

# Arquitectura de Control SocialRobotBesa

ESPECIFICACIÓN DE LA ARQUITECTURA  
JUAN SEBASTIAN LEON SUAREZ

## Contenido

Introducción .....	5
Requisitos del SMA SocialRobotBesa .....	5
Requerimientos No Funcionales .....	5
Requerimientos Funcionales .....	6
Diseño General del SMA de SocialRobotBesa .....	9
Módulo de Integración .....	26
Agente Proveedor de Servicio .....	26
Módulo de Soporte .....	28
Agente Interactivo .....	28
Agente Interprete Emocional .....	30
Agente Interprete Emocional del Usuario .....	<b>¡Error! Marcador no definido.</b>
Agente Detector de Riesgos .....	31
Agente Detector de Riesgos Éticos .....	32
Módulo de Decisión .....	34
Agente De Aprendizaje .....	34
Agente de Explicabilidad .....	35
Agente de Cooperación .....	36
Agente de Cooperación Humana .....	38
Agente de Creencias .....	39
Estado Físico del Agente .....	41
Estado de la Interacción .....	41
Perfil del Usuario .....	43
Preocupaciones Éticas .....	44
Estado Psicológico del Agente .....	45
Información de los SH .....	47
Accidentes .....	48
Afiliações .....	48
Modelo del Mundo .....	49
Agente de Motivación .....	49
Analizador de Metas .....	49
Motor de Deliberación .....	51
Razonador de Medios y Fines .....	53

Agente de Acción.....	53
Cambiando de Robot.....	55
Cambiando de Contexto de Aplicación .....	55
Agregar una Meta de Servicio .....	55
Bibliografía .....	56

## Lista de Ilustraciones

Ilustración 1. Descomposición de Metas.....	19
Ilustración 2. Roles de Interacción SocialRobotBesa.....	24
Ilustración 3. Arquitectura SocialRobotBesa.....	26
Ilustración 4. Agente Proveedor de Servicio.....	27
Ilustración 5. Agente Interactivo.....	28
Ilustración 6. Agente Interprete Emocional del Usuario.....	30
Ilustración 7. Proceso de fusión multimodal, tomado de [9].....	31
Ilustración 8. Agente Detector de Riesgos.....	32
Ilustración 9. Agente Detector de Riesgos Éticos.....	33
Ilustración 10. Agente de Aprendizaje.....	34
Ilustración 11. Agente de Explicabilidad.....	36
Ilustración 12. Agente de Cooperación.....	37
Ilustración 13. Agente de Cooperación Humana.....	38
Ilustración 14. Agente de Creencias.....	40
Ilustración 15. Estado Físico del Agente.....	41
Ilustración 16. Estado de Interacción.....	42
Ilustración 17. Perfil del Usuario.....	43
Ilustración 18. Preocupaciones éticas.....	45
Ilustración 19. Estado Psicológico del Agente.....	46
Ilustración 20. Cálculo de la valencia del evento emocional, tomado de [3].....	47
Ilustración 21. Información de Terceros.....	47
Ilustración 22. Información de Afiliaciones.....	48
Ilustración 23. Agente de Motivación.....	49
Ilustración 24. Metas Latentes.....	51
Ilustración 25. Agente Interactivo.....	54

## Lista de Tablas

Tabla 1. Requerimientos No Funcionales .....	6
Tabla 2. Requerimientos de Adaptabilidad .....	6
Tabla 3. Requerimientos de Empatía.....	7
Tabla 4. Requerimientos de Cooperación Multi Agente.....	7
Tabla 5. Requerimientos de Mediación Social.....	7
Tabla 6. Requerimientos de Seguridad.....	8
Tabla 7. Requerimientos de Autonomía .....	8
Tabla 8. Requerimientos de Toma de Decisiones .....	8
Tabla 9. Requerimientos de Manejo de Recursos .....	9
Tabla 10. Metas del SMA para la arquitectura de control SocialRobotBesa .....	16
Tabla 11. Recursos y Habilidades.....	22
Tabla 12. Roles Identificados .....	24
Tabla 13. Agentes de SocialRobotBesa.....	25
Tabla 14. Eventos del Despachador de Eventos .....	29
Tabla 15. Mapeo entre eventos.....	30
Tabla 16. Estado de las Primitivas .....	55

## Introducción

En el presente paper, se va a demostrar el diseño del sistema multiagentes que resultó del análisis de los requisitos encontrados durante la revisión del estado del arte y de a las visitas a los hogares geriátricos Calucé y Cedros. El sistema tiene como objetivo “manejar la imprevisibilidad de la interacción humano robot para generar bienestar en el usuario”.

La arquitectura le permite al robot poder reaccionar de manera oportuna, priorizando eventos de emergencia, como lo son riesgos en los cuales puede incurrir el usuario; y errores o excepciones propios del hardware y software. Así mismo, la arquitectura le facilita al robot poder relacionarse correctamente con el usuario de manera que la interacción tenga una alta aceptabilidad, al involucrar aspectos interactivos, emocionales y de personalización en los servicios provistos.

El artículo inicia con una breve introducción a los requisitos de los SARs para el manejo de la imprevisibilidad. Posteriormente, el capítulo 2 tratará la aplicación de la metodología AOPOA en el problema, donde se identificaron 11 agentes. La arquitectura de cada agente y los procesos que lo acompañan serán detallados en capítulos subsecuentes.

## Requisitos del SMA SocialRobotBesa

Para definir los requerimientos funcionales y no funcionales utilizados para la arquitectura SocialRobotBesa, se tuvo en cuenta la revisión bibliográfica realizada así como dos visitas realizadas a lo largo del proceso del trabajo de grado. Se realizó una entrevista a Juan Rojas, Gerente de Calucé Senior Living y otra a Eyder Rojas y sus empleados en Cedros y Sauces. A nivel metodológico, se hizo la revisión bibliográfica a la par que se evaluaban componentes reutilizables en otras arquitecturas. De esta manera, fue posible enlazar los requerimientos funcionales y no funcionales con la arquitectura desde su concepción.

### Requerimientos No Funcionales

En primer lugar, se presentan los drivers arquitectónicos, o requerimientos no funcionales. Estos son requerimientos orientados hacia las características del sistema que guían la manera en la que se diseña la arquitectura transversalmente. En la Tabla 1. Requerimientos No Funcionales, se presentan los drivers arquitectónicos abstraídos del estado del arte y tenidos en cuenta para SocialRobotBesa.

ID	Nombre	Explicación
RNF-01	Tiempo Real	El sistema debe responder en tiempo real a las necesidades del usuario y a los imprevistos presentados en el ambiente para manejar la imprevisibilidad de manera efectiva.
RNF-02	Extensibilidad	Dada la ubicuidad de la tecnología en la mayoría de los ambientes de interacción, el sistema debe poder incluir nuevos sensores o actuadores en cualquier momento.
RNF-03	Interoperabilidad	Teniendo en cuenta que un SAR no debe ser una tecnología excluyente, el sistema debe poder utilizar cualquier robot para llevar a cabo su función.

RNF-04	Recuperación de Errores	El sistema debe poder recuperarse de errores en el hardware y en el software en menos de 2 segundos, para no generar pérdida del interés del usuario.
RNF-05	Explicabilidad [1]	El sistema debe generar confianza a partir de la transparencia, para esto debe poder facilitar a los usuarios no técnicos, entender la toma de decisiones del robot.
RNF-06	Seguridad	El sistema debe mantener la información personal del usuario primario cifrada.
RNF-07	Autenticación	El sistema debe autenticar a los usuarios primarios, secundarios y terciarios para acceder a las funcionalidades del sistema.

Tabla 1. Requerimientos No Funcionales

### Requerimientos Funcionales

Respecto a los requerimientos funcionales, estos fueron agrupados y abstraídos a partir de la literatura y las entrevistas realizadas. En primera instancia, se tienen los requerimientos relacionados con la adaptabilidad. Estos requerimientos son de vital importancia ya que facilitan la efectividad de las intervenciones a largo plazo al brindar un mejor servicio al usuario, acorde a sus hábitos, estado emocional, características médicas y preferencias. Los requerimientos se presentan en la Tabla 2. Requerimientos de Adaptabilidad.

ID	Nombre	Explicación
ADP-1	Conciencia del contexto	El sistema debe poder reconocer riesgos relacionados al cambio de hábitos del usuario y notificar a los responsables.
ADP-2	Adaptación centrada en el humano	El sistema debe estar en la capacidad de mantener un modelo representativo del usuario y evolucionar sus características con el paso del tiempo
ADP-3	Aprendizaje Autónomo	El sistema debe poder modificar su manera de deliberar y actuar, a partir de la información capturada del ambiente y del usuario a lo largo del tiempo. Esta información puede ser explícita (retroalimentación directa) o implícita (señales no verbales, estado emocional).
ADP-4	Personalización	El sistema debe modificar las actividades y la deliberación a partir de las características

Tabla 2. Requerimientos de Adaptabilidad

Respecto a los requerimientos de empatía se presenta la Tabla 3. Requerimientos de Empatía. Con este grupo de requisitos, se espera mantener al usuario comprometido con la interacción. Para esto el robot debe mostrar señales de vida propias de una interacción real, ser consciente del estado emocional del usuario y además poder expresar su propio estado emocional.

ID	Nombre	Explicación
----	--------	-------------

EMP-1	Interacción Basada en la Personalidad	El sistema debe adaptar su personalidad a la personalidad del usuario. En específico, se hace relevante el nivel de extroversión.
EMP-2	Conciencia Emocional	El sistema debe poder interpretar las emociones del usuario desde una perspectiva multimodal.
EMP-3	Expresión Emocional	El sistema debe poder modificar su estado emocional a partir de eventos que ocurren en el ambiente y durante la interacción.
EMP-4	Demostrar Vida	El sistema debe demostrar su estado emocional mediante la utilización de sus recursos durante la interacción.

*Tabla 3. Requerimientos de Empatía*

La literatura sugiere que tanto los adultos mayores, como los niños prefieren que el robot haga parte del ecosistema tecnológico de su hogar. Teniendo esto en cuenta, el robot debe poder utilizar, colaborar y potencialmente cooperar con otros agentes y dispositivos. Los requerimientos relacionados con la cooperación tecnológica se presentan en la Tabla 4. Requerimientos de Cooperación Multi Agente.

ID	Nombre	Explicación
CMA-1	Integración Dinámica de Agentes	El sistema debe poder interactuar con otros agentes dentro del ecosistema tecnológico del usuario, para enriquecer los servicios provistos, el manejo de la imprevisibilidad y la interacción

*Tabla 4. Requerimientos de Cooperación Multi Agente*

El SAR tiene la oportunidad de mediar socialmente y promover la interacción social sana entre seres humanos. Este es un aspecto relevante en el cuidado de adultos mayores, y uno deseable en el contexto educativo. Los requerimientos de mediación social se muestran en la Tabla 5. Requerimientos de Mediación Social.

ID	Nombre	Explicación
MSC-1	Capacidad Multiusuario	El robot debe estar en la capacidad de soportar múltiples usuarios durante la realización de un servicio o bajo el marco de la interacción.
MSC-2	Cuidado socialmente inspirado	El sistema debe poder incluir los criterios de usuarios secundarios y terciarios, responsables del cuidado del usuario primario, a la hora de interactuar con el usuario primario.

*Tabla 5. Requerimientos de Mediación Social*

Si bien la seguridad es un requerimiento no funcional, el estado del arte sugiere la necesidad de manejar las excepciones del sistema con gracia. De esta manera, el SAR mantendría la confianza y la aceptabilidad de los usuarios. Los requerimientos de seguridad se muestran en la Tabla 6. Requerimientos de Seguridad.



ID	Nombre	Explicación
SAF-1	Manejo de Excepciones	El sistema debe reconocer la excepción ante el usuario y pedir perdón después de cierto tiempo.

*Tabla 6. Requerimientos de Seguridad*

La autonomía se presenta como un área de alto impacto en la gestión de la imprevisibilidad en los casos estudiados. El SAR debe poder entender el contexto de interacción y definir si tiene las autorizaciones y capacidades necesarias para realizar sus intenciones. Los requerimientos de autonomía se exponen en la Tabla 7. Requerimientos de Autonomía.

ID	Nombre	Explicación
AUT-1	Autonomía Ajustable	El sistema debe poder adaptar su autonomía según la criticalidad, responsabilidad y complejidad del entorno de la actividad.
AUT-2	Autonomía Autorizada	El sistema debe poder solicitar permiso al responsable en caso de que el nivel de autonomía sugerido sea intermedio.

*Tabla 7. Requerimientos de Autonomía*

La capacidad de planificar de manera ética y adaptativa exige que el SAR cuente con ciertas características en su etapa deliberativa. Para manejar la imprevisibilidad, el SAR debe estar en la capacidad de realizar múltiples metas paralelamente, teniendo en cuenta procesos de explicabilidad y su relación con el usuario bajo un marco de servicio específico. Estos requerimientos se pueden ver en la Tabla 8. Requerimientos de Toma de Decisiones

ID	Nombre	Explicación
TMD-1	Interacción Basada en Roles	El sistema debe poder identificar cuando un cambio de rol en el marco de una actividad debe ser realizado. Además, debe poder aplicarlo de manera oportuna.
TMD -2	Ética Embebida	El sistema debe poder identificar y mitigar riesgos éticos que puedan ocurrir bajo el marco de una interacción específica.
TMD-3	Transparencia/Explicabilidad	El sistema debe estar en la capacidad de comunicar su razonamiento a usuarios secundarios o terciarios involucrados en el marco de una interacción.
TMD-4	Intenciones Paralelas	El sistema debe poder realizar todas las intenciones que sean viables dado un momento específico. Esto podría generar una mejor interacción con el usuario.

*Tabla 8. Requerimientos de Toma de Decisiones*

Para fortalecer los procesos interactivos y la toma de decisiones, se requiere un manejo apropiado de los recursos del sistema. Si bien algunos de estos requerimientos no son abstraídos directamente de la literatura, capacidades como la expropiación de recursos son necesarias para implementar

requisitos como las intenciones paralelas. En la Tabla 9. Requerimientos de Manejo de Recursos, se describen los requerimientos asociados.

ID	Nombre	Explicación
MDR-1	Interacción Robusta	El sistema debe estar en la capacidad de utilizar sus sensores y otros agentes relacionados de manera multimodal. Asi mismo, el sistema debe poder enriquecer la interacción de manera multimodal utilizando todos los recursos apropiados para esto.
MDR -2	Interactividad	El robot debe poder mostrar movimientos dinámicos y reales dinámicamente para controlar la atención del usuario. Asi mismo, el robot debe poder generar procesos para mantener conversaciones creíbles y espontaneas.
MDR -3	Uso óptimo de Recursos	El sistema debe estar en la capacidad de expropiar los recursos utilizados por una intención, en caso de que un evento de mayor importancia deba ser atendido y estos sean requeridos.
MDR-4	Manejo del Tiempo	La naturaleza temporal de las intervenciones requiere que la arquitectura pueda manejar planes con una duración especifica, y manejar con gracia la finalización de estos. [2]

*Tabla 9. Requerimientos de Manejo de Recursos*

## Diseño General del SMA de SocialRobotBesa

Para diseñar el control inteligente de SocialRobotBesa, se tuvieron en cuenta 3 objetivos fundamentales. En primer lugar, se quería que el sistema fuera completo, en cuanto a la cobertura provista en los requisitos estudiados para el manejo de la imprevisibilidad humano robot en la interacción con SARs. En segundo lugar, se buscaba una arquitectura coherente que respetara los principios propuestos en la arquitectura BDI y su implementación en BDI-BESA. Por último, se priorizó el requisito no funcional de tener una arquitectura eficiente en el tiempo real que sea lo suficientemente reactiva para gestionar la imprevisibilidad de la interacción humano robot de manera oportuna.

La metodología AOPOA, de acuerdo con Bravo[3], es una metodología sistemática y procedimental basada en tres conceptos: descomposición jerárquica de roles, identificación de roles orientada a metas y la evolución de vínculos cooperativos entre roles. Para esta metodología, como sugieren Torres et.al [4], cada un sistema multiagentes es visto como una organización jerárquica, donde cada agente tiene un conjunto de habilidades, recursos y tareas para cumplir unos objetivos específicos sometidos al objetivo general del sistema. Para lograr el objetivo general del sistema, cada agente debe cooperar con los demás, sincronizar actividades y compartir recursos.

