y eliminación de paquetes en el procesamiento interno del robot. Para realizar la eliminación de paquetes en el gestor local es necesario conocer la lista de tareas de reconocimiento disponibles en el procesamiento interno, por otro lado, en el caso particular de la instalación de paquetes en el gestor local, es necesario conocer la lista de tareas de reconocimiento disponible en el gestor remoto, de esta manera es posible solicitar las direcciones de los paquetes necesarios para su posterior instalación.

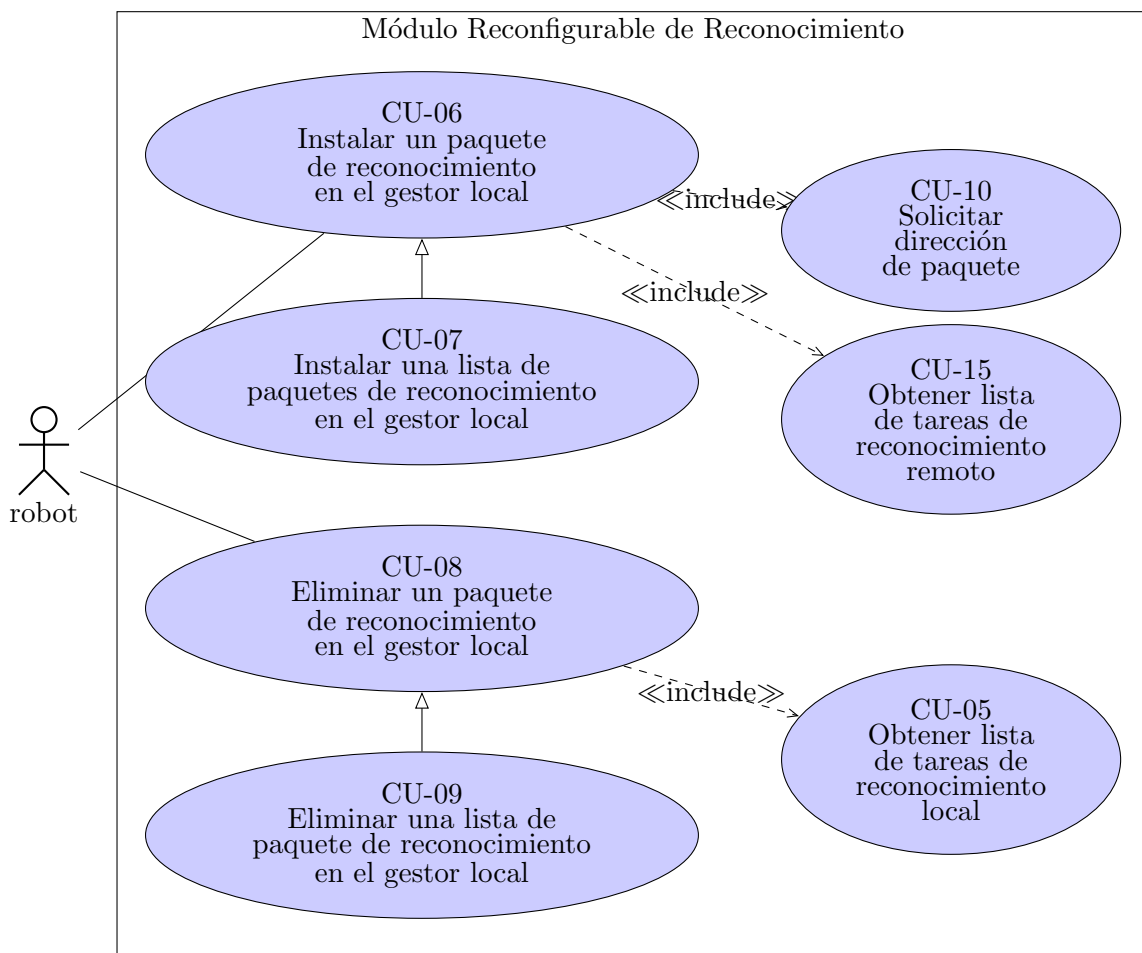


Figura A.3: Diagrama de caso de uso: Gestión de paquetes de reconocimiento local

A.3.5.2.1. CU-06: Instalar un paquete de reconocimiento en el gestor local

Tabla A.6: Descripción del Caso de Uso CU-06

Caso de Uso:	06
Nombre:	Instalar un paquete de reconocimiento en el gestor local
Evento disparador:	El robot requiere utilizar una tarea de reconocimiento remota de manera local.
Actores:	Robot
Dependencias:	CU-10: Solicitar dirección de paquete, CU-15: Obtener lista de tareas de reconocimiento remoto.
Descripción:	Permite al robot la instalación de un paquete de reconocimiento remoto en su procesamiento local.
Pre- condiciones:	<ol style="list-style-type: none"> 1. El robot tiene la lista de tareas disponibles de reconocimiento remoto. 2. Existe por lo menos un paquete de reconocimiento instalado en el gestor remoto.
Post- condiciones:	<ol style="list-style-type: none"> 1. El robot instala un paquete de reconocimiento específico en su procesamiento local.
Flujo Normal:	<ol style="list-style-type: none"> 1. El robot solicita al gestor local la instalación de un nuevo paquete de reconocimiento utilizando como referencia la lista de tareas de reconocimiento remoto. 2. El gestor local solicita al gestor remoto la dirección del paquete específico de reconocimiento. 3. El gestor local utiliza la dirección del paquete para descargarlo e instalarlo. 4. Se actualiza la lista de tareas de reconocimiento del gestor local.
Flujo Alternativo:	<ol style="list-style-type: none"> 1.A El paquete solicitado por el robot, ya se encuentra instalado. Se termina el caso de uso.

A.3.5.2.2. CU-07: Instalar una lista de paquetes de reconocimiento en el gestor local

Tabla A.7: Descripción del Caso de Uso CU-07

Caso de Uso:	07
Nombre:	Instalar una lista de paquetes de reconocimiento en el gestor local
Evento disparador:	El robot requiere disponer de una lista de tareas de reconocimiento remoto en su procesamiento local.
Actores:	Robot
Dependencias:	CU-10: Solicitar dirección de paquete, CU-15: Obtener lista de tareas de reconocimiento remoto.
Descripción:	Permite al robot la instalación de una lista de paquetes de reconocimiento remoto en su procesamiento local.
Pre-condiciones:	<ol style="list-style-type: none"> 1. Existe por lo menos un paquete de reconocimiento en el gestor remoto. 2. El robot tiene la lista de tareas disponibles de reconocimiento remoto.
Post-condiciones:	<ol style="list-style-type: none"> 1. El robot instala una lista de paquetes de reconocimiento remoto en su procesamiento local.
Flujo Normal:	<ol style="list-style-type: none"> 1. El robot solicita al gestor local la instalación de una lista de paquetes de reconocimiento utilizando como referencia la lista de tareas de reconocimiento remoto. 2. El gestor local solicita al gestor remoto las direcciones de los paquetes de reconocimiento solicitados. 3. El gestor local utiliza la direcciones de los paquetes para descargarlos e instalarlos. 4. Se actualiza la lista de tareas de reconocimiento del gestor local.
Flujo Alternativo:	<ol style="list-style-type: none"> 1.A Uno o más paquetes de reconocimiento ya se encuentran en el gestor local. Se instalan solo aquellos que no estén en el gestor local.

A.3.5.2.3. CU-08: Eliminar un paquete de reconocimiento en el gestor local

Tabla A.8: Descripción del Caso de Uso CU-08

Caso de Uso:	08
Nombre:	Eliminar un paquete de reconocimiento en el gestor local
Evento disparador:	El robot requiere eliminar paquete de reconocimiento de su gestor local.
Actores:	Robot
Dependencias:	CU-05: Obtener lista de tareas de reconocimiento local
Descripción:	Permite al robot eliminar un paquete de reconocimiento local y ende por la respectiva tarea de reconocimiento.
Pre-condiciones:	<ol style="list-style-type: none"> 1. El robot tiene la lista de tareas de reconocimiento disponibles en el gestor local. 2. El gestor local tiene instalado por lo menos un paquete de reconocimiento.
Post-condiciones:	<ol style="list-style-type: none"> 1. El robot no posee el paquete específico de reconocimiento en el gestor local.
Flujo Normal:	<ol style="list-style-type: none"> 1. El robot solicita la eliminación de un paquete de reconocimiento específico utilizando como referencia la lista de tareas del gestor local. 2. El gestor local ubica el paquete y lo elimina. 3. Se actualiza la lista de tareas de reconocimiento del gestor local.
Flujo Alternativo:	<ol style="list-style-type: none"> 2.A Si la tarea asociada al paquete a eliminar se encuentra activa, se desactiva la tarea y se elimina el paquete. Se continua con 3.

A.3.5.2.4. CU-09: Eliminar una lista de paquetes de reconocimiento en el gestor local