AOPOA se compone de dos fases iterativas: análisis y diseño. Durante el análisis, se detectan los roles y tareas del sistema a partir de los requerimientos funcionales y no funcionales. Posteriormente, se definen los objetivos del sistema teniendo en mente los requerimientos. Teniendo en cuenta los objetivos del sistema, se detectan las habilidades y recursos necesarios para conseguirlo mediante la implementación de tareas.

De igual manera, se abstraen roles a partir de las tareas identificadas y se itera progresivamente hasta llegar a roles con una granularidad media. Respecto a la interacción entre roles, se toman los roles encontrados y se definen relaciones de cooperación en conjunto con la información necesaria para cada vínculo. Una vez esto ha sido definido, se definen los flujos de la información de la cooperación entre roles y se agrupan los roles según múltiples criterios.

De acuerdo con Bravo[3], los criterios de agrupación pueden involucrar la similaridad de los objetivos, la distribución de habilidades o el acceso a recursos. Por último, una vez se han agrupado los roles en agentes, Angel et.al [5] mencionan que deben ser formalizados según una arquitectura interna. En el caso de este trabajo de grado, se tiene como objetivo general del sistema “Generar Bienestar para el usuario en ambientes de Imprevisibilidad”.

En primer lugar, se realizó una descomposición de metas del sistema. En este caso, se consideró que la meta general del sistema sería “Generar Bienestar para el adulto mayor en ambientes de Imprevisibilidad”. Posteriormente, se realizó un proceso iterativo para separar estas metas y generar un árbol con metas de granularidad media. El resultado de las metas se puede ver en la

# Meta	Meta	ID Submetas	Submetas
O1	Generar Bienestar para el usuario en ambientes de Imprevisibilidad	O1.1	Generar bienestar al adulto mayor
		O1.2	Mitigar la imprevisibilidad de la interacción
O1.1	Generar bienestar al adulto mayor	O1.1.1	Brindar Soporte Emocional al usuario
		O1.1.2	Relacionarse con otros Stakeholders
		O1.1.3	Adaptarse al contexto del usuario
		O1.1.4	Personalizar Interacción
		O1.1.5	Realizar Actividades con o sin supervisión
O1.1.1	Brindar Soporte Emocional al usuario	O1.1.1.1	Disminuir la Soledad
		O1.1.1.2	Mantener Emociones Placenteras
O1.1.1.1	Disminuir la Soledad	O1.1.1.1.1	Aumentar Interacción Social
		O1.1.1.1.2	Mejorar Apoyo Social
O1.1.1.2	Mantener Emociones Placenteras	O1.1.1.2.1	Demostrar Habilidades Sociales
		O1.1.1.2.2	Mantener estados emocionales positivos del usuario
O1.1.1.2.1	Demostrar Habilidades Sociales	O1.1.1.2.1.1	Demostrar Empatía
		O1.1.1.2.1.2	Demostrar Habilidades Conversacionales

		O1.1.1.2.1.3	Interactuar de Manera Enriquecida / multimodal
		O1.1.1.2.1.4	Expresar Señales de Vida
O1.1.1.2.2	Mantener estados emocionales positivos del usuario	O1.1.1.2.2.1	Conversar Empáticamente
		O1.1.1.2.2.2	Animar Usuario
O1.1.2	Relacionarse con otros Stakeholders	O1.1.2.1	Solicitar Intervención
		O1.1.2.2	Solicitar Información para interacción con usuario
		O1.1.2.3	Autenticar SH
		O1.1.2.4	Personalizar interacción con otros SH
		O1.1.2.5	Generar Reportes Periódicos
O1.1.3	Adaptarse al contexto del usuario	O1.1.3.1	Controlar un robot específico
		O1.1.3.2	Conectarse con el ambiente del usuario
O1.1.3.1	Controlar un robot específico	O1.1.3.1.1	Reconocer Recursos del robot
		O1.1.3.1.2	Adaptar las Habilidades del robot según los recursos disponibles
		O1.1.3.1.3	Adaptar Interacción según recursos del robot disponibles
O1.1.3.2	Conectarse con el ambiente del usuario	O1.1.3.2.1	Relacionarse con otros Agentes
		O1.1.3.2.2	Reconocer Recursos del ambiente
		O1.1.3.2.3	Integrarse con Nuevas Fuentes de Información
O1.1.4	Personalizar Interacción	O1.1.4.1	Evolucionar Perfil del Usuario
		O1.1.4.2	Personalizar Actividades, habilidades sociales, emociones
		O1.1.4.3	Adaptar Interfaces de interacción con usuario
O1.1.4.1	Personalizar Actividades, habilidades sociales, emociones	O1.1.4.1.1	Seleccionar la mejor actividad según el contexto emocional y social
		O1.1.4.1.2	Seleccionar la mejor actividad según el contexto de la interacción
O1.2	Mitigar la imprevisibilidad de la interacción	O1.2.1	Aprender sobre el Usuario
		O1.2.2	Asegurar Seguridad durante la Interacción
		O1.2.3	Tomar Decisiones inteligentemente
		O1.2.4	Gestionar los recursos eficientemente
O1.2.1	Aprender sobre el Usuario	O1.2.1.1	Reconocer cotidianidad del usuario
		O1.2.1.2	Aprender de la interacción con el usuario

O1.2.1.1	Reconocer cotidianidad del usuario	O1.2.1.1.1	Identificar Actividades
O1.2.1.1.1	Identificar Actividades	O1.2.1.1.1.1	Identificar Hábitos
		O1.2.1.1.1.2	Identificar Riesgo de Eventos Indeseables
		O1.2.1.1.1.3	Identificar Anomalías
O1.2.1.2	Aprender de la interacción con el usuario	O1.2.1.2.1	Generar Experimentos de Aprendizaje
		O1.2.1.2.2	Aprender de la interacción verbal
		O1.2.1.2.3	Reconocer Estado Emocional del Usuario
		O1.2.1.2.4	Procesar datos de forma multimodal
O1.2.1.2.1	Generar Experimentos de Aprendizaje	O1.2.1.2.1.1	Conocer Mejor al usuario
		O1.2.1.2.1.2	Adquirir Habilidades
O1.2.2	Asegurar Seguridad durante la Interacción	O1.2.2.1	Manejar Excepciones del sistema
		O1.2.2.2	Recuperarse de Errores del sistema
		O1.2.2.3	Asegurar la privacidad de la información
		O1.2.2.4	Reaccionar Oportunamente (Tiempo Real)
O1.2.2.1	Manejar Excepciones del sistema	O1.2.2.1.1	Revisar Estado de Sensores
		O1.2.2.1.2	Revisar Estado del sistema de cómputo y comunicación
		O1.2.2.1.3	Revisar Estado de Actuadores
O1.2.3	Tomar Decisiones inteligentemente	O1.2.3.1	Modular la ejecución de las decisiones
O1.2.3.1	Modular la ejecución de las decisiones	O1.2.3.2	Razonar Éticamente
		O1.2.3.3	Explicar proceso de Toma de Decisiones
		O1.2.3.4	Gestionar la persistencia y desistencia del robot
O1.2.3.1	Modular la ejecución de las decisiones	O1.2.3.1.1	Adaptar Personalidad
		O1.2.3.1.2	Adaptar el Nivel de Autonomía al contexto
		O1.2.3.1.3	Seleccionar el mejor rol para la tarea
O1.2.3.1.2	Adaptar el Nivel de Autonomía al contexto	O1.2.3.1.2.1	Adaptar autonomía en cuanto a toma de decisiones
		O1.2.3.1.2.2	Adaptar Autonomía en cuanto a la realización de acciones
O1.2.3.2	Razonar Éticamente	O1.2.3.2.1	Anticipar efectos negativos en la persona
		O1.2.3.2.2	Mitigar Efectos Éticos Negativos

		O1.2.3.2.3	Detectar Riesgos y efectos Éticos
O1.2.3.3	Explicar proceso de Toma de Decisiones	O1.2.3.3.1	Capturar Registros de sensores, estado y acciones del agente y del usuario
		O1.2.3.3.2	Sintetizar reporte
O1.2.3.4	Gestionar la persistencia y desistencia del robot	O1.2.3.4.1	Gestionar el tiempo de las intervenciones
O1.2.4	Gestionar los recursos eficientemente	O1.2.4.1	Expropiar recursos
		O1.2.4.2	Compartir recursos

Tabla 10. Metas del SMA para la arquitectura de control SocialRobotBesa. Gráficamente, la descomposición de metas se puede ver en la Ilustración 1. Descomposición de Metas dividido en las categorías de requerimientos presentadas en la **¡Error! No se encuentra el origen de la referencia..**

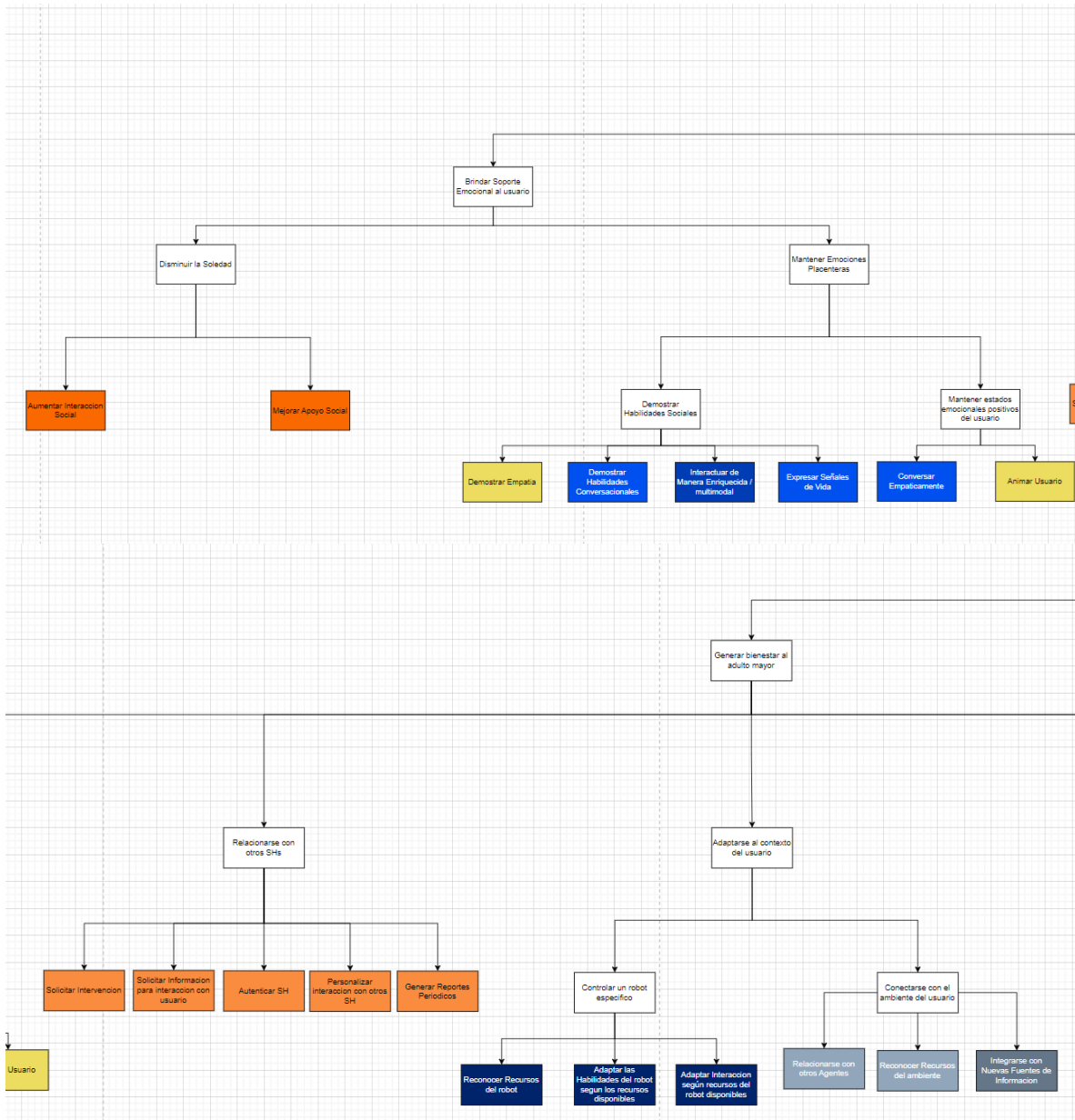
# Meta	Meta	ID Submetas	Submetas
O1	Generar Bienestar para el usuario en ambientes de Imprevisibilidad	O1.1	Generar bienestar al adulto mayor
		O1.2	Mitigar la imprevisibilidad de la interacción
O1.1	Generar bienestar al adulto mayor	O1.1.1	Brindar Soporte Emocional al usuario
		O1.1.2	Relacionarse con otros Stakeholders
		O1.1.3	Adaptarse al contexto del usuario
		O1.1.4	Personalizar Interacción
		O1.1.5	Realizar Actividades con o sin supervisión
O1.1.1	Brindar Soporte Emocional al usuario	O1.1.1.1	Disminuir la Soledad
		O1.1.1.2	Mantener Emociones Placenteras
O1.1.1.1	Disminuir la Soledad	O1.1.1.1.1	Aumentar Interacción Social
		O1.1.1.1.2	Mejorar Apoyo Social
O1.1.1.2	Mantener Emociones Placenteras	O1.1.1.2.1	Demostrar Habilidades Sociales
		O1.1.1.2.2	Mantener estados emocionales positivos del usuario
O1.1.1.2.1	Demostrar Habilidades Sociales	O1.1.1.2.1.1	Demostrar Empatía
		O1.1.1.2.1.2	Demostrar Habilidades Conversacionales
		O1.1.1.2.1.3	Interactuar de Manera Enriquecida / multimodal
		O1.1.1.2.1.4	Expresar Señales de Vida
O1.1.1.2.2	Mantener estados emocionales positivos del usuario	O1.1.1.2.2.1	Conversar Empáticamente
		O1.1.1.2.2.2	Animar Usuario
O1.1.2	Relacionarse con otros Stakeholders	O1.1.2.1	Solicitar Intervención
		O1.1.2.2	Solicitar Información para interacción con usuario

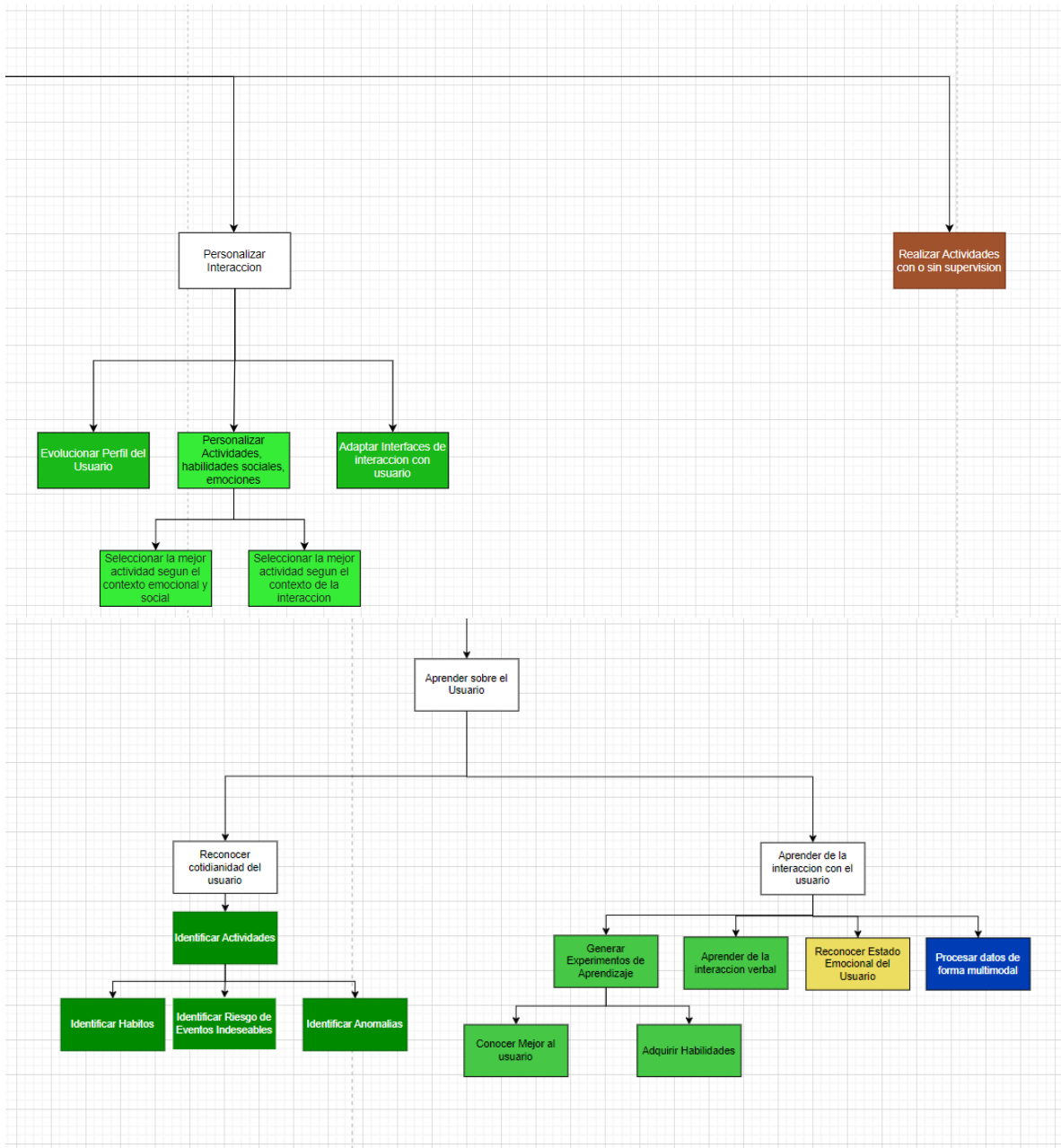
		O1.1.2.3	Autenticar SH
		O1.1.2.4	Personalizar interacción con otros SH
		O1.1.2.5	Generar Reportes Periódicos
O1.1.3	Adaptarse al contexto del usuario	O1.1.3.1	Controlar un robot específico
		O1.1.3.2	Conectarse con el ambiente del usuario
O1.1.3.1	Controlar un robot específico	O1.1.3.1.1	Reconocer Recursos del robot
		O1.1.3.1.2	Adaptar las Habilidades del robot según los recursos disponibles
		O1.1.3.1.3	Adaptar Interacción según recursos del robot disponibles
O1.1.3.2	Conectarse con el ambiente del usuario	O1.1.3.2.1	Relacionarse con otros Agentes
		O1.1.3.2.2	Reconocer Recursos del ambiente
		O1.1.3.2.3	Integrarse con Nuevas Fuentes de Información
O1.1.4	Personalizar Interacción	O1.1.4.1	Evolucionar Perfil del Usuario
		O1.1.4.2	Personalizar Actividades, habilidades sociales, emociones
		O1.1.4.3	Adaptar Interfaces de interacción con usuario
O1.1.4.1	Personalizar Actividades, habilidades sociales, emociones	O1.1.4.1.1	Seleccionar la mejor actividad según el contexto emocional y social
		O1.1.4.1.2	Seleccionar la mejor actividad según el contexto de la interacción
O1.2	Mitigar la imprevisibilidad de la interacción	O1.2.1	Aprender sobre el Usuario
		O1.2.2	Asegurar Seguridad durante la Interacción
		O1.2.3	Tomar Decisiones inteligentemente
		O1.2.4	Gestionar los recursos eficientemente
O1.2.1	Aprender sobre el Usuario	O1.2.1.1	Reconocer cotidianidad del usuario
		O1.2.1.2	Aprender de la interacción con el usuario
O1.2.1.1	Reconocer cotidianidad del usuario	O1.2.1.1.1	Identificar Actividades
O1.2.1.1.1	Identificar Actividades	O1.2.1.1.1.1	Identificar Hábitos
		O1.2.1.1.1.2	Identificar Riesgo de Eventos Indeseables
		O1.2.1.1.1.3	Identificar Anomalías
O1.2.1.2	Aprender de la interacción con el usuario	O1.2.1.2.1	Generar Experimentos de Aprendizaje
		O1.2.1.2.2	Aprender de la interacción verbal

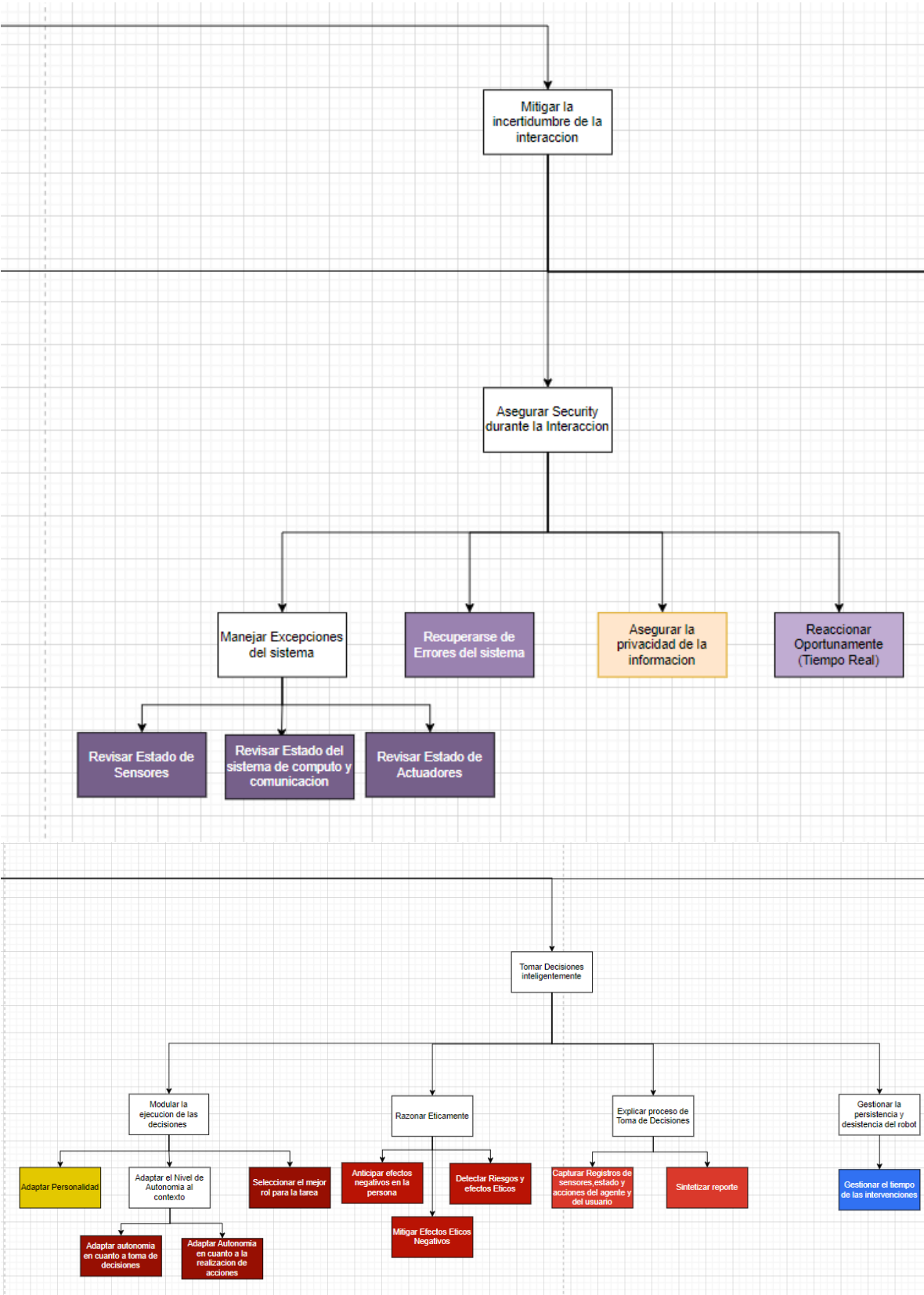
		O1.2.1.2.3	Reconocer Estado Emocional del Usuario
		O1.2.1.2.4	Procesar datos de forma multimodal
O1.2.1.2.1	Generar Experimentos de Aprendizaje	O1.2.1.2.1.1	Conocer Mejor al usuario
		O1.2.1.2.1.2	Adquirir Habilidades
O1.2.2	Asegurar Seguridad durante la Interacción	O1.2.2.1	Manejar Excepciones del sistema
		O1.2.2.2	Recuperarse de Errores del sistema
		O1.2.2.3	Asegurar la privacidad de la información
		O1.2.2.4	Reaccionar Oportunamente (Tiempo Real)
O1.2.2.1	Manejar Excepciones del sistema	O1.2.2.1.1	Revisar Estado de Sensores
		O1.2.2.1.2	Revisar Estado del sistema de cómputo y comunicación
		O1.2.2.1.3	Revisar Estado de Actuadores
O1.2.3	Tomar Decisiones inteligentemente	O1.2.3.1	Modular la ejecución de las decisiones
O1.2.3.1	Modular la ejecución de las decisiones	O1.2.3.2	Razonar Éticamente
		O1.2.3.3	Explicar proceso de Toma de Decisiones
		O1.2.3.4	Gestionar la persistencia y desistencia del robot
O1.2.3.1	Modular la ejecución de las decisiones	O1.2.3.1.1	Adaptar Personalidad
		O1.2.3.1.2	Adaptar el Nivel de Autonomía al contexto
		O1.2.3.1.3	Seleccionar el mejor rol para la tarea
O1.2.3.1.2	Adaptar el Nivel de Autonomía al contexto	O1.2.3.1.2.1	Adaptar autonomía en cuanto a toma de decisiones
		O1.2.3.1.2.2	Adaptar Autonomía en cuanto a la realización de acciones
O1.2.3.2	Razonar Éticamente	O1.2.3.2.1	Anticipar efectos negativos en la persona
		O1.2.3.2.2	Mitigar Efectos Éticos Negativos
		O1.2.3.2.3	Detectar Riesgos y efectos Éticos
O1.2.3.3	Explicar proceso de Toma de Decisiones	O1.2.3.3.1	Capturar Registros de sensores, estado y acciones del agente y del usuario
		O1.2.3.3.2	Sintetizar reporte
O1.2.3.4	Gestionar la persistencia y desistencia del robot	O1.2.3.4.1	Gestionar el tiempo de las intervenciones
O1.2.4	Gestionar los recursos eficientemente	O1.2.4.1	Expropiar recursos
		O1.2.4.2	Compartir recursos



Tabla 10. Metas del SMA para la arquitectura de control SocialRobotBesa







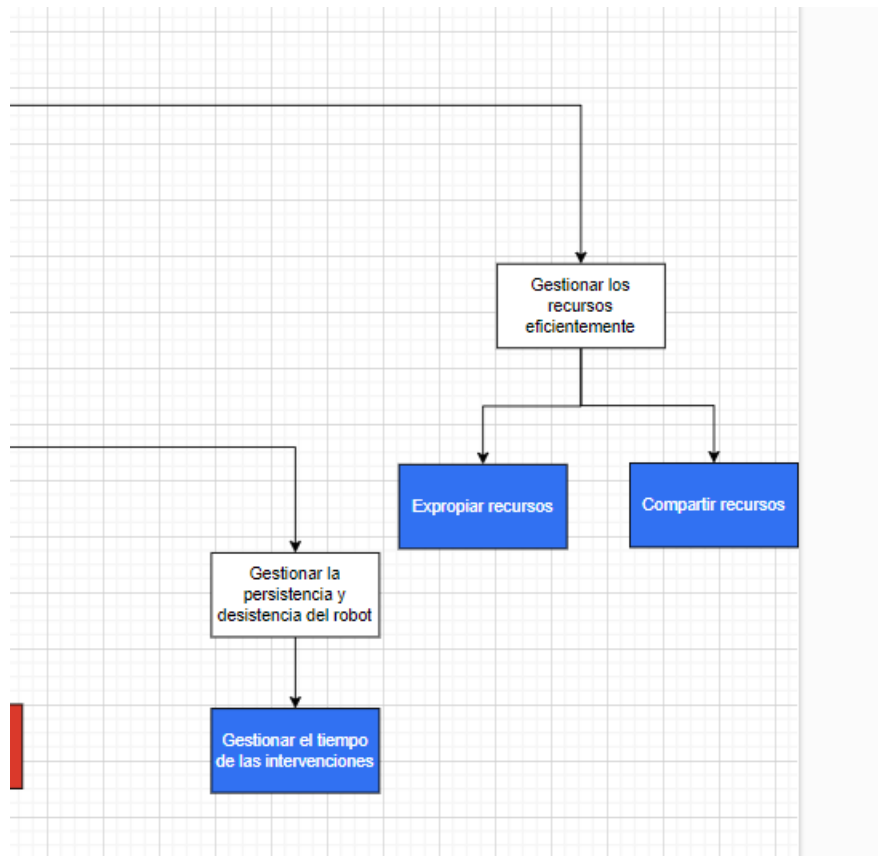


Ilustración 1. Descomposición de Metas

Para cada meta estudiada, se abstrajeron recursos y habilidades necesarios para cumplir con las metas del sistema. Estas son presentadas en la Tabla 11. Recursos y Habilidades

Tipo	ID	Descripción
Recurso	R1	Video
Recurso	R2	Voz
Recurso	R3	Actuadores
Recurso	R4	Micrófono
Recurso	R5	Pantalla
Recurso	R6	Perfiles de los usuarios
Recurso	R7	Perfil del usuario
Recurso	R8	Leds
Recurso	R9	Modelo de emociones
Recurso	R10	Brazos
Recurso	R11	Piernas
Recurso	R12	Ojos
Recurso	R13	Modelo emocional del usuario
Recurso	R14	Micrófonos
Recurso	R15	Reproductor de sonido
Recurso	R16	Información de capacidades/recursos del robot

Recurso	R17	Música externa
Recurso	R18	Información sobre los sensores del robot
Recurso	R19	Conocimiento de habilidades del robot
Recurso	R20	Contexto de la interacción
Recurso	R21	Información sobre otros agentes
Recurso	R22	Canal de comunicación
Recurso	R23	Otras entradas de datos
Recurso	R24	Información sobre las fuentes de información
Recurso	R25	Información sobre la interacción con el usuario
Recurso	R26	Video (información emocional)
Recurso	R27	Modelo de voz del usuario
Recurso	R28	Sensores adicionales identificados
Recurso	R29	Oídos
Recurso	R30	Perfil de Stakeholders
Recurso	R31	Información de contacto de los Stakeholders
Recurso	R32	Perfil del Stakeholders
Recurso	R33	Información sobre recursos (internos externos) disponibles
Recurso	R34	Contexto de la interacción
Recurso	R35	Modelo de hábitos
Recurso	R36	Modelo de situaciones riesgosas
Recurso	R37	Sensores
Recurso	R38	Modelo de actividades
Recurso	R39	Contexto de hábitos del usuario a nivel temporal
Recurso	R40	Modelo de actividades indeseables
Recurso	R41	Información proveniente de sensores
Recurso	R42	Información sobre la actividad a realizar
Recurso	R43	Modelo emocional
Recurso	R44	Información sobre implicaciones éticas de la actividad
Recurso	R45	Flujo de información sobre el estado de los sensores
Recurso	R46	Información del contexto de la interacción y el usuario
Recurso	R47	Información capturada sobre sensores
Recurso	R48	Información temporal sobre la actividad
Recurso	R49	Información sobre recursos disponibles y en uso
Recurso	R50	Información sobre estado de actuadores y sensores
Recurso	R51	Información sobre la toma de decisiones
Habilidad	A1	Hablar
Habilidad	A2	Mostrar emociones
Habilidad	A3	Balancear necesidades de usuarios
Habilidad	A4	Detectar emociones
Habilidad	A5	Seguir al usuario
Habilidad	A6	Identificar estado del juego
Habilidad	A7	Hacer filtrado colaborativo
Habilidad	A8	Identificar emociones