Tabla A.9: Descripción del Caso de Uso CU-09

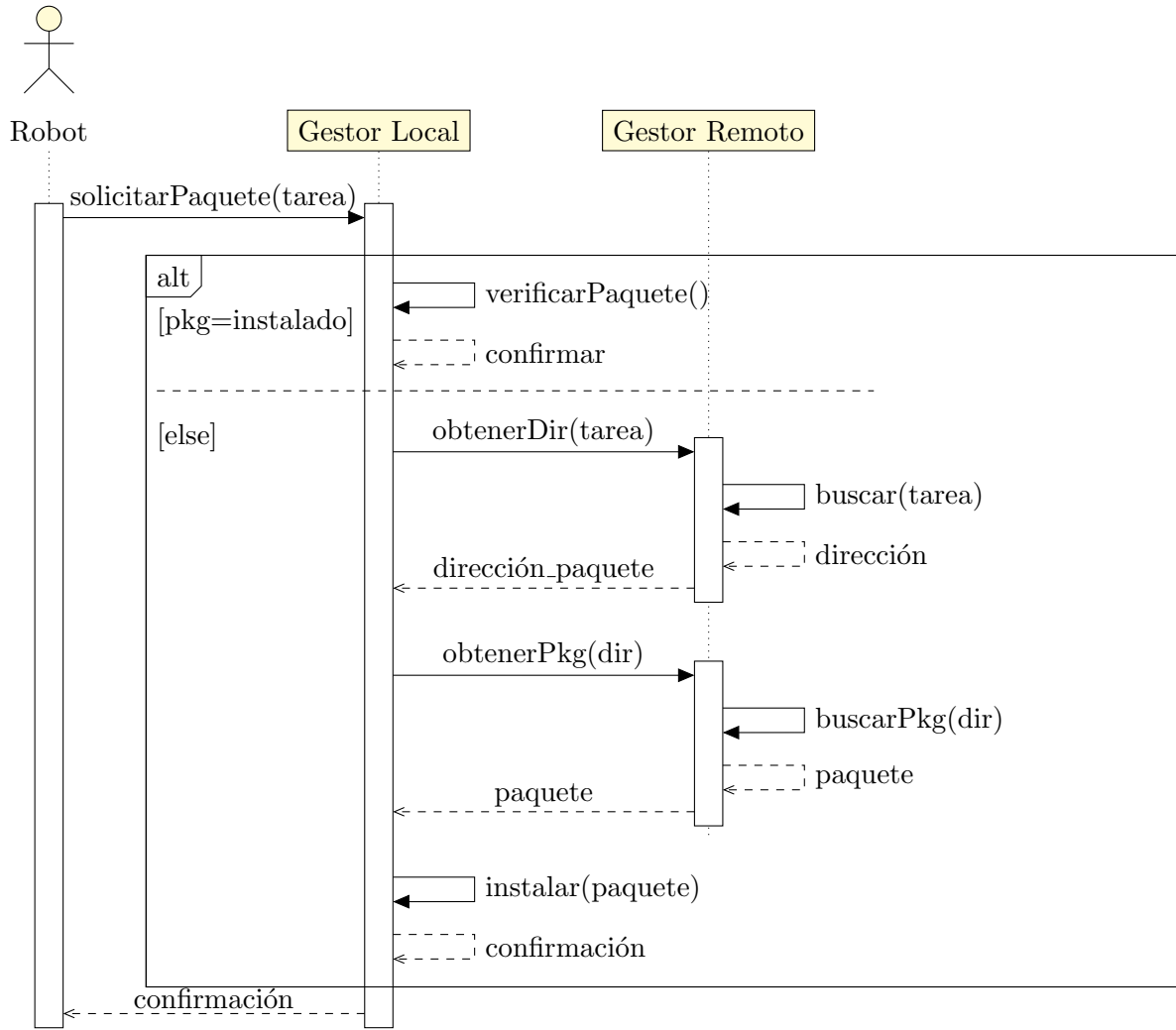
Caso de Uso:	09
Nombre:	Eliminar una lista de paquetes de reconocimiento en el gestor local
Evento disparador:	El robot requiere eliminar una lista de paquetes de reconocimiento de su procesamiento local.
Actores:	Robot
Dependencias:	CU-05: Obtener lista de tareas de reconocimiento local
Descripción:	Permite al robot eliminar una lista de paquetes de reconocimiento local y ende las respectivas tareas de reconocimiento.
Pre-condiciones:	<ol style="list-style-type: none"> 1. El robot tiene la lista de tareas de reconocimiento disponibles en el gestor local. 2. El gestor local tiene instalado por lo menos un paquete de reconocimiento.
Post-condiciones:	<ol style="list-style-type: none"> 1. El robot no posee los paquetes de reconocimiento pertenecientes a la lista en el gestor local.
Flujo Normal:	<ol style="list-style-type: none"> 1. El robot solicita la eliminación de una lista de paquetes de reconocimiento en el gestor local, utilizando como referencia la lista de tareas del gestor local. 2. El gestor local ubica los paquetes y los elimina. 3. Se actualiza la lista de tareas de reconocimiento del gestor local.
Flujo Alternativo:	<ol style="list-style-type: none"> 2.A Si alguna de las tareas asociadas a los paquetes a eliminar se encuentra activa, se desactivan las tareas y se eliminan los respectivos paquetes. Se continua con 3.

A.3.5.2.5. CU-10: Solicitar la dirección de un paquete

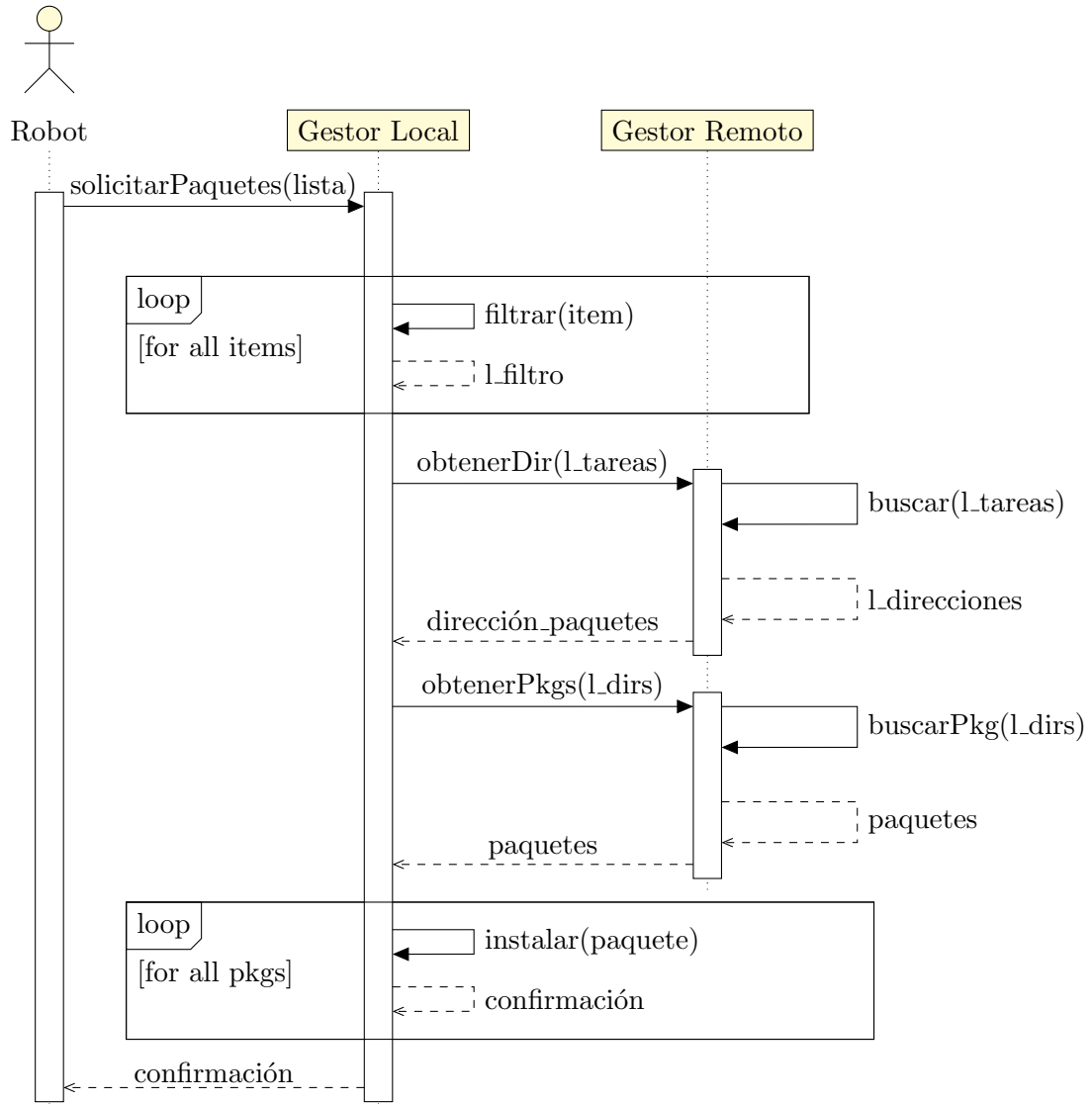
Tabla A.10: Descripción del Caso de Uso CU-10

Caso de Uso:	10
Nombre:	Solicitar la dirección de un paquete
Evento disparador:	El robot requiere la instalación de un paquete de reconocimiento del gestor remoto en el gestor local, para instalar este paquete, el gestor local necesita la dirección del paquete.
Actores:	Robot
Dependencias:	CU-06: Instalar un paquete de reconocimiento en el gestor local
Descripción:	Permite al gestor local solicitar direcciones de paquetes de reconocimiento del gestor remoto.
Pre-condiciones:	1. El gestor remoto tiene instalado por lo menos un paquete de reconocimiento.
Post-condiciones:	1. El gestor local obtiene la dirección de paquete de reconocimiento específico del gestor remoto.
Flujo Normal:	<ol style="list-style-type: none"> 1. El robot, mediante el gestor local, solicita al gestor remoto la dirección de un paquete específico de reconocimiento, utilizando como referencia la lista de tareas de reconocimiento remoto. 2. El gestor remoto retorna la dirección del paquete al gestor local. 3. El gestor local obtiene la dirección del paquete de reconocimiento específico.
Flujo Alternativo:	

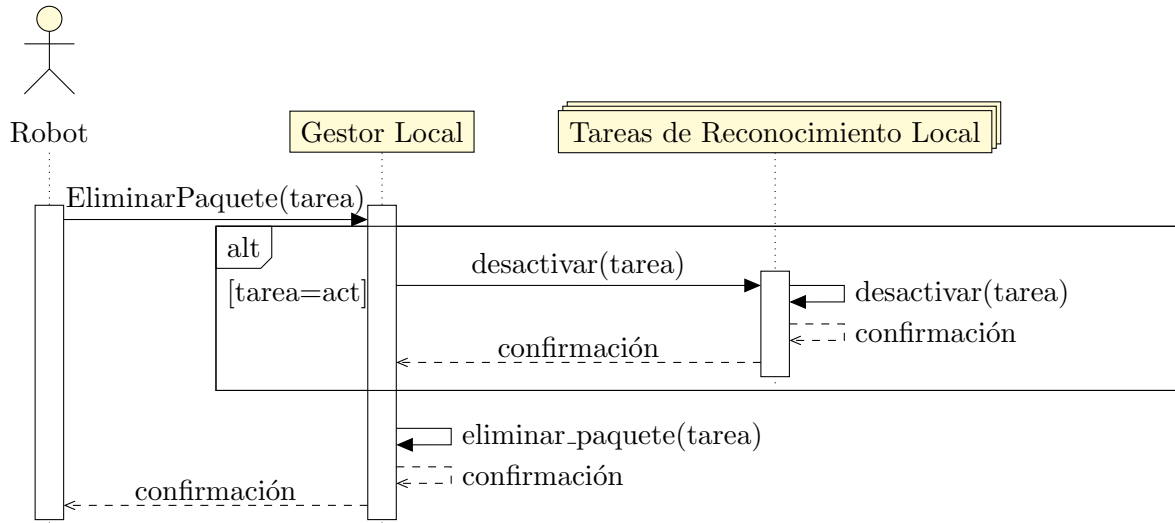
A.3.5.2.6. Diagrama de secuencia CU-06: Instalar un paquete de reconocimiento en el gestor local