Habilidad	A9	Realizar video llamadas
Habilidad	A10	Mostrar interés
Habilidad	A11	Entender sentimientos del usuario
Habilidad	A12	Escuchar
Habilidad	A13	Mostrar alegría
Habilidad	A14	Mostrar tristeza
Habilidad	A15	Mostrar ira
Habilidad	A16	Procesamiento de video para detectar emociones en un periodo de tiempo
Habilidad	A17	Responder preguntas
Habilidad	A18	Hablar contextualmente
Habilidad	A19	Mostrar texto
Habilidad	A20	Responder coherentemente la conversación
Habilidad	A21	Hacer sonidos
Habilidad	A22	Mover los actuadores
Habilidad	A23	Identificar ira
Habilidad	A24	Procesar datos adicionales
Habilidad	A25	Identificar capacidades para realizar la tarea
Habilidad	A26	Contactar al SH
Habilidad	A27	Hacer preguntas
Habilidad	A28	Entender las respuestas
Habilidad	A29	Enviar mensajes por WhatsApp
Habilidad	A30	Enviar mensaje de texto
Habilidad	A31	Enviar correo
Habilidad	A32	Generar archivos
Habilidad	A33	Integrarse a los elementos dispuestos en el hardware
Habilidad	A34	Comunicarse con otros agentes
Habilidad	A35	Cooperar con otros agentes
Habilidad	A36	Procesar video para identificar objetos
Habilidad	A37	Entender los eventos enviados por otros agentes
Habilidad	A38	Aprender de la interacción
Habilidad	A39	Recibir retroalimentación
Habilidad	A40	Conversar con el usuario
Habilidad	A41	Modular todos los actuadores disponibles
Habilidad	A42	Adaptar actuadores (gestionar volumen)
Habilidad	A43	Procesar emociones del usuario en un marco temporal
Habilidad	A44	Procesar video e información adicional
Habilidad	A45	Mostrar media
Habilidad	A46	Reproducir sonido
Habilidad	A47	Reconocer actividades
Habilidad	A48	Inferir actividades que no son habituales
Habilidad	A49	Generar experimentos seguros
Habilidad	A50	Reconocer situaciones donde se requiere más información del usuario
Habilidad	A51	Reconocer situaciones donde se requiere más habilidad

Habilidad	A52	Identificar información importante en lo que dice el usuario
Habilidad	A53	Procesar texto
Habilidad	A54	Voz a Texto
Habilidad	A55	Identificar error de los sensores
Habilidad	A56	Gestionar excepción
Habilidad	A57	Identificar uso de recursos computacionales severo
Habilidad	A58	Identificar error en los actuadores
Habilidad	A59	Identificar posible colisión o daño en los mismos
Habilidad	A60	Encriptar la información
Habilidad	A61	Procesar la información en tiempo casi real
Habilidad	A62	Cambiar nivel de extroversión/introversión
Habilidad	A63	Determinar la criticidad de la actividad
Habilidad	A64	Determinar si se tienen los recursos/habilidades suficientes para la actividad
Habilidad	A65	Determinar el estado de salud del usuario
Habilidad	A66	Pedir autorización a un SH autorizado
Habilidad	A67	Cambiar de rol
Habilidad	A68	Calcular el rol según las necesidades del contexto y actividad
Habilidad	A69	Predecir riesgos dada una tarea
Habilidad	A70	Cambiar el actuar para reducir el riesgo a corto y largo plazo
Habilidad	A71	Capturar datos de manera contextual
Habilidad	A72	Procesar datos
Habilidad	A73	Registrar datos
Habilidad	A74	Dar sentido de negocio a los datos
Habilidad	A75	Expresar reporte al SH
Habilidad	A76	Tener noción del tiempo
Habilidad	A77	Conocer cuando los recursos ya no son útiles o necesarios
Habilidad	A78	Parar la utilización de un recurso correctamente
Habilidad	A79	Acceder paralelamente a flujos de datos de un mismo recurso

*Tabla 11. Recursos y Habilidades*

Haciendo uso de las metas, los recursos y las habilidades se generan roles. Cada rol presenta un grupo de tareas o funcionalidades conjuntas del sistema multiagentes. Los roles y los agentes relacionados se muestran en la Tabla 12. Roles Identificados.

Rol	Descripción	Agente Responsable
Rol Procesador de Información	Procesa la información de los sensores para alimentar las creencias del agente	Agente Proveedor de Servicios
Rol Multi Usuario	Gestiona paralelamente múltiples interacciones con usuarios durante el mismo marco de actividad.	Agente de Motivación, Agente Interprete Emocional del Usuario, Agente de Creencias, Agente Interactivo

Rol Interactivo	Se comunica con el usuario utilizando los recursos de manera óptima. Se encarga de procesar los eventos de los actuadores y utilizarlos para el bienestar del usuario. Se encarga de conversar con el usuario de manera coherente. Genera movimientos coherentes	Agente Interactivo
Rol Modulador	Enriquece la interacción según las configuraciones dadas (modelo E, recursos, criterios de adaptación + personalización de interfaces, rol en cuanto a autoridad)	Agente de Acción
Rol Emocional	Gestiona el modelo emocional y personalidad del robot a partir de la generación de eventos emocionales a partir de la interacción	Agente Interprete Emocional, Agente de Creencias
Rol Ético	Identifica y mitiga riesgos éticos a largo plazo que ocurren durante la intervención. Las variables que revisa son a largo plazo	Agente de Riesgos Éticos, Agente de Acción
Rol Empático	Identifica las emociones del usuario a partir de la información entrante	Agente Interprete Emocional del Usuario
Rol Personalización	Determina la mejor manera de realizar actividades/procesos en términos de contenido y selección de actividades	Agente de Acción. Agente de Motivación
Rol Adaptador	Gestiona y evolución el perfil del usuario para generar bienestar, a partir de la interacción	Agente de Aprendizaje, Agente Interprete Emocional del Usuario
Rol Experimentador	A partir de la evaluación del desempeño de la interacción con el usuario y del estado del ambiente y de sí mismo, genera experimentos para conseguir recursos o mejorar habilidades	Agente de Aprendizaje
Rol de Excepción	Diagnostica y Evalúa el estado de sensores, actuadores y recursos de cómputo para gestionar posibles excepciones en tiempo real	Agente proveedor de servicios, Agente de motivación
Rol de Error	Detecta y recupera el sistema de errores inesperados.	Agente proveedor de servicios, Agente de motivación
Rol de Andamiaje	Gestiona el rol y su cambio durante la interacción con el usuario. Se encarga de seleccionar las estrategias correctas o parametrizar la manera de realizar una actividad según la dinámica usuario-robot.	Agente Interprete Emocional del Usuario, Agente de motivación
Rol de autonomía	Determina el nivel de autonomía para realizar una acción a partir de los recursos, habilidades y criterios de interacción con el usuario. Incluye si se puede hacer o si debo pedir permiso para hacerlo	Agente de Motivación



Rol Explicativo	Captura toda la información relacionada a la toma de decisiones en el sistema para sintetizar reportes de toma de decisiones	Agente de Explicabilidad, Agente de Motivación
Rol Monitor de tiempo	Gestiona el transcurso de las actividades a través del tiempo para facilitar un manejo correcto de los recursos	Agente de Motivación
Rol Gestor de Recursos	Administra la utilización de los recursos según las actividades que deban ser realizadas por el sistema, esto incluye la expropiación y la paralelización de estos.	Agente de Motivación
Rol Cooperación Humanos	Coopera, Coordina, Compite con los demás humanos relacionados al usuario para generar bienestar	Agente de Cooperación con Humanos, Agente Interactivo
Rol Cooperación Agentes	Coopera, Coordina, Compite con los demás agentes/entradas de información relacionados al usuario para generar bienestar	Agente de Cooperación Con Agentes
Rol Monitor de hábitos	Aprende, gestiona y analiza los hábitos del usuario	Agente de Riesgos
Rol Activitas	Realiza las actividades de los ejes de bienestar	Agente de Motivación

Tabla 12. Roles Identificados

Por último, se presentan los lazos de cooperación entre los roles que determinan los procesos y comportamientos esperados del SMA. Estos se pueden ver en la Ilustración 2. Roles de Interacción SocialRobotBesa.

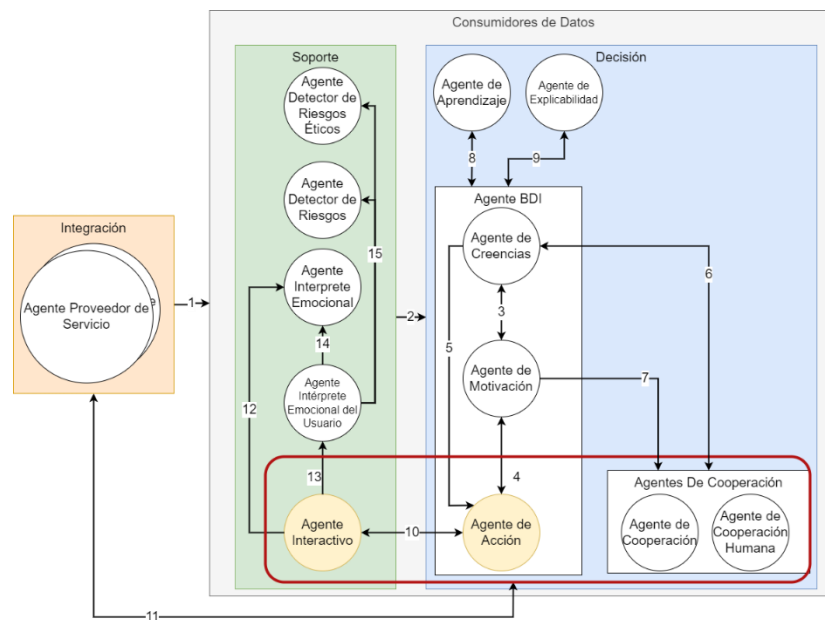


Ilustración 2. Roles de Interacción SocialRobotBesa

Una vez identificados los roles y sus procesos de cooperación, es posible agrupar los roles en agentes. En el caso de SocialRobotBesa, los agentes identificados se encuentran en la Tabla 13. Agentes de SocialRobotBesa.

<b>Agente</b>	<b>Propósito</b>
<b>Agente Proveedor de Servicios</b>	Los agentes proveedores de servicio tienen como función traducir las peticiones de servicio de los demás agentes, de manera que puedan ser entendida por el robot. Estos agentes hacen de capa de traducción y pueden tener modelos de inteligencia artificial para procesar datos específicos de la interacción.
<b>Agente Detector de Riesgo</b>	Utilizar la información proveniente de los sensores del sistema y aquella presente en las creencias del agente BDI, para detectar riesgos relacionados con el cambio de hábitos, accidentes espontáneos e incluso riesgos relacionados a afecciones medicas como la hipertensión, la ansiedad o la depresión. En caso de detectar un riesgo este agente informará al agente BDI, quien activará un plan relacionado con su mitigación.
<b>Agente Detector de Riesgos Éticos</b>	Identificar riesgos éticos presentados en la literatura [6], [7] como lo pueden ser la sobre confianza, el apego o la soledad, a partir de la información provista por los sensores del sistema y aquella presente en las creencias del agente BDI. En caso de detectar un riesgo este agente informará al agente BDI, quien activará un plan relacionado con su mitigación.
<b>Agente Interprete Emocional</b>	Utilizar la información resultante de la interacción y de eventos propios del ambiente para crear eventos emocionales basados en el modelo OCC [8] para afectar el estado emocional del robot. El agente interpreta a partir de la información recibida el tipo de evento, el sujeto asociado al evento y el objeto/contexto en el evento. Este agente permite expresar emociones correctamente y generar empatía con el usuario.
<b>Agente Interprete Emocional del Usuario</b>	Hacer de agente de fusión de decisión para los múltiples eventos que indican la emoción del usuario durante la interacción. Se encarga de disminuir la imprevisibilidad resultante de los sensores con relación al estado emocional del usuario, que es vital para tomar decisiones correctamente en el sistema. Este agente es adaptado del trabajo de Alcalá [9].
<b>Agente de Creencias</b>	Agente encargado de almacenar y transmitir la información relevante relacionada con el estado del robot, el estado de la interacción, el perfil del usuario, los riesgos éticos y el modelo del mundo.
<b>Agente De Motivación</b>	Agente encargado del proceso de toma de decisiones, teniendo en cuenta recursos, tiempo y autonomía. Se encarga así mismo de iniciar los procesos de deliberación, acción y expropiación de la arquitectura.
<b>Agente de Acción</b>	Tiene como función el enriquecer las acciones relacionadas a tareas y traducirlas en primitivas. Así mismo, el agente de acción se encarga de coordinar las peticiones de servicio hacia los agentes proveedores de servicio y de transmitir los eventos de expropiación.

*Tabla 13. Agentes de SocialRobotBesa*

Una vez los agentes de SocialRobotBesa han sido definidos, se procede a definir cada agente de manera formal. Para esto, se sigue la metodología de desarrollo incremental. De acuerdo con esta metodología, los requisitos de la arquitectura se dividen en múltiples grupos e iterativamente cada grupo de requerimientos es agregado a la arquitectura hasta que esta sea finalizada. Como resultado

de la arquitectura iterativa se tiene la Ilustración 3. Arquitectura SocialRobotBesa. Esta se puede ver completa en la imagen Arquitectura en la misma carpeta del anexo.

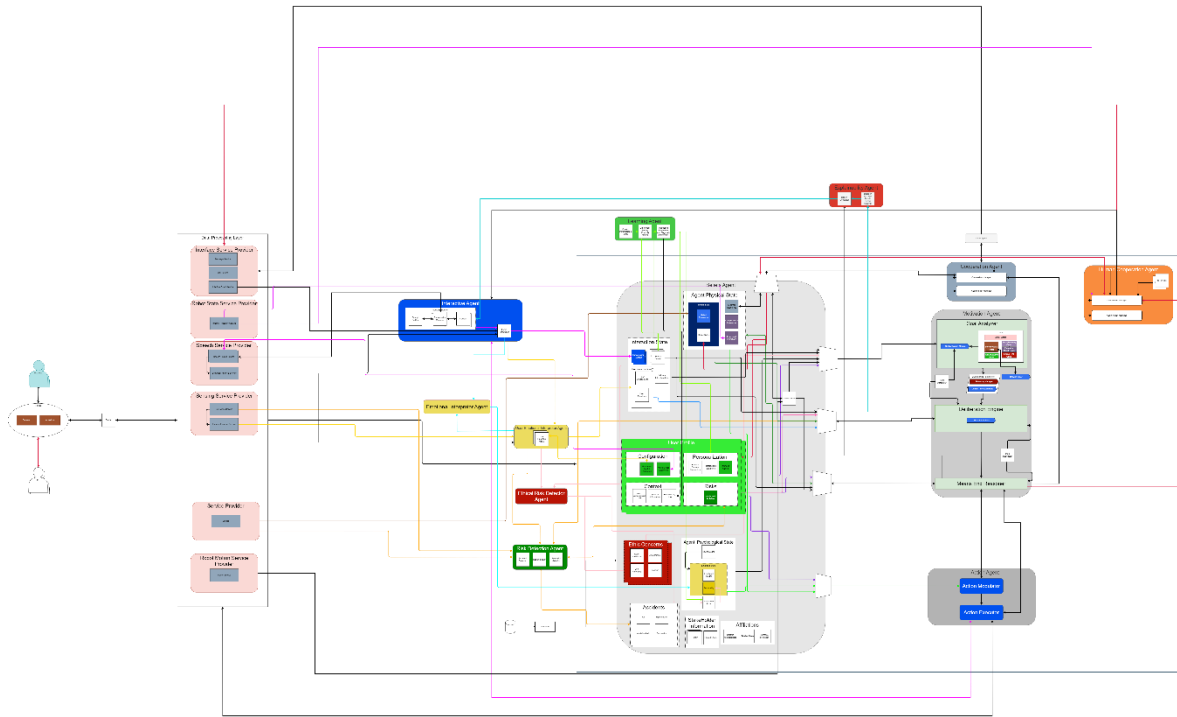


Ilustración 3. Arquitectura SocialRobotBesa

Como es posible evidenciar, la arquitectura consta de 11 tipos de agentes diferentes que pueden ser instanciados de diferentes maneras según las necesidades del contexto de aplicación. A continuación, se presentará cada tipo de agente de manera formal, en conjunto con sus interacciones con otros agentes y la información que requiere para funcionar correctamente. Cabe resaltar que los agentes están divididos acorde a los modelos presentados en la Ilustración 2. Roles de Interacción SocialRobotBesa, es decir el módulo de integración, el módulo de soporte y el módulo de decisión.