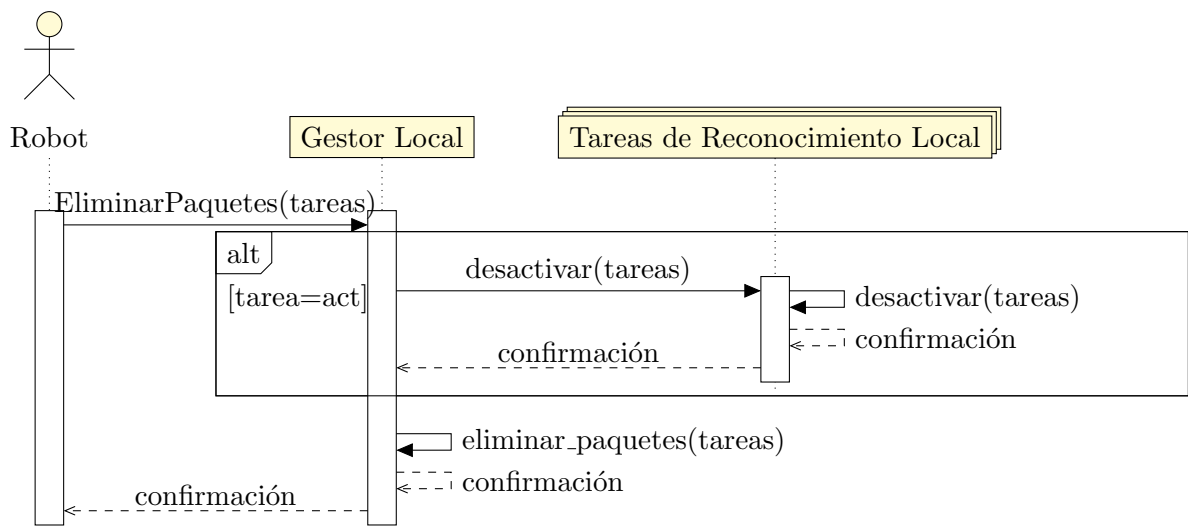
A.3.5.2.7. Diagrama de secuencia CU-07: Instalar una lista de paquetes de reconocimiento en el gestor local



A.3.5.2.8. Diagrama de secuencia CU-08: Eliminar un paquete de reconocimiento del gestor local



A.3.5.2.9. Diagrama de secuencia CU-09: Eliminar una lista de paquetes de reconocimiento del gestor remoto



A.3.5.3. Gestión de tareas de reconocimiento remoto

La gestión de tareas de reconocimiento remoto comprende todas las actividades que se encargan de la activación y desactivación de tareas de reconocimiento remoto. Las tareas de reconocimiento remoto son gestionadas por el operador, quien debe tomar la decisión de qué tareas de reconocimiento deben estar disponibles para ser solicitadas por el robot, idealmente todas tareas de reconocimiento en el gestor remoto deben estar disponibles. Para realizar cualquier actividad de activación o desactivación es necesario que el operador conozca las tareas de reconocimiento disponibles en el gestor remoto.

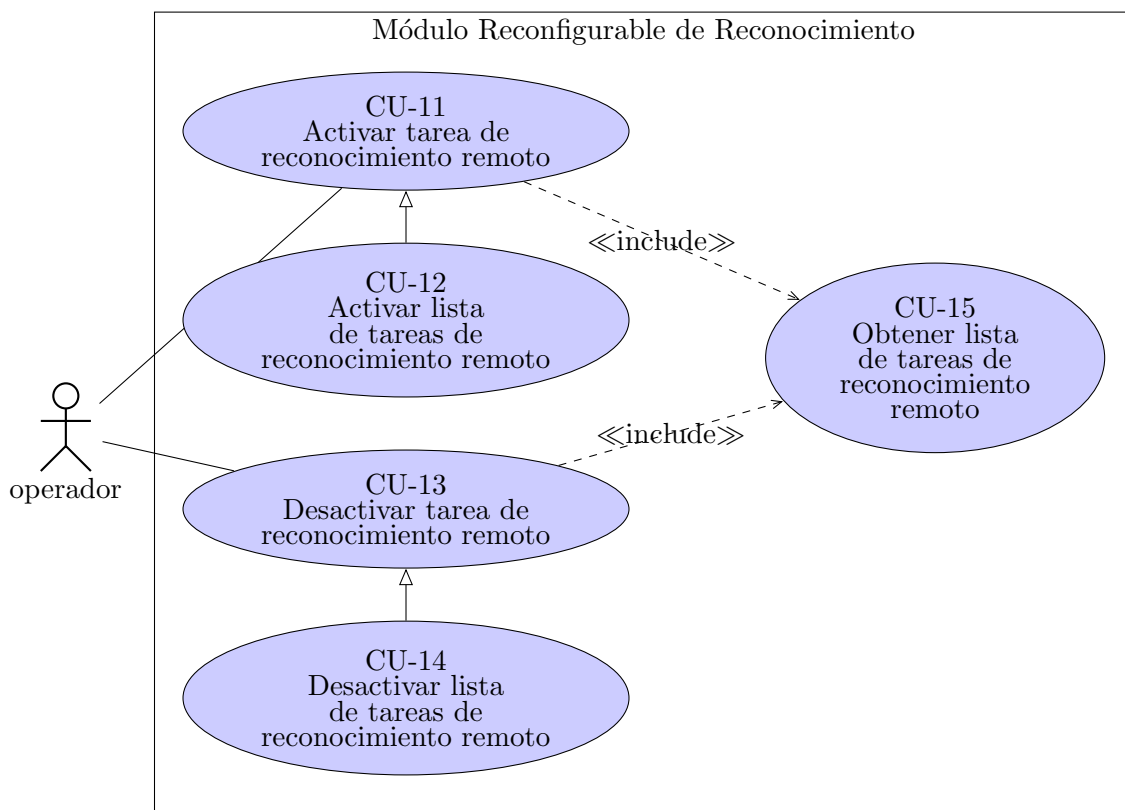


Figura A.4: Diagrama de caso de uso: Gestión de tareas de reconocimiento remoto

A.3.5.3.1. CU-11: Activar tarea de reconocimiento remoto

Tabla A.11: Descripción del Caso de Uso CU-11

Caso de Uso:	11
Nombre:	Activar tarea de reconocimiento remoto
Evento disparador:	El operador requiere activar una tarea de reconocimiento remoto.
Actores:	Operador
Dependencias:	CU-15: Obtener lista de tareas de reconocimiento remoto
Descripción:	Permite al operador solicitar la activación de una tarea de reconocimiento remoto, para ser usada posteriormente.
Pre-condiciones:	<ol style="list-style-type: none"> 1. El operador tiene la lista de las tareas disponibles de reconocimiento remoto. 2. Existe por lo menos un paquete de reconocimiento instalado en el gestor remoto.
Post-condiciones:	<ol style="list-style-type: none"> 1. La tarea específica de reconocimiento está activada y disponible para ser usada remotamente.
Flujo Normal:	<ol style="list-style-type: none"> 1. El operador selecciona una tarea específica para su activación, utilizando como referencia la lista de tareas disponibles en el gestor remoto. 2. El operador solicita al gestor remoto la activación de una tarea de reconocimiento específica. 3. El gestor remoto activa la tarea de reconocimiento. 4. Se actualiza el estado de la tarea de reconocimiento.
Flujo Alternativo:	<ol style="list-style-type: none"> 2.A La tarea de reconocimiento solicitada ya se encuentra activada. El caso de uso termina.

A.3.5.3.2. CU-12: Activar lista de tareas de reconocimiento remoto

Tabla A.12: Descripción del Caso de Uso CU-12

Caso de Uso:	12
Nombre:	Activar lista de tareas de reconocimiento remoto
Evento disparador:	El operador requiere activar una lista de tareas de reconocimiento remoto.
Actores:	Operador
Dependencias:	CU-15: Obtener lista de tareas de reconocimiento remoto
Descripción:	Permite al operador solicitar la activación de una lista de tareas de reconocimiento remoto, para ser usadas posteriormente.
Pre-condiciones:	<ol style="list-style-type: none"> 1. El operador tiene la lista de las tareas disponibles de reconocimiento remoto. 2. Existe por lo menos un paquete de reconocimiento instalado en el gestor remoto.
Post-condiciones:	<ol style="list-style-type: none"> 1. Las tareas específicas de reconocimiento remoto están activadas y disponibles para ser usadas.
Flujo Normal:	<ol style="list-style-type: none"> 1. El operador selecciona un conjunto de tareas específicas para su activación, utilizando la lista de tareas disponibles en el gestor remoto. 2. El operador solicita al gestor remoto la activación de una lista de tareas de reconocimiento. 3. El gestor remoto activa las tareas de reconocimiento. 4. Se actualiza el estado de las tareas de reconocimiento.
Flujo Alternativo:	<ol style="list-style-type: none"> 2.A Una o más tareas de reconocimiento ya se encuentran activadas en el gestor remoto. Se activan solo aquellas que se encuentran desactivadas.

A.3.5.3.3. CU-13: Desactivar tarea de reconocimiento remoto

Tabla A.13: Descripción del Caso de Uso CU-13

Caso de Uso:	13
Nombre:	Desactivar una tarea de reconocimiento remoto
Evento disparador:	El operador requiere desactivar una tarea de reconocimiento remoto.
Actores:	Operador
Dependencias:	CU-15: Obtener lista de tareas de reconocimiento remoto
Descripción:	Permite al operador solicitar la desactivación de una tarea de reconocimiento remoto.
Pre-condiciones:	<ol style="list-style-type: none"> 1. El operador tiene la lista de las tareas disponibles de reconocimiento remoto. 2. Existe por lo menos un paquete de reconocimiento instalado en el gestor remoto.
Post-condiciones:	<ol style="list-style-type: none"> 1. La tarea específica de reconocimiento remoto está desactivada.
Flujo Normal:	<ol style="list-style-type: none"> 1. El operador selecciona una tarea específica para su desactivación, utilizando como referencia la lista de tareas disponibles en el gestor remoto. 2. El operador solicita al gestor remoto la desactivación de una tarea de reconocimiento específica. 3. El gestor remoto desactiva la tarea de reconocimiento. 4. Se actualiza el estado de la tarea de reconocimiento.
Flujo Alternativo:	<ol style="list-style-type: none"> 2.A La tarea de reconocimiento solicitada ya se encuentra desactivada. El caso de uso termina.

A.3.5.3.4. CU-14: Desactivar lista de tareas de reconocimiento remoto

Tabla A.14: Descripción del Caso de Uso CU-14

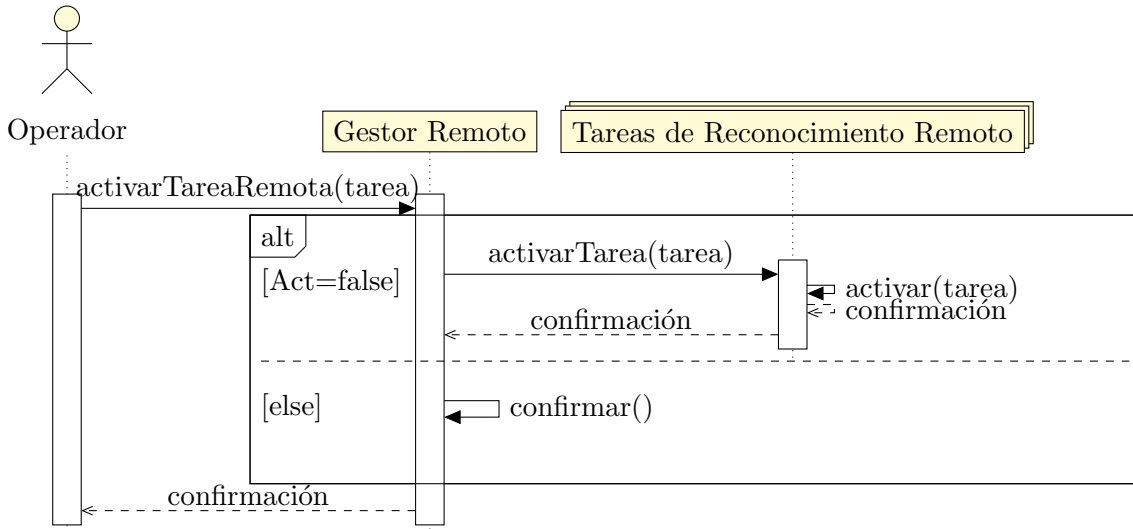
Caso de Uso:	14
Nombre:	Desactivar lista de tareas de reconocimiento remoto
Evento disparador:	El operador requiere desactivar una lista de tareas de reconocimiento remoto.
Actores:	Operador
Casos de uso relacionados:	CU-15: Obtener lista de tareas de reconocimiento remoto.
Descripción:	Permite al operador solicitar la desactivación de una lista de tareas de reconocimiento remoto.
Pre-condiciones:	<ol style="list-style-type: none"> 1. El operador tiene la lista de las tareas disponibles de reconocimiento remoto. 2. Existe por lo menos un paquete de reconocimiento instalado en el gestor remoto.
Post-condiciones:	<ol style="list-style-type: none"> 1. Las tareas de reconocimiento están desactivadas.
Flujo Normal:	<ol style="list-style-type: none"> 1. El operador selecciona un conjunto de tareas específicas para su desactivación, utilizando como referencia la lista de tareas disponibles en el gestor remoto. 2. El operador solicita al gestor remoto la desactivación de una lista de tareas de reconocimiento. 3. El gestor remoto desactiva las tareas de reconocimiento. 4. Se actualiza el estado de las tareas de reconocimiento.
Flujo Alternativo:	<ol style="list-style-type: none"> 2.A Una o más tareas de reconocimiento ya se encuentran desactivadas. Se desactivan solo aquellas que se encuentran activadas.

A.3.5.3.5. CU-15: Obtener lista de tareas de reconocimiento remoto

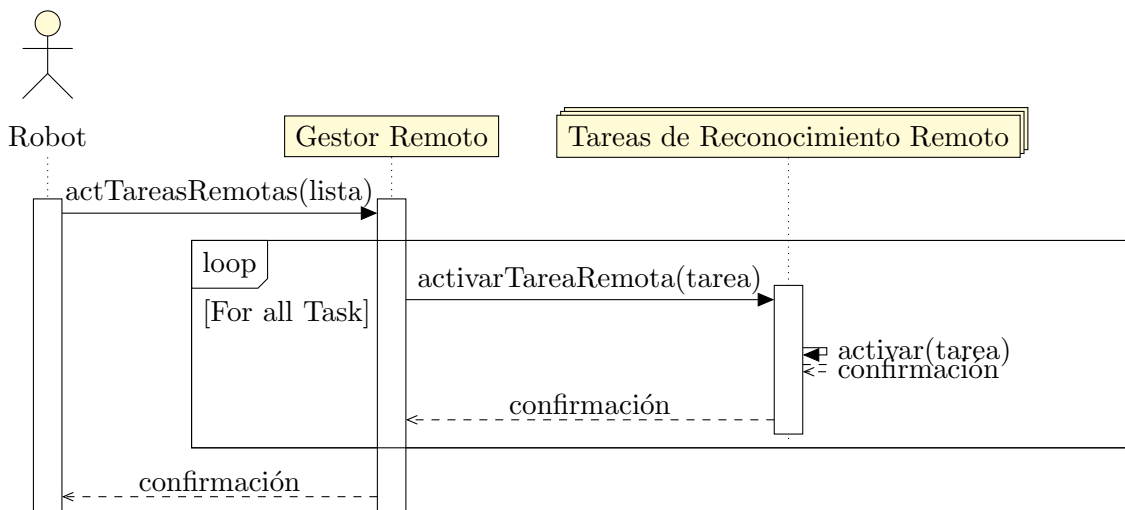
Tabla A.15: Descripción del Caso de Uso CU-15

Caso de Uso:	15
Nombre:	Obtener lista de tareas de reconocimiento remoto
Evento disparador:	El robot o el operador requiere conocer la lista de tareas de reconocimiento remoto.
Actores:	Robot y Operador
Dependencias:	CU-11: Activar tarea de reconocimiento remoto, CU-13: Desactivar tarea de reconocimiento remoto.
Descripción:	Permite al robot u operador, obtener la lista de tareas de reconocimiento disponibles en el gestor remoto.
Pre-condiciones:	1. El gestor remoto tiene instalado por lo menos un paquete de reconocimiento.
Post-condiciones:	1. El robot u operador, obtiene una lista de todas las tareas de reconocimiento remoto con información de sus estados.
Flujo Normal:	<ol style="list-style-type: none"> 1. El robot u operador, solicita al gestor remoto la lista de tareas de reconocimiento. 2. El gestor remoto retorna una lista de todas las tareas de reconocimiento disponibles junto con su estado (activado/desactivado). 3. El robot u operador, obtiene la lista de tareas de reconocimiento remoto.
Flujo Alternativo:	2.A El gestor remoto no tiene ninguna tarea de reconocimiento disponible. Se retorna una lista vacía.

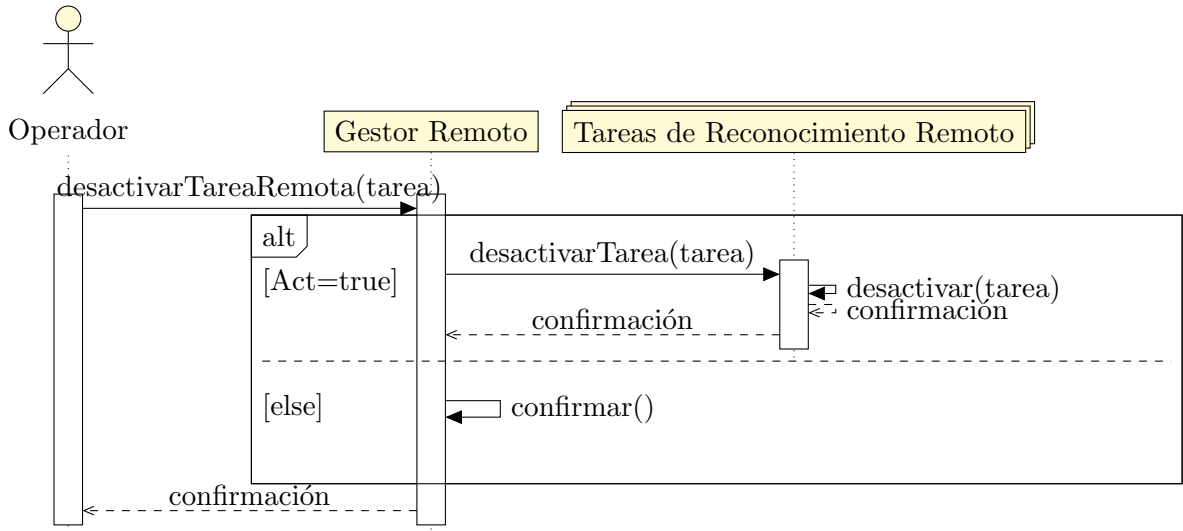
A.3.5.3.6. Diagrama de secuencia CU-11: Activar tarea de reconocimiento remoto



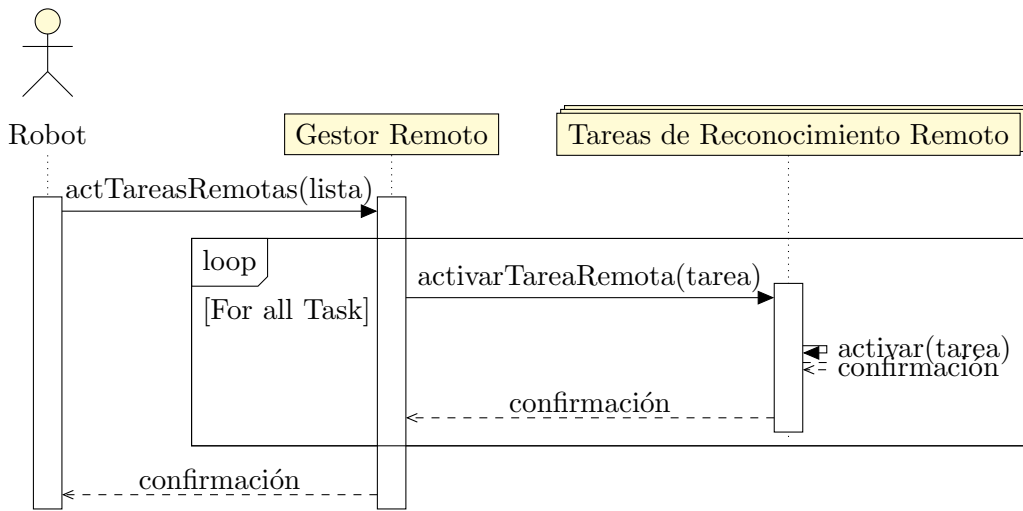
A.3.5.3.7. Diagrama de secuencia CU-12: Activar lista de tareas de reconocimiento remoto

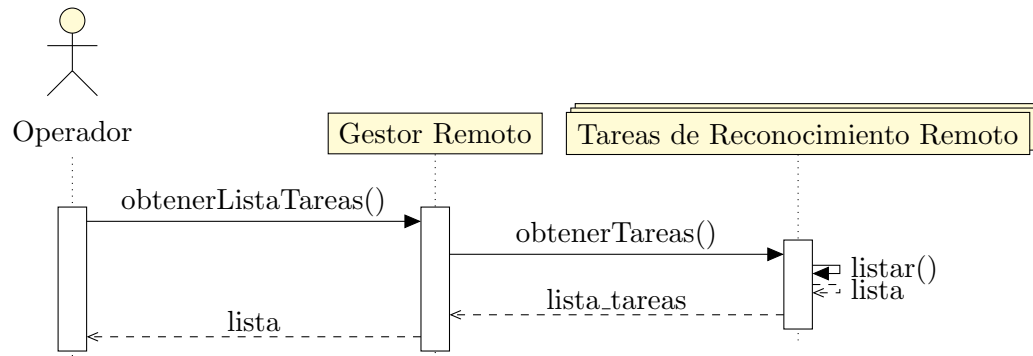


A.3.5.3.8. Diagrama de secuencia CU-13: Desactivar tarea de reconocimiento remoto



A.3.5.3.9. Diagrama de secuencia CU-14: Desactivar lista de tareas de reconocimiento remoto



A.3.5.3.10. Diagrama de secuencia CU-15: Obtener lista de tareas de reconocimiento remoto

A.3.5.4. Gestión de paquetes de reconocimiento remoto

La gestión de paquetes de reconocimiento remoto comprende todas las actividades de instalación y eliminación de paquetes en el procesamiento externo del robot. El operador es el encargado de la instalación y eliminación de paquetes en el gestor remoto, para realizar la instalación de un paquete el operador debe ingresar la dirección del mismo, y este paquete debe contener todos los requisitos detallados en la sección A.3.8. En el caso particular de la actividad de eliminación de paquetes, el operador debe conocer qué paquetes pueden ser eliminados, por lo que es necesario obtener la lista de tareas de reconocimiento que se encuentran disponibles en el gestor remoto.