### Módulo de Integración

El módulo de integración tiene como objetivo integrar diferentes tecnologías y brindar capacidades de extensibilidad e interoperabilidad al sistema a de control inteligente. Este módulo se compone únicamente de agentes *Proveedor de Servicio*, los cuales dependen estrechamente del caso de aplicación donde se va a instanciar la arquitectura.

### Agente Proveedor de Servicio

El agente *Proveedor de Servicio* se compone de diferentes servicios, los cuales son registrados en un componente llamado *Registro de Servicios*, el cual brinda desacoplamiento entre proveedores y consumidores en la arquitectura al ocultar los detalles de implementación. Las creencias del agente se pueden representar como una tupla:

$$(Serv, PM, PrS)$$

Donde:

- *Serv* representa el conjunto de servicios provistos por el agente.
- *PM* representa la relación entre primitivas del sistema y primitivas del robot
- *PrS* representa el estado de las primitivas solicitadas

Este agente cumple con 5 responsabilidades fundamentales dentro de la arquitectura de control inteligente. En primer lugar, este agente se encarga de gestionar la integración con componentes externos, así como de implementar la interfaz de integración con el robot. En segundo lugar, este agente se encarga de implementar módulos con capacidades de procesamiento de datos como lo pueden ser Voz a Texto, Análisis de Sentimiento e incluso detección de objetos. En tercer lugar, el agente *Proveedor de Servicio* se encarga de manejar la lógica relacionada con la confirmación de primitivas una vez estas han sido finalizadas por el robot. En cuarto lugar, este agente implementa la lógica de expropiación a nivel de primitiva al mantener el estado de las primitivas en ejecución y por tanto al recibir un evento de expropiación está en la capacidad de pedirle al robot que la detenga. Por último el agente *Proveedor de Servicio*, funciona como punto de entrada de eventos externos al sistema que pueden venir de múltiples fuentes como el robot o sensores adicionales.

La arquitectura mínima de un agente *Proveedor de Servicio* se muestra en la Ilustración 4. Agente Proveedor de Servicio. Como se puede ver, un agente *Proveedor de Servicio* consta de al menos un servicio y su adaptador respectivo. Cabe mencionar que cada servicio debe tener obligatoriamente un adaptador para gestionar la transformación de la información que llega del robot o que va dirigida hacia este.

De igual manera cada servicio representa una interfaz uniforme, la cual puede ser extendida para agregar nuevos servicios ofrecidos por un mismo proveedor. El adaptador debe implementar la traducción para cada servicio provisto por la interfaz.

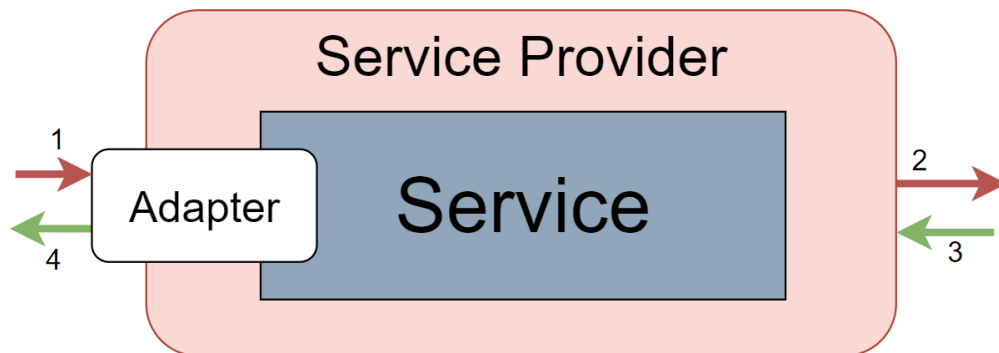


Ilustración 4. Agente Proveedor de Servicio

Este agente tiene consta de 4 interacciones. En primer lugar la representada en la flecha 1, muestra el flujo de eventos del robot hacia el agente *Proveedor de Servicio*. Esta interacción se representa con la tupla  $(id_x, re_x)$ , donde  $id_x$  representa el identificador del evento y  $re_x$  representa la información asociada al evento.  $id_x$  puede contener un entero representando la respuesta exitosa de un servicio solicitado previamente o un -1 para indicar que es un evento sin petición.

La flecha 2 representa el flujo de eventos internos del *Proveedor de Servicio* hacia agentes en otros módulos de la arquitectura de control. Esta salida se presenta con la tupla  $(id_x, be_x)$ , donde  $id_x$

representa el identificador del evento y  $be_x$  representa la información asociada al evento una vez traducida por el adaptador.

La flecha 3 representa de peticiones de servicio hacia el agente *Proveedor de Servicio*. Esta interacción se presenta con la tupla  $(id_x, req_x)$ , donde  $id_x$  representa el identificador de la petición de servicio y  $req_x$  representa la información asociada a la primitiva del servicio a solicitar.

La flecha 4 representa la petición de servicio hacia el robot una vez transformada por el agente *Proveedor de Servicio*. Esta interacción se presenta con la tupla  $(id_x, rreq_x)$ , donde  $id_x$  representa el identificador de la petición de servicio y  $rreq_x$  representa la información transformada de la primitiva del servicio a solicitar.

### Módulo de Soporte

El módulo de soporte tiene como objetivo proveer información relevante para el caso de aplicación derivada de información proveniente del módulo de integración. Así mismo, algunos agentes que componen este módulo pueden comunicarse directamente con el módulo de integración para solicitar servicios de manera reactiva. Este módulo se compone de 5 agentes.

### Agente Interactivo

El agente interactivo tiene el objetivo de gestionar y reaccionar la interacción no estructurada. Su función dentro del sistema es de vital importancia ya que permite gestionar la imprevisibilidad causada por la interacción verbal y enriquece el sistema al proveer conversaciones creíbles. En este caso esto hace referencia a eventos verbales, eventos táctiles y eventos de interfaces como tabletas, que se den durante la interacción con el humano. Para esto el agente podría valerse de modelos de lenguaje natural grandes como Chat-GPT. Este agente también tiene como función retransmitir los eventos verbales o táctiles que sean de naturaleza emocional en la interacción como lo puede ser un “Te quiero” o un abrazo, de manera que estos puedan afectar el estado emocional del robot. El agente interactivo se compone de cuatro componentes como se muestra en la Ilustración 5. Agente Interactivo. La arquitectura y funcionalidad de este agente fue inspirada en los trabajos de varios autores [10] [11] [12].

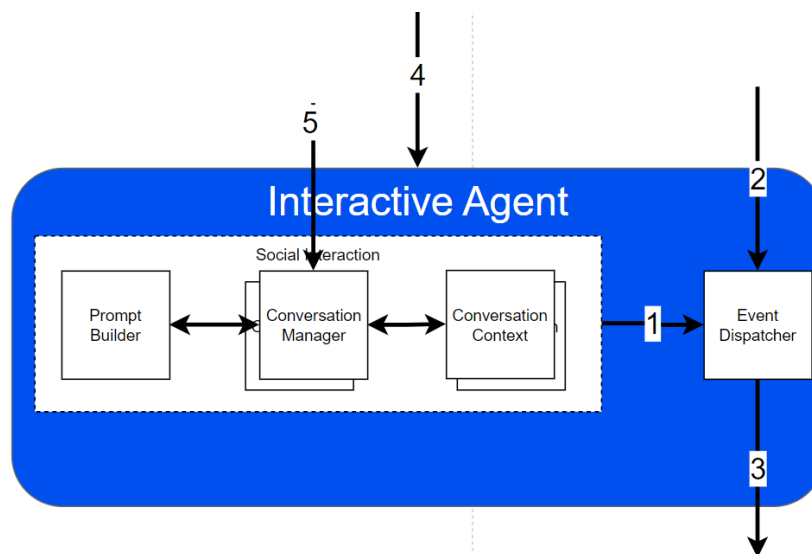


Ilustración 5. Agente Interactivo

En primer lugar se tiene el *Prompt Builder*. Este componente está encargado de construir instrucciones para alimentar a los modelos de lenguaje natural encargados de la conversación. Dada una tupla  $(t, c, u)$ , donde  $t$  es el tipo de instrucción,  $c$  es el concepto relacionado y  $u$  es el identificador del usuario relacionado, el componente genera una instrucción  $I_{t,c,u}$  que será utilizada por el componente *Conversation Manager*.

El *Conversation Manager* tiene el objetivo de integrar al agente con un LLM y de gestionar las instrucciones  $I_{t,c,u}$  enviadas por el *Prompt Builder*. Para esto, también se vale del contexto de la conversación almacenado en el *Conversation Context*.

Respecto al *Conversation Context*, este hace la función de repositorio de eventos y respuestas generadas por el *Conversation Manager* y recibidas del Módulo de Integración. Estas respuestas se representan por el conjunto  $R_u$ , donde  $u$  es identificador del usuario relacionado.

Por otro lado, el componente *Event Dispatcher* se encarga de traducir un evento interactivo de un usuario  $ei_u$  en un evento relevante para el sistema. La llegada de los eventos  $ei_u$  se da a través de la línea 1, si el evento es ocasionado por el dialogo. Por otro lado, el evento  $ei_u$  puede ser generado mediante la línea 2 provisto por agentes *Proveedores de Servicio* u otro tipo de fuente no verbal. Los posibles eventos generados se muestran en la Tabla 14. Eventos del Despachador de Eventos. Cabe resaltar que los eventos generados salen por la línea 3 hacia los agentes indicados en la columna Agente Receptor.

Evento	Descripción	Agente Receptor
$em_u$	Evento emocional del usuario, puede ser ocasionado por las emociones expresadas por el usuario a partir de su dialogo. Un evento $em_u$ , se puede representar como una tupla $em_u = (E_u, u)$ . $E_u$ es un conjunto de emociones del usuario con sus valencias.	Agente <i>Interprete Emocional del Usuario</i>
$emir$	Evento emocional interactivo del robot, puede ser ocasionado por palabras como “te quiero”, “no quiero estar aquí” o por eventos relacionados con el tacto como abrazos o golpes. Un evento $emir$ , se puede representar como una tupla $emir = (tie)$ donde $tie$ es el tipo de evento	Agente <i>Interprete Emocional</i>
$ec_u$	Evento conversacional del usuario, es ocasionado por todas las interacciones del usuario almacenado. Este evento se almacena en las creencias del agente de <i>Creencias</i> y en el <i>Conversation Context</i> . Un evento $ec_u$ se puede representar como una tupla $ec_u = (text, q)$ donde $text$ es el texto del evento y $q$ el tipo de evento asociado. El evento puede ser una <i>query</i> del usuario, una respuesta del robot o únicamente un acto conversacional.	Agente de <i>Creencias</i>

Tabla 14. Eventos del Despachador de Eventos

Respecto a la línea 4, esta representa eventos relacionados con el agente de *Explicabilidad*. Este agente envía una colección de explicaciones  $EXP_q = \{exp_1, exp_2, \dots, exp_n\}$  cuando se determine que es necesario explicar verbalmente la toma de decisiones del robot. Por último, la línea 5 representa la entrada de eventos espontáneos generados por el dialogo del usuario y transmitidos por un agente *Proveedor de Servicios*.

### Agente Interprete Emocional Del Usuario

El agente *Interprete Emocional del Usuario* tiene como función hacer de agente multimodal de fusión para detectar de manera robusta las emociones del usuario de acuerdo con el trabajo de Alcalá[9]. De esta manera, el agente *Interprete Emocional del Usuario* permite reducir la imprevisibilidad en la toma de decisiones, al permitirle al sistema obtener información confiable sobre el estado mental del usuario. Este agente puede soportar un modelo emocional por cada usuario como se muestra en la Ilustración 6. Agente Interprete Emocional del Usuario.

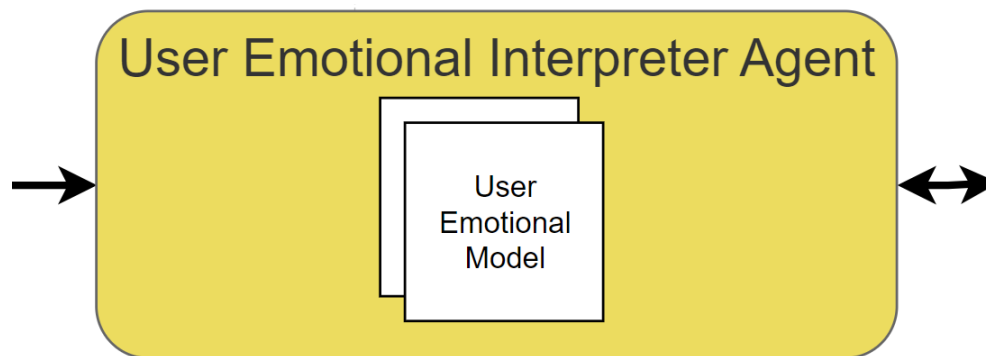


Ilustración 6. Agente Interprete Emocional del Usuario

Así mismo, este agente es el encargado de interpretar eventos informativos provenientes del agente Interactivo para afectar el estado emocional del robot. El agente genera eventos OCC [8] basados en el modelo emocional implementado por De la Peña[13], de acuerdo a eventos de interacción relacionados por el sistema. Para esto hace un mapeo entre eventos del sistema y eventos OCC, como muestra el ejemplo en la Tabla 15. Mapeo entre eventos.

Evento Interactivo	Origen	Evento OCC
El estado emocional del usuario es positivo	Agente Interprete Emocional del Usuario	<i>(made, robot, user feel good)</i>
El usuario dijo "Te amo"	Agente Interactivo	<i>(loves, user, robot)</i>

Tabla 15. Mapeo entre eventos

Este agente de fusión recibe eventos modales por parte de los agentes modales (*Agentes Proveedor de Servicio*), quienes implementan modelos de inteligencia artificial para detectar las emociones del usuario a partir de su expresión facial, su disposición corporal, sus signos vitales y su voz.

El agente *Interprete Emocional Del Usuario* toma la valencia de los eventos emocionales recibidos por los agentes modales y los almacena. Posteriormente, el agente toma el valor de la valencia y lo multiplica con el valor de la confiabilidad de la lectura. Por último, el agente *Interprete Emocional Del Usuario*, promedia las valencias por cada emoción según el resultado de cada agente y notifica un evento de reconocimiento al agente de *Motivación* y al agente *Interprete Emocional*, con las emociones que estén por encima de un umbral fijo. El evento de reconocimiento incluye la emoción y la valencia ponderada. Este proceso se muestra en la Ilustración 7. Proceso de fusión multimodal, tomado de [9].