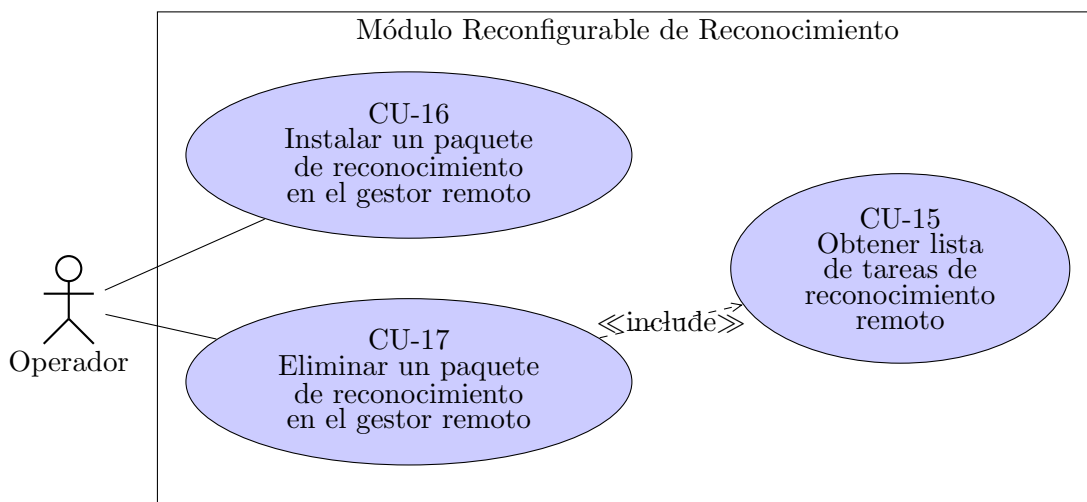


Figura A.5: Diagrama de caso de uso: Gestión de paquetes de reconocimiento remoto

A.3.5.4.1. CU-16: Instalar un paquete de reconocimiento en el gestor remoto

Tabla A.16: Descripción del Caso de Uso CU-16

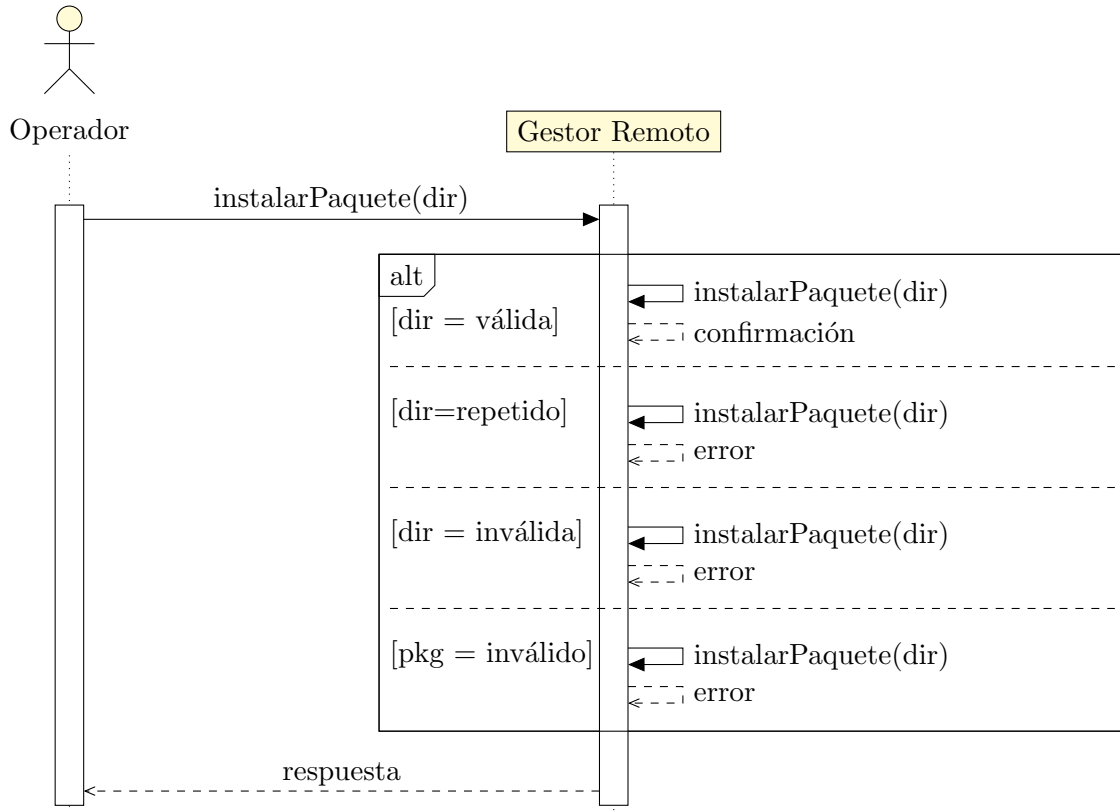
Caso de Uso:	16
Nombre:	Instalar un paquete de reconocimiento remoto
Evento disparador:	El operador requiere instalar un nuevo paquete de reconocimiento en el gestor remoto.
Actores:	Operador
Dependencias:	CU-17: Eliminar un paquete de reconocimiento remoto.
Descripción:	Permite al operador instalar un nuevo paquete de reconocimiento en el gestor remoto
Pre-condiciones:	<ol style="list-style-type: none"> 1. El operador se encuentra conectado al gestor remoto. 2. El operador conoce la dirección del nuevo paquete de reconocimiento.
Post-condiciones:	<ol style="list-style-type: none"> 1. El gestor remoto instala el nuevo paquete de reconocimiento.
Flujo Normal:	<ol style="list-style-type: none"> 1. El operador solicita al gestor remoto la instalación de un nuevo paquete, indicando la dirección del mismo. 2. El gestor remoto obtiene el paquete y procede a instalarlo. 3. El gestor remoto actualiza su lista de tareas de reconocimiento disponible.
Flujo Alternativo:	<ol style="list-style-type: none"> 1.A El operador ingresa la dirección de un paquete ya instalado. Se retorna un mensaje de error y termina el caso de uso. 1.B El operador ingresa una dirección no existente. Se retorna un mensaje de error y termina el caso de uso. 2.A El paquete obtenido por el gestor remoto es inválido. Se retorna un mensaje de error y termina el caso de uso.

A.3.5.4.2. CU-17: Eliminar un paquete de reconocimiento en el gestor remoto

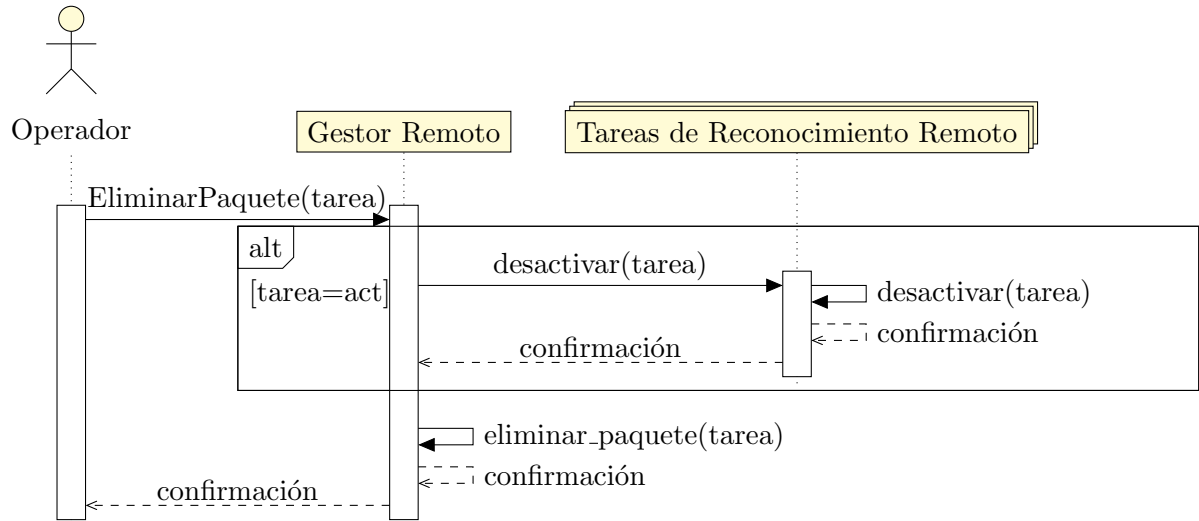
Tabla A.17: Descripción del Caso de Uso CU-17

Caso de Uso:	17
Nombre:	Eliminar un paquete de reconocimiento remoto
Evento disparador:	El operador requiere eliminar un paquete de reconocimiento remoto
Actores:	Operador
Dependencias:	CU-15: Obtener lista de tareas de reconocimiento remoto.
Descripción:	Permite al operador eliminar un paquete de reconocimiento del gestor remoto.
Pre-condiciones:	<ol style="list-style-type: none"> 1. El gestor remoto tiene instalado por lo menos un paquete de reconocimiento. 2. El operador tiene la lista de tareas de reconocimiento del gestor remoto.
Post-condiciones:	<ol style="list-style-type: none"> 1. El gestor remoto elimina el paquete de reconocimiento indicado por el operador.
Flujo Normal:	<ol style="list-style-type: none"> 1. El operador solicita al gestor remoto la eliminación de un paquete de reconocimiento, utilizando como referencia la lista de tareas de reconocimiento remoto. 2. El gestor remoto elimina el paquete. 3. Se actualiza la lista de tareas de reconocimiento en el gestor remoto.
Flujo Alternativo:	<ol style="list-style-type: none"> 3.A Si la tarea asociada al paquete de reconocimiento se encuentra activa, se desactiva y se elimina. Se continua con 4.

A.3.5.4.3. Diagrama de secuencia CU-16: Instalar un paquete de reconocimiento remoto



A.3.5.4.4. Diagrama de secuencia CU-17: Eliminar un paquete de reconocimiento del gestor remoto



A.3.5.5. Servicios de reconocimiento

Los servicios de reconocimiento comprenden las actividades de solicitudes de reconocimiento en el procesamiento interno o externo de robot, en otras palabras, son todas aquellas tareas de reconocimiento que pueden ser solicitadas por el robot. Para realizar una solicitud de reconocimiento local o remoto, es necesario que el robot obtenga la lista de tareas de reconocimiento disponible en el gestor local o en el gestor remoto, respectivamente.

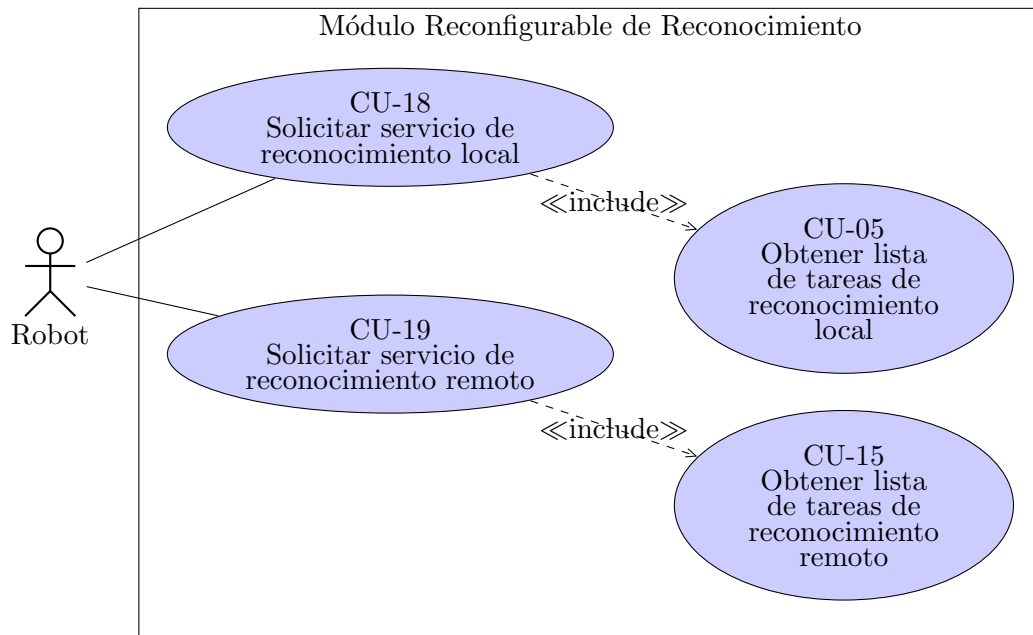


Figura A.6: Diagrama de caso de uso: Solicitud de servicios de reconocimiento

A.3.5.5.1. CU-18: Solicitar servicio de reconocimiento local

Tabla A.18: Descripción del Caso de Uso CU-18

Caso de Uso:	18
Nombre:	Solicitar servicio de reconocimiento local
Evento disparador:	El robot requiere un servicio de reconocimiento local.
Actores:	Robot
Dependencias:	CU-05: Obtener lista de tareas de reconocimiento local.
Descripción:	Permite al robot solicitar los servicios de las tareas de reconocimiento activadas en su procesamiento interno.
Pre-condiciones:	1. El robot tiene la lista de tareas disponibles de reconocimiento local.
Post-condiciones:	1. El robot recibe la respuesta del reconocimiento.
Flujo Normal:	<ol style="list-style-type: none"> 1. El robot selecciona una tarea de reconocimiento, utilizando como referencia la lista de tareas de reconocimiento local en estado activo. 2. El robot solicita el servicio de reconocimiento, indicando la dirección del archivo de entrada de la tarea específica de reconocimiento. 3. El robot recibe la respuesta de la tarea de reconocimiento local.
Flujo Alternativo:	<ol style="list-style-type: none"> 2.A La dirección indicada por el robot es inválida. La respuesta del servicio es nula. Se termina el caso de uso. 2.B El archivo de entrada es inválido o está corrupto. La respuesta del servicio es nula. Se termina el caso de uso.

A.3.5.5.2. CU-19: Solicitar un servicio de reconocimiento remoto

Tabla A.19: Descripción del Caso de Uso CU-19

Caso de Uso:	19
Nombre:	Solicitar servicio de reconocimiento remoto
Evento disparador:	El robot requiere un servicio de reconocimiento remoto.
Actores:	Robot
Dependencias:	CU-15: Obtener lista de tareas de reconocimiento remoto.
Descripción:	Permite al robot solicitar los servicios de las tareas de reconocimiento activadas en su procesamiento externo.
Pre-condiciones:	1. El robot tiene la lista de tareas disponibles de reconocimiento remoto.
Post-condiciones:	1. El robot recibe la respuesta del reconocimiento.
Flujo Normal:	<ol style="list-style-type: none"> 1. El robot selecciona una tarea de reconocimiento, utilizando como referencia la lista de tareas de reconocimiento remoto en estado activo. 2. El robot solicita el servicio de reconocimiento, indicando la dirección del archivo de entrada de la tarea específica de reconocimiento. 3. El robot recibe la respuesta de la tarea de reconocimiento remota.
Flujo Alternativo:	<ol style="list-style-type: none"> 2.A La dirección indicada por el robot es inválida. La respuesta del servicio es nula. Se termina el caso de uso. 2.B El archivo de entrada es inválido o está corrupto. La respuesta del servicio es nula. Se termina el caso de uso.

A.3.5.5.3. Diagrama de secuencia CU-18: Solicitar servicio de reconocimiento local

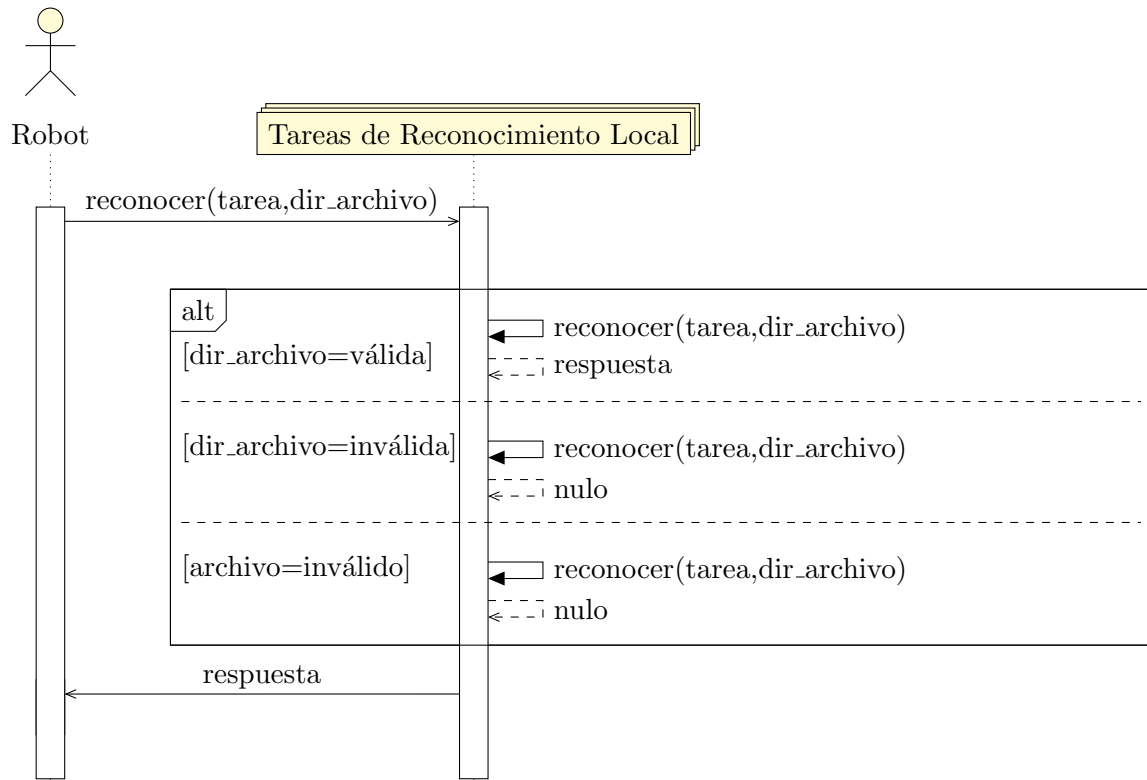


Figura A.7: Diagrama de secuencia CU-18: Solicitar servicio de reconocimiento local.

A.3.5.5.4. Diagrama de secuencia CU-19: Solicitar servicio de reconocimiento remoto

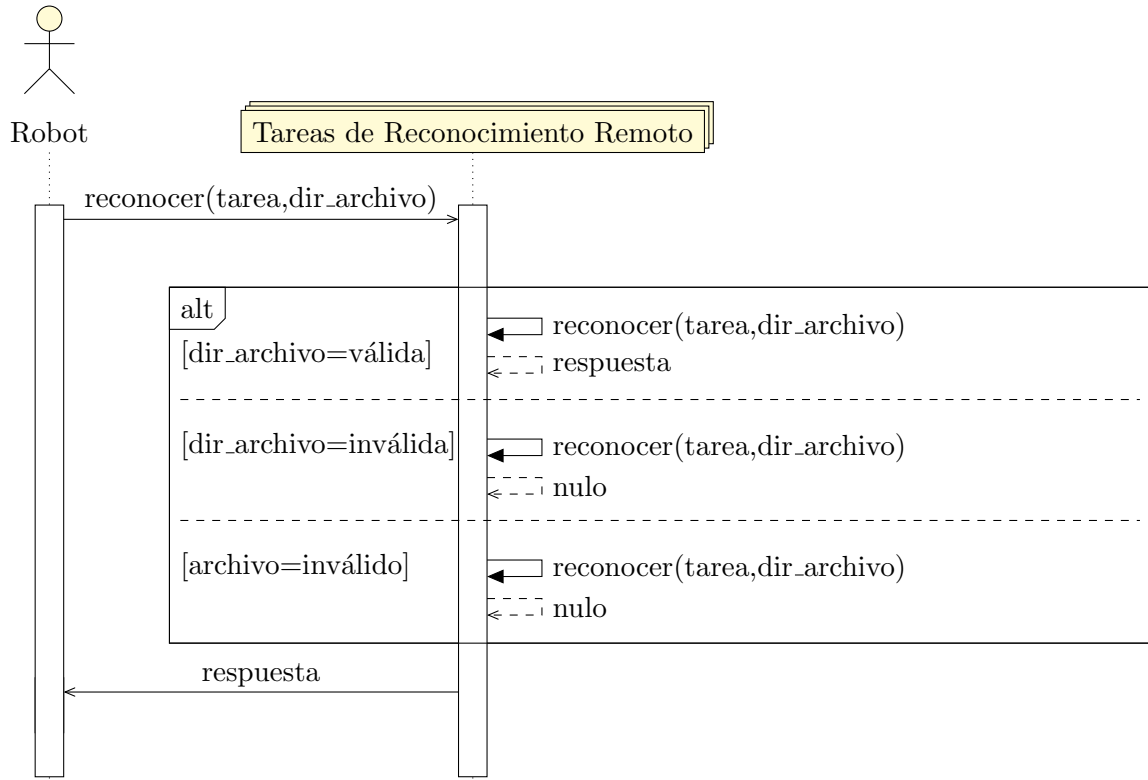


Figura A.8: Diagrama de secuencia CU-19: Solicitar servicio de reconocimiento remoto.