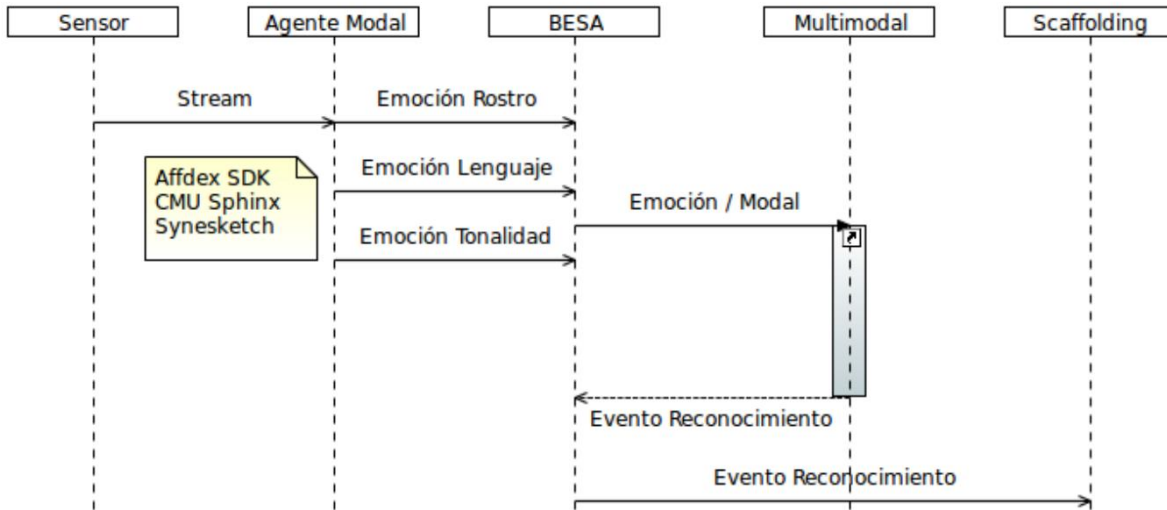


Ilustración 7. Proceso de fusión multimodal, tomado de [9]

Las creencias de este agente se pueden representar como:

$$(W_{EU}, M_{EU})$$

Donde:

- $W_{EU} = \{(W_{1,u_1} \dots W_{n,u_1}) \dots (W_{x,u_k} \dots W_{n,u_z})\}$  es el conjunto de pesos utilizados en el modelo emocional del usuario para definir la multimodalidad según n agentes modales.
- $M_{EU} = \{m_1 \dots m_n\}$  es el conjunto de relaciones entre los eventos de los demás agentes y el evento OCC generado.

Este agente está estrechamente relacionado con las creencias que involucran el estado de interacción del usuario, como lo son sus emociones y su nivel de atención. Por otro lado, se pueden representar como ( $M$ ) donde  $M$  es el mapeo entre eventos del sistema y eventos OCC. Cabe resaltar que este agente está estrechamente relacionado a la creencia del estado Psicológico del agente, ya que al enviar los eventos OCC al agente de *Creencias* se afecta únicamente ese módulo. Este agente representa una parte vital y necesaria en las habilidades de empatía expresadas por la arquitectura.

#### Agente Detector de Riesgos

El agente *Detector De Riesgos* tiene dos responsabilidades principales en la arquitectura. En primer lugar, se encarga de evolucionar y detectar los hábitos del usuario a lo largo del tiempo. En segundo lugar, tiene como objetivo el detectar riesgos relacionados con el cambio de hábitos o riesgos relacionados con la interacción humano-robot como lo pueden ser accidentes médicos o golpes. Este agente representa la capacidad del sistema de detectar y responder oportunamente a eventos imprevistos y a cambios alarmantes en el usuario a lo largo del tiempo.

Esto implica que el agente detector de riesgos debe contar con modelos de inteligencia artificial para lograr la detección. El agente de riesgos se relaciona con metas de mitigación o solución de riesgos en el agente de Motivación. Así mismo, tiene un componente dedicado para riesgos dentro del perfil del usuario y uno dedicado para la detección de accidentes.



La arquitectura del agente se muestra en la Ilustración 8. Agente Detector de Riesgos. Como es posible evidenciar, se cuenta con 3 módulos principales: el detector de hábitos, el detector de anomalías en los hábitos y el detector de accidentes. Cabe resaltar que el detector de accidentes puede ser especializado varias veces durante la implementación con el fin de detectar varias actividades que puedan ocasionar un accidente.

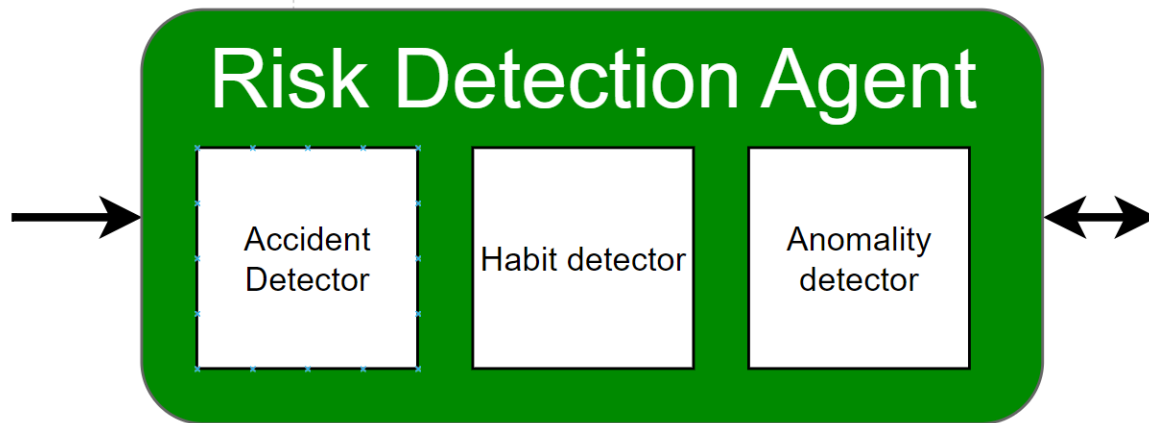


Ilustración 8. Agente Detector de Riesgos

El agente *Detector De Riesgos* también puede recibir información de 2 fuentes diferentes:

- Agentes *Proveedores de Servicio*, quienes le proveen información sensorial como video en vivo para la detección de actividades.
- Agente *Detector Emocional del Usuario*, el cual le facilita la identificación de riesgos emocionales como la ansiedad o la depresión.

Las creencias del agente se pueden generalizar como la tupla

$$(Ac, Acd, Hab)$$

Donde:

- $Ac = \{ac_1 \dots ac_n\}$  es el conjunto de actividades y accidentes que deben ser detectados por el agente. La información de las actividades puede variar según los algoritmos de inteligencia artificial utilizados y según el caso de aplicación.
- $Acd = \{acd_1 \dots acd_n\}$  es el conjunto con las funciones que retornan si un accidente o anomalía ha sido detectado o no. Su implementación depende del tipo de riesgo a detectar.
- $Hab = \{(h_{1,u_1} \dots h_{n,u_1}), (h_{1,u_k} \dots h_{n,u_k})\}$  es el conjunto de hábitos diferenciados por usuario, copiados de los perfiles del usuario en las creencias. Cabe resaltar que cada hábito  $h_{n,u_k} = \{(t_1, ac_1) \dots (t_n, ac_n)\}$  está compuesto de actividades presentes en  $Ac$  e identificadas temporalmente

#### Agente Detector de Riesgos Éticos

El agente *Detector de Riesgos Éticos* representa una capa de ética embebida para la toma de decisiones del robot. Su función se inspira en dos trabajos encontrados durante el estado del arte [14] [15], sin embargo busca alejarse de verificar que cada acción realizada sea ética, ya que esto

queda en manos del desarrollador al llenar las funciones de legalidad de las metas del modelo BDI Besa.

El agente *Detector de Riesgos Éticos* busca detectar y mitigar riesgos éticos que ocurran a largo plazo durante la interacción. De esta manera, el sistema logra reducir la desconfianza en sus capacidades e incluso puede reducir la imprevisibilidad ocasionada por la interacción humano robot a largo plazo.

A nivel general, el agente funciona como un detector al implementar heurísticas y algoritmos de IA para identificar los riesgos. Un ejemplo, puede ser la frecuencia de uso del sistema por parte del usuario para detectar apego.

A manera de mitigación, el agente está en la capacidad de notificar a las creencias en su componente de riesgos éticos. Una vez las creencias han sido adaptadas, se pueden activar planes implementados para la mitigación de los riesgos específicos. Por otro lado, el agente puede afectar la modulación de acciones mediante los roles de interacción para mitigar riesgos relacionados a las emociones. Este puede ser el caso del apego.

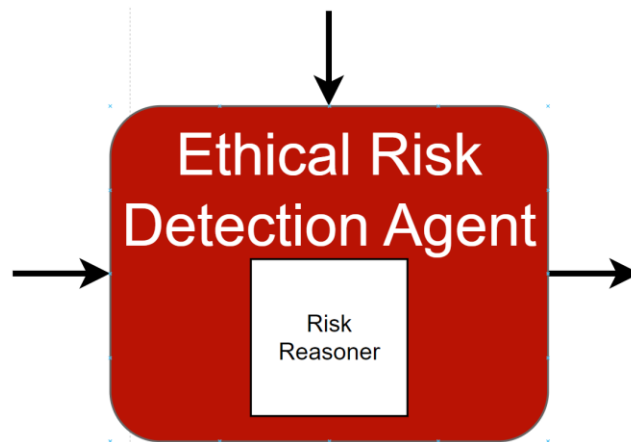


Ilustración 9. Agente Detector de Riesgos Éticos

La arquitectura del agente se puede ver en la Ilustración 9. Agente Detector de Riesgos Éticos. Es posible evidenciar que consta de un razonador de riesgos, cuya especificación puede depender del algoritmo de IA utilizado o del contexto de aplicación específico. Sin embargo existen ciertos riesgos, como el apego que son generales para los robots de asistencia social.

Las creencias del agente se pueden representar como:

$$(RE, RD, Rk, Rd, Rm)$$

Donde:

- $RE$  son los eventos recibidos por el agente de los agentes *Proveedor de Servicio*.
- $RD$  son los eventos enviados por el agente hacia el agente de *Creencias* cuando un riesgo es detectado.
- $Rk = \{rk_1 \dots rk_n\}$  es un conjunto de riesgos a evaluar por el agente al momento de recibir información de alguna fuente.

- $Rd = \{rd_1 \dots rd_n\}$  es el conjunto con las funciones que retornan si un riesgo ha sido detectado o no. Su implementación depende del tipo de riesgo ético a detectar. En el caso de la autonomía del usuario, la función podría evaluar si a medida que pasa el tiempo el usuario ha dejado de realizar actividades con el apoyo del robot y más bien está dejando que el robot las realice por él.
- $Rd = \{rd_1 \dots rd_n\}$  representa el grupo de estrategias para la mitigación de los riesgos detectados.

### Módulo de Decisión

Los agentes del módulo de decisión SocialRobotBesa son aquellos encargados de realizar los procesos funcionales más importantes del sistema, desde la personalización y adaptación, hasta la ejecución de servicios para el bienestar del usuario. Este grupo de agentes se divide entre los agentes de *Cooperación*, el agente de *Aprendizaje*, el agente de *Explicabilidad* y el agente *BDI*.

A su vez, *el agente BDI* se encuentra dividido en tres subagentes. El agente de *Creencias*, el agente de *Motivación* y el agente de *Acción*. Cada uno de estos subagentes fue adaptado de las tesis doctorales de John Páez [16], Angela Bravo [3] y Alejandra Gonzalez [5].

### Agente De Aprendizaje

El agente de aprendizaje es uno de los agentes de mayor relevancia en lo que refiere a procesos de adaptación a lo largo del tiempo para la gestión de la imprevisibilidad. La arquitectura puntual de este agente está basada en q-learning y está enfocada hacia el aprendizaje y adaptación de los servicios provistos a partir de la interacción. Ahora bien, cabe mencionar que esta arquitectura es solo un inicio para este agente, ya que es posible integrar más modelos de inteligencia artificial para otro tipo de objetivos durante la interacción. La arquitectura del agente se muestra en la Ilustración 10. Agente de Aprendizaje.

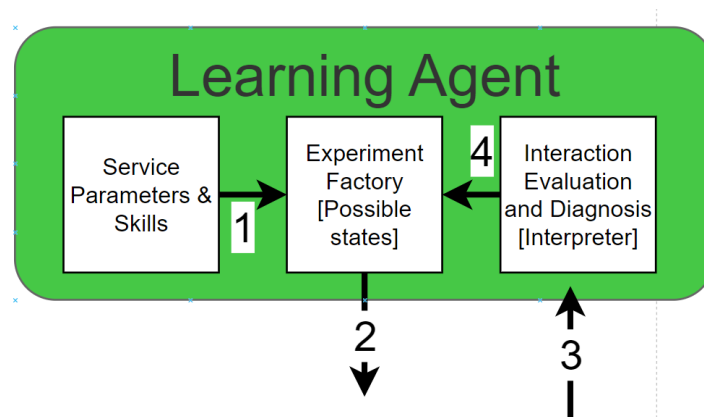


Ilustración 10. Agente de Aprendizaje

El agente consta de 3 módulos:

- **Parámetros de servicios y habilidades:** Descriptor de los servicios que incluye los parámetros utilizados durante el servicio. A partir de este se generan los posibles estados como se ve en la interacción 1.
- **Fábrica de Experimentos:** Posibles estados explorados o a explorar. Pueden representar cambios en la emocionalidad del agente, en el tipo de contenido dado por la actividad,

incluso puede estar relacionado con la dificultad de un servicio de juegos. Cuando se ha determinado el experimento a realizar, se actualiza el contexto de servicio en el agente de *Creencias* como se evidencia en la interacción 2.

- Diagnóstico y Evaluación de la Interacción: Encargado de interpretar la reacción del usuario en términos de recompensa. Hace de función de evolución de los experimentos generados por el agente. Está recibiendo las actualizaciones en las creencias relacionadas con el estado de interacción del usuario como se evidencia en la interacción 3. Así mismo, retroalimenta al componente de experimentación para que pueda tomar una decisión a partir de su desempeño. Esto se evidencia en la interacción 4

Por otro lado las creencias del agente representan como:

$$(S, S', Sd, Ps, EVf, AE)$$

Donde:

- $S = \{s_1 \dots s_n\}$  es el conjunto de servicios que pueden ser utilizados durante el aprendizaje.
- $S'$  es el conjunto de servicios que son intenciones en un momento específico.
- $Sd = \{sd_1 \dots sd_n\}$  es el conjunto de descriptores de servicio que especifica los parámetros que pueden ser modificados por experimento.
- $Ps = \{ps_1 \dots ps_n\}$  representa los posibles estados para el aprendizaje de políticas bajo el contexto del servicio
- $EVf = \{evf_1 \dots evf_n\}$  es el conjunto de funciones de evaluación y desempeño para determinar la recompensa del experimento.
- $AE$  son los eventos recibidos de las creencias para la evaluación de la interacción con el usuario a partir del experimento generado.

#### Agente de Explicabilidad

El agente de *Explicabilidad* se encarga de la recolección de registros originados por el proceso de toma de decisiones y el cambio en las creencias del agente. Este agente permite facilitar la implementación del aspecto explicativo ético visto en el estado del arte al permitir mostrar porque el robot selecciona una intención en específico a partir de sus creencias. La arquitectura del agente se muestra en la Ilustración 11. Agente de Explicabilidad.

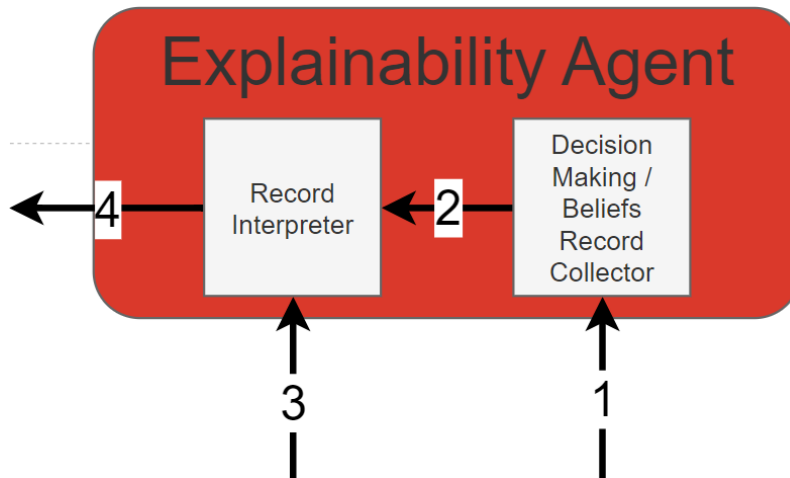


Ilustración 11. Agente de Explicabilidad

Como es posible evidenciar la arquitectura formal del agente *Explicativo* consta de dos componentes:

- El Colector de Registros: se encarga de procesar la información enviada por el agente de creencias cuando recibe una notificación y por el agente de Motivación cuando se realiza el proceso de deliberación (Interacción 1). Recibe un evento con la información e implementa la lógica de traducción a registro. Una vez finalizado, pasa los registros al interpretador de registros (Interacción 2).
- El Interpretador de Registros: recibe la notificación del agente de Motivación para la traducción a lenguaje interpretable (Interacción 3) y para el envío de los registros al agente Interactivo (Interacción 4).