A.3.6. Requisitos de ejecución

- Para ejecutar correctamente el Módulo Reconfigurable de Reconocimiento, se debe disponer de 2 computadoras. Además, éstas deberán estar conectadas a la misma red.

A.3.7. Restricciones de diseño

- La versión del framework ROS para realizar la implementación debe ser Kinetic Kame para la versión 16.04 de Ubuntu; esto bajo la justificación de que es la versión con mayor soporte en el momento de esta investigación. Según [37], los requerimientos de hardware del sistema operativo Ubuntu 16.04, deben cumplir como mínimo los siguientes requisitos:

1. Procesador de 1Ghz.
2. 1.5 Gb de memoria RAM.
3. 10 Gb de espacio en el disco duro para la instalación.
4. Manejadores de CD/DVD o puerto USB para la instalación.
5. Acceso a Internet para la instalación de actualizaciones y paquetes específicos del hardware de la computadora.

Por lo tanto, las computadoras que servirán de procesamiento interno y externo del robot deberán cumplir con los anteriores requisitos.

A.3.8. Otros Requisitos

Los paquetes que se instalarán en el gestor remoto deberán cumplir con los siguientes requisitos:

1. La estructura general de un paquete de reconocimiento debe ser de la siguiente manera:

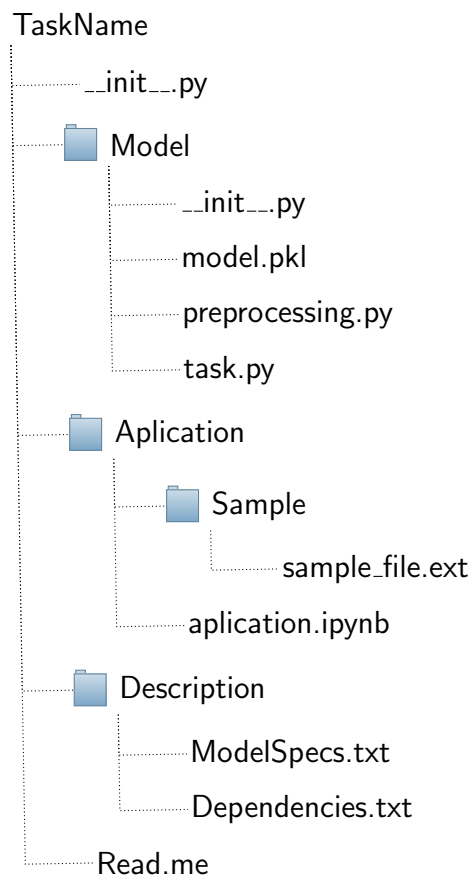


Figura A.9: Árbol de directorios de un paquete de reconocimiento.

A continuación se presenta una descripción de cada carpeta y archivo perteneciente al árbol de directorio anterior, ver Figura A.9.

a) **TaskName**: Es el directorio principal y además lleva como nombre el identificador de la tarea de reconocimiento. Contiene 3 directorios: Model, Application y Description, y 3 archivos:

- **__init__.py**: Archivo de código python, para la importación de archivos.
- **Read.me**: Archivo con comentarios adicionales sobre el funcionamiento y características adicionales del paquete.

b) **Model**: Es un directorio en el cual se almacenan 3 archivos:

- **__init__.py**: Archivo de código python, para la importación de archivos.
- **model.pkl**: Es un archivo que representa una versión persistente del modelo entrenado de la tarea de reconocimiento.
- **preprocessing.py**: Es un archivo de código python que contiene los algoritmos necesarios para pre-procesar la entrada de la tarea de reconocimiento. Este archivo debe ser usado por el archivo task.py, el cual se encarga de cargar la representación persistente del modelo en memoria para su posterior uso.
- **task.py** Es un archivo de código python que contiene los algoritmos necesarios para recibir la dirección de la muestra o archivo de entrada, cargar la representación persistente del modelo en memoria y además retornar el valor correspondiente del reconocimiento de la muestra en particular.

Para realizar todos los pasos anteriores y permitir una alta cohesión entre los componentes de la arquitectura, el archivo task.py debe contener una clase con el nombre “RosTask”, que implemente los métodos de la siguiente clase abstracta (ver Figura A.10):

c) **Application**: Este directorio contiene un conjunto de archivos que permitirán mostrar el funcionamiento de las tareas de reconocimiento.

- **Sample**: Este directorio contiene el archivo o archivos de muestra (**sample_file.ext**) con la respectiva extensión que utiliza el modelo

Task
<i>+ execute(String dir): String</i>
<i>- restore_model(String dir): Model</i>
<i>- preprocessing(String dir) : Vector</i>
<i>- prediction(Vector vc): String</i>

Figura A.10: Clase abstracta “Task” para la implementación de paquetes de reconocimiento

de la tarea de reconocimiento. Estas muestras son utilizadas por el archivo `application.ipynb`, el cual muestra un ejemplo concreto del reconocimiento del mismo.

- **application.ipynb:** Este archivo de extensión ipynb, es un jupyter notebook [39], el cual contiene descripciones y diferentes algoritmos en código python, detallando de manera concisa un ejemplo concreto del reconocimiento de los archivos de la carpeta **Sample**.

d) **Description:** Este directorio contiene el archivo **ModelSpec.txt**, el cual contiene una especificación concreta de características generales del modelo y del algoritmo. Todos los paquetes deben contener este archivo y además este debe contener la siguiente información:

- Nombre del algoritmo de aprendizaje de máquina.
- Hiperparámetros utilizados por algoritmo de aprendizaje de máquina.
- Tasa de reconocimiento.
- Número de muestras de entrenamiento.
- Número de muestras de prueba.
- Número de muestras de validación (si aplica).
- Nombre de las bases de datos utilizadas.
- Especificación detallada del formato de entrada del modelo.
- Descripción concreta del tipo de reconocimiento.
- Tipo de salida: Discreta o continua.
- Número de clases (si aplica).

- Otros.

Adicionalmente, se encuentra el archivo **Dependencies.txt**, el cual contiene un archivo con todas las dependencias necesarias para ejecutar los códigos encontrados en el paquete.

2. Todos los archivos de código python incluyendo los códigos en el jupyter notebook (archivo **aplication.ipynb**), deben estar escritos en python 3.0.

Bibliografía

- [1] “Ieee recommended practice for software requirements specifications,” *IEEE Std 830-1998*, pp. 1–40, Oct 1998.
- [2] N. M. AboElenein, K. M. Amin, M. Ibrahim, and M. M. Hadhoud, “Improved text-independent speaker identification system for real time applications,” in *Electronics, Communications and Computers (JEC-ECC), 2016 Fourth International Japan-Egypt Conference on*. IEEE, 2016, pp. 58–62.
- [3] M. T. Al-Kaltakchi, W. L. Woo, S. S. Dlay, and J. A. Chambers, “Comparison of i-vector and gmm-ubm approaches to speaker identification with timit and nist 2008 databases in challenging environments,” in *Signal Processing Conference (EUSIPCO), 2017 25th European*. IEEE, 2017, pp. 533–537.
- [4] C.-N. Anagnostopoulos, T. Iliou, and I. Giannoukos, “Features and classifiers for emotion recognition from speech: a survey from 2000 to 2011,” *Artificial Intelligence Review*, vol. 43, no. 2, pp. 155–177, Feb 2015. [Online]. Available: <https://doi.org/10.1007/s10462-012-9368-5>
- [5] A. Araujo, J. Pérez, and W. Rodriguez, “Aplicación de una red neuronal convolucional para el reconocimiento de personas a través de la voz.” Sexta Conferencia Nacional de Computación, Informática y Sistemas., 11 2018.
- [6] P. Barros, G. I. Parisi, D. Jirak, and S. Wermter, “Real-time gesture recognition using a humanoid robot with a deep neural architecture,” in *Humanoid Robots (Humanoids), 2014 14th IEEE-RAS International Conference on*. IEEE, 2014, pp. 646–651.

- [7] R. Bates Graber, “Ekman: The face of man: Expressions of universal emotions in a new guinea village,” 05 2017.
- [8] F. Bimbot, J.-F. Bonastre, C. Fredouille, G. Gravier, I. Magrin-Chagnolleau, S. Meignier, T. Merlin, J. Ortega-García, D. Petrovska-Delacrétaz, and D. A. Reynolds, “A tutorial on text-independent speaker verification,” *EURASIP Journal on Advances in Signal Processing*, vol. 2004, no. 4, p. 101962, 2004.
- [9] G. Bonaccorso, *Machine Learning Algorithms*. Packt Publishing, 2017. [Online]. Available: <https://books.google.co.ve/books?id=-ZDDwAAQBAJ>
- [10] H. Bruyninckx, “Open robot control software: the orocos project,” in *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, vol. 3. IEEE, 2001, pp. 2523–2528.
- [11] F. Burkhardt, A. Paeschke, M. A. Rolfes, W. F. Sendlmeier, and B. Weiss, “A database of german emotional speech,” in *INTERSPEECH*, 2005.
- [12] J. P. Campbell, “Testing with the yoho cd-rom voice verification corpus,” in *Acoustics, Speech, and Signal Processing, 1995. ICASSP-95., 1995 International Conference on*, vol. 1. IEEE, 1995, pp. 341–344.
- [13] P. Chandrasekar, S. Chapaneri, and D. Jayaswal, “Emotion recognition from speech using discriminative features,” vol. 101, pp. 31–36, 09 2014.
- [14] Y. D. Chavhan, B. S. Yelure, and K. N. Tayade, “Speech emotion recognition using rbf kernel of libsvm,” in *2015 2nd International Conference on Electronics and Communication Systems (ICECS)*, Feb 2015, pp. 1132–1135.
- [15] N. Churamani, P. Anton, M. Brügger, E. Fließwasser, T. Hummel, J. Mayer, W. Mustafa, H. G. Ng, T. L. C. Nguyen, Q. Nguyen, *et al.*, “The impact of personalisation on human-robot interaction in learning scenarios,” in *Proceedings of the 5th International Conference on Human Agent Interaction*. ACM, 2017, pp. 171–180.