Cabe resaltar que este agente es extensible y es posible añadir más eventos al enviar información a sus interfaces de entrada. Su creencia se puede expresar como  $(RG, RP)$ , donde  $RG = \{rg_1 \dots rg_n\}$  corresponde al conjunto de registros almacenados por el agente de explicabilidad y donde  $RP = \{rp_1 \dots rp_n\}$  es el conjunto de registros procesados por el agente. Así mismo un registro  $rg_x = \{rT, t_x\}$  y un registro  $rp_x = \{rTp, t_x\}$ , donde  $rT$  es un objeto de creencias o de la toma de decisiones a ser procesado,  $rTp$  es un registro interpretable y ya procesado por el agente y  $t_x$  es la marca de tiempo relacionada.

La salida de este módulo se puede interpretar como la tupla  $(RR)$ , la cual corresponde al conjunto de registros relevantes para ultima toma de decisiones del agente de Motivación.

### Agente de Cooperación

El agente de *Cooperación* es el responsable de manejar la comunicación y la sincronización entre agentes digitales. El intercambio de mensajes entre agentes puede ser realizado mediante cualquier protocolo que sea soportado por el manager de cooperación (Interacción 1). Este agente es adaptado del trabajo de Bravo [3] y se muestra en la Ilustración 12. Agente de Cooperación.

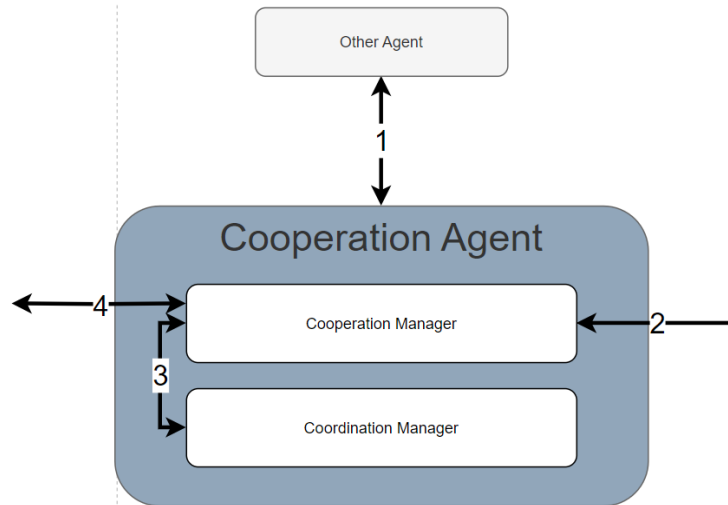


Ilustración 12. Agente de Cooperación

El agente de cooperación requiere de un manager de cooperación cuya información enviada o recibida puede ser descrita, como mínimo, por la tupla:

$$(CE_r, CE_q, CE_{AC}, CE_{AH})$$

Donde:

- $CE_r$  son los eventos relacionados a las respuestas de otros agentes que después son retransmitidos al agente de Creencias para actualizar información nueva sobre la interacción (Interacción 4).
- $CE_q$  son eventos relacionados a peticiones hechas por el agente de Motivación con el fin de realizar una acción a través de otro agente.
- $CE_{AC}$  son eventos que indican la finalización de una acción por parte de un agente involucrado en un proceso de coordinación. Posteriormente este evento es enviado al manager de Coordinación (Interacción 3).
- $CE_{HC}$  son eventos que indican la finalización de una acción por parte de un humano involucrado en un proceso de coordinación entre el agente, otros agentes y humanos (Interacción 2). Posteriormente este evento es enviado al manager de Coordinación.

Cabe mencionar que pueden añadirse más eventos según las necesidades del contexto de aplicación.

Por otro lado, el agente de Cooperación posee un manager de Coordinación el cual se encarga de la implementación de protocolos de cooperación y de coordinación con los demás agentes. Sus eventos pueden describirse como:

$$(CE_{SC}, CE_{EN}, CE_{SN}, CE_{FA})$$

Donde:

- $CE_{SN}$  son los eventos relacionados al inicio de protocolos de cooperación o coordinación. Estos son enviados a otros agentes interesados con el fin de iniciar la negociación.

- $CE_{EN}$  son eventos relacionados con la finalización de la negociación para los protocolos. En esto se definen los agentes involucrados en la cooperación.
- $CE_{SC}$  son eventos que indican el inicio del proceso de coordinación entre agentes.
- $CE_{FA}$  son eventos que indican la finalización de una acción por parte de un agente involucrado en la coordinación.

### Agente de Cooperación Humana

El agente de *Cooperación Humana* es el responsable de manejar la comunicación y la sincronización del agente SocialRobotBesa con agentes humanos. Como se vio en el estado del arte, la autonomía adaptativa requiere de incluir a los actores involucrados en el cuidado de usuario durante la ejecución de los planes. El intercambio de mensajes entre humanos y el agente son mediados por infraestructura tecnológica soportado por el Agente Interactivo (Interacción 6) o los agentes *Proveedor de Servicio* (Interacción 4). Este agente se muestra en la Ilustración 13. Agente de Cooperación Humana. Cabe resaltar que este agente hace parte de la capa de reactividad del sistema la enviar notificaciones de inmediato cuando se detecta un accidente.

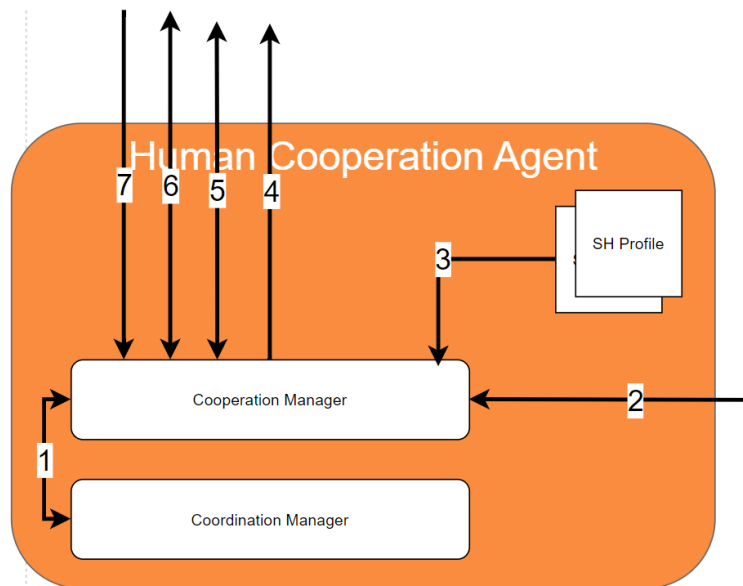


Ilustración 13. Agente de Cooperación Humana

El agente de cooperación humana requiere de un manager de cooperación cuya información enviada o recibida puede ser descrita, como mínimo, por la tupla:

$$(CE_{AUH}, CE_{rh}, CE_{qh}, CE_{AC}, CE_{AH})$$

Donde:

- $CE_{AUH}$  son los eventos relacionados con el proceso de autenticación de un tercero (Interacción 3). Si el tercero es autenticado, el humano puede ser tenido en cuenta para los procesos de cooperación.
- $CE_{rh}$  son los eventos relacionados a las respuestas del humano que después son retransmitidos al agente de Creencias para actualizar información nueva sobre la interacción

(Interacciones 6 y 7). Como ejemplo se tiene la respuesta a la autorización de una meta con autonomía media por parte de un cuidador.

- $CE_q$  son eventos relacionados a peticiones hechas por el agente de Motivación con el fin de solicitar la ayuda de un agente humano (Interacción 2). El mejor ejemplo, es cuando el agente de *Motivación* solicita al cuidador que autorice la realización de una meta con baja autonomía.
- $CE_{AC}$  son eventos que indican la finalización de una acción por parte de un agente digital involucrado en un proceso de coordinación. Posteriormente este evento es enviado al manager de Coordinación (Interacción 1).
- $CE_{AC}$  son eventos que indican la finalización de una acción por parte de un humano involucrado en un proceso de coordinación entre el agente, otros agentes y humanos. Posteriormente este evento es enviado al manager de Coordinación (Interacción 5).

Cabe mencionar que pueden añadirse más eventos según las necesidades del contexto de aplicación.

Por otro lado, el agente de Cooperación Humana posee un manager de Coordinación el cual se encarga de la implementación de protocolos de cooperación y de coordinación con los seres humanos. Este módulo se encarga de manejar la imprevisibilidad resultante de trabajar junto a los humanos, quienes no siempre cumplen sus promesas. Sus eventos pueden describirse como:

$$(CE_{SC}, CE_{EN}, CE_{SN}, CE_{FA})$$

Donde:

- $CE_{SN}$  son los eventos relacionados al inicio de protocolos de cooperación o coordinación. Estos son enviados a otros humanos interesados con el fin de iniciar la negociación.
- $CE_{EN}$  son eventos relacionados con la finalización de la negociación para los protocolos. En esto se definen los humanos involucrados en la cooperación.
- $CE_{SC}$  son eventos que indican el inicio del proceso de coordinación entre el agente y los humanos.
- $CE_{FA}$  son eventos que indican la finalización de una acción por parte de un humano involucrado en la cooperación.

### Agente de Creencias

El objetivo del agente de Creencias es el de manejar las creencias del agente de robótica de asistencia social. Este agente es adaptado del trabajo de Bravo [3] y se muestra en la Ilustración 14. Agente de Creencias. Las creencias en esta arquitectura se agruparon varios componentes que pueden ser suscritos por los diferentes agentes del módulo de soporte y del módulo de decisión. Esto se debe a que el agente de Creencias funciona como un proveedor de servicio de información para el agente de Motivación, el agente de Acción y para otros agentes del módulo de soporte. Para el poder manejar la imprevisibilidad en tiempo real, se hace necesario que las creencias sean proactivas y que no sea necesario que los demás agentes las consulten de vez en cuando.

Las creencias pueden ser actualizadas por eventos asincrónicos provenientes de 4 orígenes:

- Los agentes de Cooperación: los eventos recibidos por otros agentes que provean nueva información y los eventos de la colaboración con seres humanos también permiten la actualización de las creencias. Por ejemplo, si un wearable envía información médica sobre



el usuario esto afectaría las creencias del estado de la interacción. Por otro lado, si el cuidador autoriza realizar una meta de servicio, esto también se vería reflejado en las creencias.

- Información calculada por agentes de decisión secundarios: Los eventos generados por agentes como el de Aprendizaje y el de explicabilidad, pueden generar cambios en las creencias. Por ejemplo, si el agente de aprendizaje considera que el usuario gusta más de la actividad de cuentería y lo comunica al agente de Creencias, las creencias se verán afectadas.
- Información provista por el Módulo de Soporte: los eventos informativos de la capa de soporte pueden generar información valiosa al negocio que queda reflejada y compartida en las creencias. Por ejemplo, el agente de riesgos éticos podría detectar el riesgo de apego durante la interacción y enviar el reporte su componente específico dentro de las creencias.
- Información sensorial procesada por los agentes Proveedor de Servicio: eventos asociados con lecturas de sensores también pueden actualizar las creencias. Por ejemplo, un evento verbal va a cambiar el contenido del estado de la interacción en las creencias.

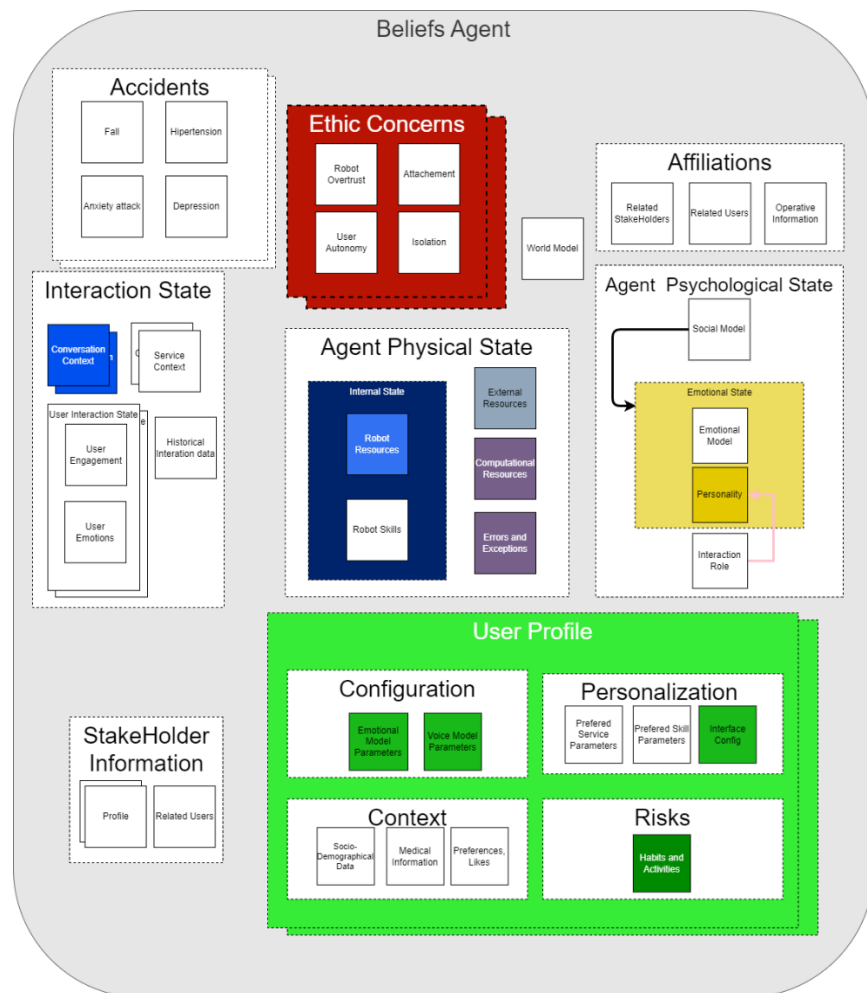


Ilustración 14. Agente de Creencias

Cada componente de las creencias se explica a detalle a continuación.

## Estado Físico del Agente

El componente del estado físico del agente contiene creencias sobre los recursos del agente según su naturaleza. Estos se muestran a detalle en la Ilustración 15. Estado Físico del Agente.

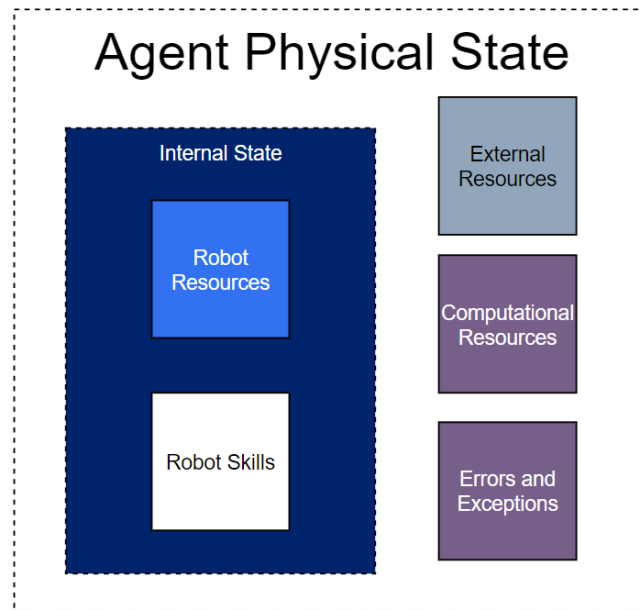


Ilustración 15. Estado Físico del Agente

El estado Físico del agente se puede modelar como la tupla:

$$(Rr, Sk, Er, Cr, \varepsilon, \epsilon)$$

Donde,

- $Rr = \{(r_1, s_1) \dots (r_n, s_n)\}$  es el conjunto de recursos que tiene el robot y su estado. Es importante que esto este indicado en las creencias ya que los recursos del robot podrían cambiar dinámicamente.
- $Sk = \{sk_1 \dots sk_n\}$  es el conjunto de habilidades del robot, las cuales pueden restringir o no su autonomía.
- $Er = \{(er_1, s_1) \dots (er_n, s_n)\}$  es el conjunto de recursos externos adquiridos a partir de la colaboración con otros agentes y el estado de estos recursos.
- $Cr = \{(cr_1, s_1) \dots (cr_n, s_n)\}$  es el conjunto de recursos computacionales como la RAM y la CPU, y su estado.
- $\varepsilon = \{\varepsilon_1 \dots \varepsilon_n\}$  es el conjunto de errores que pueden darse en el robot o en el sistema.
- $\epsilon = \{\epsilon_1 \dots \epsilon_n\}$  es el conjunto de excepciones que pueden darse en el robot o en el sistema.