- [16] F. Cummins, M. Grimaldi, T. Leonard, J. Simko, F. Cummins@ucd, M. Ie, and I. , “The chains speech corpus: Characterizing individual speakers,” 08 2018.
- [17] P. P. Dahake, K. Shaw, and P. Malathi, “Speaker dependent speech emotion recognition using mfcc and support vector machine,” in *2016 International Conference on Automatic Control and Dynamic Optimization Techniques (ICACDOT)*, Sept 2016, pp. 1080–1084.
- [18] A. Dattalo. (2018) Wiki ros introduction. [Online]. Available: <http://wiki.ros.org/ROS/Introduction>
- [19] M. de Jong, K. Zhang, A. M. Roth, T. Rhodes, R. Schmucker, C. Zhou, S. Ferreira, J. Cartucho, and M. Veloso, “Towards a robust interactive and learning social robot,” in *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, 2018, pp. 883–891.
- [20] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, “Front-end factor analysis for speaker verification,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2011.
- [21] A. P. Dempster, N. M. Laird, and D. B. Rubin, “Maximum likelihood from incomplete data via the em algorithm,” *Journal of the royal statistical society. Series B (methodological)*, pp. 1–38, 1977.
- [22] I. El Makrini, K. Merckaert, D. Lefeber, and B. Vanderborght, “Design of a collaborative architecture for human-robot assembly tasks,” in *Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on*. IEEE, 2017, pp. 1624–1629.
- [23] L. Feng and L. K. Hansen, *A new database for speaker recognition*. IMM, Informatik og Matematisk Modelling, DTU, 2005.
- [24] M. Fowler and K. Scott, *UML gota a gota*. Pearson Educación, 1999. [Online]. Available: <https://books.google.co.ve/books?id=AL0YkFeaHwIC>

- [25] J. S. Garofolo, “Timit acoustic phonetic continuous speech corpus,” *Linguistic Data Consortium, 1993*, 1993.
- [26] B. Gerkey, R. T. Vaughan, and A. Howard, “The player/stage project: Tools for multi-robot and distributed sensor systems,” in *Proceedings of the 11th international conference on advanced robotics*, vol. 1, 2003, pp. 317–323.
- [27] M. Ghai, S. Lal, S. Duggal, and S. Manik, “Emotion recognition on speech signals using machine learning,” in *2017 International Conference on Big Data Analytics and Computational Intelligence (ICBDAC)*, March 2017, pp. 34–39.
- [28] T. Giannakopoulos and A. Pikrakis, *Introduction to audio analysis: a MATLAB approach*. Academic Press, 2014.
- [29] G. Gordon, S. Spaulding, J. K. Westlund, J. J. Lee, L. Plummer, M. Martinez, M. Das, and C. Breazeal, “Affective personalization of a social robot tutor for children’s second language skills.” in *AAAI*, 2016, pp. 3951–3957.
- [30] S. Grochowski, “Corpora-speech database for polish diphones,” in *Fifth European Conference on Speech Communication and Technology*, 1997.
- [31] J. H. L. Hansen and S. E. Bou-Ghazale, “Getting started with susas: a speech under simulated and actual stress database,” in *EUROSPEECH*, 1997.
- [32] J. Hennebert, H. Melin, D. Petrovska, and D. Genoud, “Polycost: A telephone-speech database for speaker recognition,” *Speech communication*, vol. 31, no. 2-3, pp. 265–270, 2000.
- [33] J. I. Hualde, *Los sonidos del español: Spanish Language edition*. Cambridge University Press, 2013.
- [34] T. Iliou and C.-N. Anagnostopoulos, “Classification on speech emotion recognition-a comparative study,” *animation*, vol. 4, p. 5, 2010.
- [35] P. Jackson and S. Haq, “Surrey audio-visual expressed emotion (savee) database,” *University of Surrey: Guildford, UK*, 2014.

- [36] J.-S. R. Jang, “Audio signal processing and recognition,” *Roger Jang’s Homepage*, 2011.
- [37] jbicha. (2018) Wiki ubuntu gnome 16.04.5 lts. [Online]. Available: <https://wiki.ubuntu.com/XenialXerus/ReleaseNotes/UbuntuGNOME>
- [38] V. Kirandzhiska and N. Ackovska, “Sound features used in emotion classification,” 2012.
- [39] T. Kluyver, B. Ragan-Kelley, F. Pérez, B. E. Granger, M. Bussonnier, J. Frederic, K. Kelley, J. B. Hamrick, J. Grout, S. Corlay, *et al.*, “Jupyter notebooks—a publishing format for reproducible computational workflows.” in *ELPUB*, 2016, pp. 87–90.
- [40] A. N. A. Kumar and S. A. Muthukumaraswamy, “Text dependent voice recognition system using mfcc and vq for security applications,” in *Electronics, Communication and Aerospace Technology (ICECA), 2017 International conference of*, vol. 2. IEEE, 2017, pp. 130–136.
- [41] O.-W. Kwon, K. Chan, J. Hao, and T.-W. Lee, “Emotion recognition by speech signals,” in *Eighth European Conference on Speech Communication and Technology*, 2003.
- [42] Y. Li, H. Wang, B. Ding, P. Shi, and X. Liu, “Toward qos-aware cloud robotic applications: a hybrid architecture and its implementation,” in *Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress (UIC/ATC/ScalCom/CBDCOM/IoP/SmartWorld), 2016 Intl IEEE Conferences*. IEEE, 2016, pp. 33–40.
- [43] J.-C. Liu, F.-Y. Leu, G.-L. Lin, and H. Susanto, “An mfcc-based text-independent speaker identification system for access control,” *Concurrency and Computation: Practice and Experience*, vol. 30, no. 2, p. e4255, 2018.
- [44] Y. Lukic, C. Vogt, O. Dürr, and T. Stadelmann, “Speaker identification and clustering using convolutional neural networks,” in *2016 IEEE 26th International*

- Workshop on Machine Learning for Signal Processing (MLSP), Vietri sul Mare, Italy, 13-16 Sept. 2016.* IEEE, 2016.
- [45] A. Mahtani, L. Sanchez, E. Fernandez, and A. Martinez, *Effective Robotics Programming with ROS*. Packt Publishing, 2016. [Online]. Available: <https://books.google.co.ve/books?id=FtDcDgAAQBAJ>
- [46] O. Martin, I. Kotsia, B. Macq, and I. Pitas, “The enterface’05 audio-visual emotion database,” in *Proceedings of the 22Nd International Conference on Data Engineering Workshops*, ser. ICDEW ’06. Washington, DC, USA: IEEE Computer Society, 2006, pp. 8–. [Online]. Available: <http://dx.doi.org/10.1109/ICDEW.2006.145>
- [47] A. Maurya, D. Kumar, and R. Agarwal, “Speaker recognition for hindi speech signal using mfcc-gmm approach,” *Procedia Computer Science*, vol. 125, pp. 880–887, 2018.
- [48] H. Melin, “Databases for speaker recognition: Activities in cost250 working group 2,” *COST 250-Speaker Recognition in Telephony, Final Report 1999*, 1999.
- [49] M. Montemerlo, N. Roy, and S. Thrun, “Perspectives on standardization in mobile robot programming: The carnegie mellon navigation (carmen) toolkit,” in *Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, vol. 3. IEEE, 2003, pp. 2436–2441.
- [50] N. Morán, J. Pérez, and W. Rodriguez, “Reconocimiento de estados emocionales de personas mediante la voz utilizando algoritmos de aprendizaje de máquina,” vol. 5, pp. 41–52, 12 2018.
- [51] A. Nagrani, J. S. Chung, and A. Zisserman, “Voxceleb: a large-scale speaker identification dataset,” *arXiv preprint arXiv:1706.08612*, 2017.
- [52] N. M. Nasrabadi, “Pattern recognition and machine learning,” *Journal of electronic imaging*, vol. 16, no. 4, p. 049901, 2007.

- [53] B. Obama, *The audacity of hope: Thoughts on reclaiming the American dream*. Canongate Books, 2007.
- [54] E. Obediente, *Fonética y fonología*. Universidad Los Andes, 1998.
- [55] J. O’Kane, *A Gentle Introduction to ROS*. O’Kane, 2013. [Online]. Available: <https://books.google.co.ve/books?id=OzFHngEACAAJ>
- [56] Y. Pan, P. Shen, and L. Shen, “Feature extraction and selection in speech emotion recognition,” *Proceeding of the onlinepresent. org*, vol. 2, pp. 64–69, 2012.
- [57] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, *et al.*, “Scikit-learn: Machine learning in python,” *Journal of machine learning research*, vol. 12, no. Oct, pp. 2825–2830, 2011.
- [58] A. K. Pérez, C. A. Quintero, S. Rodríguez, E. Rojas, O. Peña, and F. De La Rosa, “Identification of multimodal signals for emotion recognition in the context of human-robot interaction,” in *International Symposium on Intelligent Computing Systems*. Springer, 2018, pp. 67–80.
- [59] M. Quigley, K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, “Ros: an open-source robot operating system,” in *ICRA Workshop on Open Source Software*, 2009.
- [60] A. Quilis and J. A. Fernández, *Curso de fonética y fonología del española: para estudiantes angloamericanos*. Consejo Superior de Investigaciones Científicas, Instituto”Miguel de Cervantes,”), 1973, vol. 2.
- [61] J. G. Razuri, D. Sundgren, R. Rahmani, A. Larsson, A. M. Cardenas, and I. Bonet, “Speech emotion recognition in emotional feedback for human-robot interaction,” *International Journal of Advanced Research in Artificial Intelligence*, vol. 4, no. 2, 2015. [Online]. Available: <http://dx.doi.org/10.14569/IJARAI.2015.040204>
- [62] P. Refaeilzadeh, L. Tang, and H. Liu, *Cross-Validation*. Boston, MA:

- Springer US, 2009, pp. 532–538. [Online]. Available: https://doi.org/10.1007/978-0-387-39940-9_565
- [63] D. Reynolds, “Universal background models,” in *Encyclopedia of biometrics*. Springer, 2009, pp. 1349–1352.
- [64] D. A. Reynolds, R. C. Rose, *et al.*, “Robust text-independent speaker identification using gaussian mixture speaker models,” *IEEE transactions on speech and audio processing*, vol. 3, no. 1, pp. 72–83, 1995.
- [65] V. P. Rosas, R. Mihalcea, and L. Morency, “Multimodal sentiment analysis of spanish online videos,” *IEEE Intelligent Systems*, vol. 28, no. 3, pp. 38–45, May 2013.
- [66] I. Stefanus, R. J. Sarwono, and M. I. Mandasari, “Gmm based automatic speaker verification system development for forensics in bahasa indonesia,” in *Instrumentation, Control, and Automation (ICA), 2017 5th International Conference on*. IEEE, 2017, pp. 56–61.
- [67] S. Steidl, “Automatic classification of emotion related user states in spontaneous children’s speech,” 2009.
- [68] Z.-H. Tan, N. B. Thomsen, X. Duan, E. Vlachos, S. E. Shepstone, M. H. Rasmussen, and J. L. Højvang, “isociobot: A multimodal interactive social robot,” *International Journal of Social Robotics*, vol. 10, no. 1, pp. 5–19, 2018.
- [69] C. J. Van der Hofstadt Román, *El libro de las habilidades de comunicación*. Ediciones Díaz de Santos, 2005.
- [70] F. Wilcoxon, “Individual comparisons by ranking methods,” *Biometrics bulletin*, vol. 1, no. 6, pp. 80–83, 1945.