Es importante mencionar que  $Rr$ ,  $Sk$  y  $Er$  son utilizados por el agente de Motivación para el proceso de deliberación donde se verifican los conflictos por recursos. De igual manera,  $Cr$ ,  $\varepsilon$  y  $\epsilon$  pueden ser utilizados por metas centradas en la recuperación de errores o en el manejo de excepciones.

## Estado de la Interacción

El componente de estado de interacción tiene como objetivo relacionar toda la información resultante de la interacción con los usuarios. Se compone de 4 subcomponentes principales, los

cuales serán descritos a continuación. El estado de interacción se muestra en la Ilustración 16. Estado de Interacción

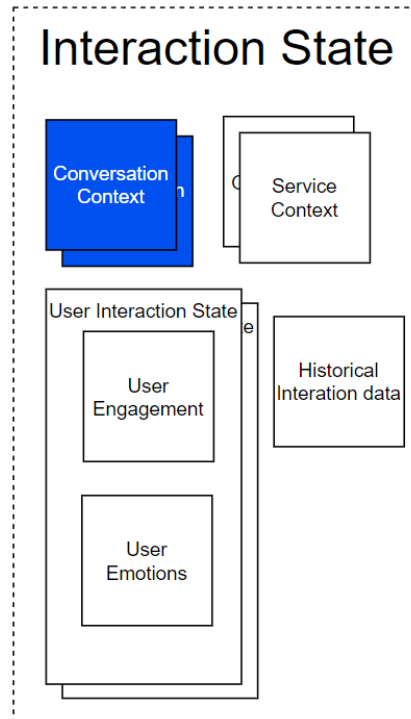


Ilustración 16. Estado de Interacción

El estado de la Interacción se puede modelar como la tupla:

$$(Cc, Sc, UIs, H)$$

Donde,

- $Cc = \{cc_{u_1} \dots cc_{u_n}\}$  es el contexto de la conversación de los diferentes usuarios con el agente. Cada estado de conversación se puede entender como el conjunto de registros verbales generados entre el usuario  $u$  y el agente. Esta creencia es actualizada exclusivamente por el agente Interactivo.
- $Sc = \{sc_{s_1} \dots sc_{s_n}\}$  es el conjunto de contextos de servicio del sistema. Dependiendo del servicio el contenido de esta creencia es diferente. Así mismo, el contexto de servicio debe implementar la lógica para manejar uno o múltiples usuarios.
- $UIs = \{uis_{u_1} \dots uis_{u_n}\}$  es el conjunto de estados de interacción por usuario. Esto incluye la información de las emociones y del engagement del usuario. La información de las emociones es un conjunto de tuplas  $E_{u_x} = \{(e_1) \dots (e_n)\}$  con cada emoción  $e_x$  representada por una valencia. Esta creencia es actualizada exclusivamente por el agente Interprete Emocional del Usuario.
- $H = \{(Cc, Sc, UIs, t_1) \dots (Cc, Sc, UIs, t_n)\}$  es la información histórica de las intervenciones con cada usuario. Sirve de repositorio de información histórica y facilita la toma de decisiones a largo plazo.

## Perfil del Usuario

El componente de perfil del usuario tiene como objetivo brindar un modelo del usuario fiable y útil para la toma de decisiones de los demás agentes. Este es un componente central de la arquitectura ya que facilita la personalización, adaptación y detección de riesgos. El componente se muestra en la Ilustración 17. Perfil del Usuario.

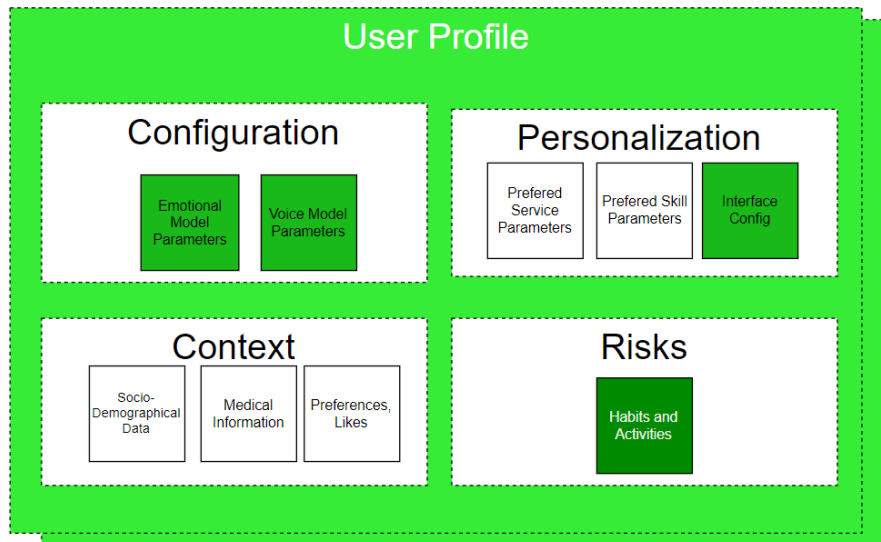


Ilustración 17. Perfil del Usuario

### Configuración

Este subcomponente gestiona la información relacionada con los parámetros del modelo emocional y del modelo de voz del usuario para brindar una mejor interacción. Esto se puede representar como:

$$(W_E, W_V)$$

Donde:

- $W_E$  son los pesos relacionados con el modelo emocional del usuario para la fusión multimodal.
- $W_V$  son los pesos relacionados al modelo de voz a texto del usuario de manera que estos también puedan ser personalizados.

### Personalización

Este subcomponente incluye las configuraciones realizadas mediante experimentos y su desempeño. De igual manera incluye la información relacionada con las preferencias del usuario a en cada servicio. Esto se puede representar como:

$$PSP, PSKP, IC$$

Donde:

- $PSP$  son la configuración preferencia de los servicios. Por ejemplo, para musicoterapia un usuario específico puede preferir que el robot no baile y más bien muestra la letra de la canción.

- *PSKP*, es la configuración preferida de las habilidades del robot. Por ejemplo, para la habilidad bailar un usuario puede preferir que la danza sea lenta y formal en vez de rápida y caótica-
- *IC*, es la configuración de interfaz que si bien se relaciona con las necesidades cognitivas del usuario, también puede ser sujeta a personalización por parte del usuario principal. Por ejemplo, si bien un usuario no tiene problemas de vista podría preferir un texto más grande en la tablet por comodidad.

#### Contexto

Este subcomponente incluye Información socio demográfica, médica y de preferencias del usuario. Con esta información se busca apoyar procesos de aprendizaje y permitir la modulación correcta del contenido de las actividades al igual que de la interacción. Esto se puede representar como:

$$(Pr_{SD}, Pr_{MI}, Pr_{PRF})$$

Donde:

- $Pr_{SD}$  representa la información socio demográfica como cultura, edad, nombre y sobrenombre. Esta información es utilizada para la personalización más básica, como decirle al usuario por su nombre.
- $Pr_{MI}$  se compone de la información medica del usuario. Esta puede ser especifica según el contexto, pero a nivel general podría relacionarse a un usuario con ceguera parcial el cual requiere comunicación verbal en vez de comunicación por contenido visual.
- $Pr_{PRF}$  representa las preferencias del usuario. Puede estar relacionado a conceptos como preferencias de genero musical o de género literario, así como las preferencias por cada servicio provisto por el robot.

#### Riesgos

Este subcomponente contiene información relacionada con riesgos habituales del usuario, que pueden ser medidos durante la interacción. Por ejemplo, hábitos y actividades. Este componente se encarga de apoyar las actividades del agente detector de riesgos. Esto se puede representar como:

$$(Ac, Hab)$$

Donde:

- $Ac$  son las actividades identificadas para un usuario específico.
- $Hab$  es el conjunto de hábitos compuesto por múltiples actividades presentes en  $Ac$ , que sirven para identificar si un habito ha cambiado o si se ha presentado una anormalidad.

#### Preocupaciones Éticas

Es el componente encargado de almacenar las variables éticas de riesgo, puede corresponder a un número o a un booleano que indica si el riesgo se presenta. De esta manera, permite que la meta correspondiente se active para la selección durante la deliberación del agente de Motivación. Este componente es gestionado únicamente por el agente detector de riesgos éticos.

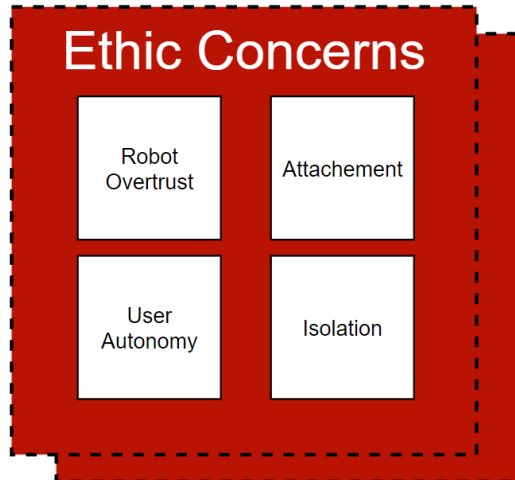
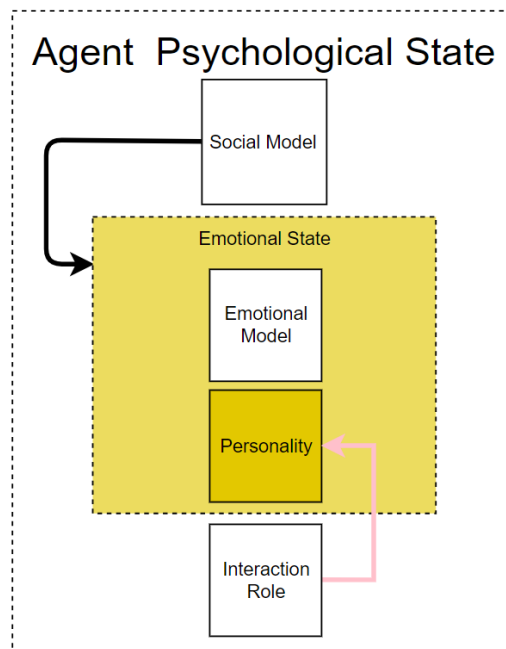


Ilustración 18. Preocupaciones éticas

Las creencias de este agente varían bastante según su contexto de aplicación sin embargo, a nivel general, sus creencias se pueden representar como  $(Rx)$  donde  $Rx = \{r_1 \dots r_x\}$  que representa el conjunto de variables detectoras de riesgo gestionadas por el agente de detector de riesgos.

#### Estado Psicológico del Agente

Este componente es sin duda uno de los componentes más importante del agente de Creencias. El propósito de este módulo es el de representar el estado emocional del agente y características moduladoras como el modelo social, la personalidad y los roles de interacción. La estructura de este componente se presenta en Ilustración 19. Estado Psicológico del Agente. Este módulo es gestionado preferencialmente por el agente Interprete Emocional, ya que es quien le envía eventos para actualizarlo. En esta arquitectura este componente de las creencias es utilizado como base para generar el enriquecimiento emocional de las acciones por parte del agente de *Acción*.



Este componente está basado en el modelo emocional definido por Bravo [3] e implementado por De la Peña [13]. Este se encuentra basado en el trabajo de Ortony, Clore y Collins [8] y los aportes realizados por Steunebrink [17]. En el modelo OCC, cada evento se define por una acción, el sujeto de la acción y el objeto/contexto de la acción. Un ejemplo específico se relaciona con la frase “el robot (sujeto de la acción) alegra (acción) al adulto mayor (objeto de la acción)”.

Es decir que un evento emocional se representa como:

$$\epsilon = (ev, ag, ob)$$

En esta arquitectura los eventos emocionales son generados por la interacción del usuario y por servicios que así lo requieran. En cuanto a los eventos emocionales generados por la interacción, se puede decir que estos se originan de los agentes *Interprete Emocional del Usuario* para fomentar la empatía y del agente *Interactivo* con el fin de hacer que el robot también cambie su emoción partiendo de lo que se habla con el usuario. Respecto a los eventos generados por servicios que lo requieran, el mejor ejemplo es cuentaría, donde para actuar el cuento el agente de Motivación envía eventos emocionales internos que modifican el estado emocional del agente.

El estado emocional se representa como un conjunto  $EA = (EA_1 \dots EA_n)$  de ejes emocionales. Cada eje tiene dos emociones y un rango de valencias específico. La emoción positiva varía de 0 a 1 y la emoción negativa de 0 a -1. El ejemplo clásico es felicidad – tristeza, donde 1 indica felicidad máxima, 0 ausencia total de emoción y -1 tristeza máxima.

Además, cada eje emocional tiene un valor predeterminado ( $v_k$ ) que corresponde al estado emocional base del agente. Este valor  $v_k$  permite modelar el estado predeterminado de un agente, lo que significa que es posible hacer agentes propensos a la felicidad o a la tristeza.

Así mismo, el valor numérico de la representación de los eventos, los agentes y los objetos para un evento emocional depende de un diccionario de etiquetas semánticas [13]. En este se especifica si por ejemplo un evento es deseable o indeseable, y se asocia un valor a este. De igual manera, se especifica el grado de afecto que se tiene por un agente en particular y la influencia que tiene un evento sobre un objeto. En el caso de este trabajo de grado, se considera esto como la personalidad.

La personalidad se puede representar como:

$$Pers = (R_{ev}, R_{ag}, R_{obj})$$

Donde:

- $R_{ev}$  es el conjunto de relaciones del agente con los eventos. Es decir que tan deseables o indeseables estos son.
- $R_{ag}$  es el conjunto de las relaciones de afecto que tiene este agente con otros agentes.
- $R_{obj}$  es el conjunto de relaciones de influencia que tiene un objeto sobre las emociones de un agente.

Por último, para determinar si un evento aumenta o disminuye el nivel de un eje, se tienen en cuenta una serie de reglas a partir de los signos de los valores del evento, del agente y del objeto. Esto se muestra en la Ilustración 20. Cálculo de la valencia del evento emocional.

$ev_{\epsilon_{k_i}}$	$ag_{\epsilon_{k_i}}$	$ob_{\epsilon_{k_i}}$	$v_{\epsilon_{k_i}}$
-v	-v	-v	1
-v	-v	+v	-1
+v	-v	-v	-1
+v	-v	+v	-1
-v	+v	-v	1
-v	+v	+v	-1
+v	+v	-v	-1
+v	+v	+v	1

Ilustración 20. Cálculo de la valencia del evento emocional, tomado de [3]

En general el modelo emocional puede ser representado por [3]:

$$EEMB = (I_e, EAD, I_{eD})$$

Donde:

- $I_e$  es el conjunto de intensidades de cada eje emocional
- $EAD$  es el eje emocional dominante
- $I_{eD}$  es la intensidad del eje emocional dominante.

Respecto a los roles de interacción, estos están en la capacidad de cambiar los valores numéricos que representan la personalidad. Es decir, los roles pueden influir en las relaciones que tiene el agente con el demás agente, en la importancia que les asignan a ciertos objetos e incluso en la deseabilidad de un evento en específico.

Así mismo, los roles de interacción tienen la capacidad de cambiar el valor  $v_k$  de un agente, de manera que pueden cambiar su estado emocional predeterminado. Como trabajo futuro la implementación del modelo se puede extender a lo mencionado en [3].

#### Información de los SH

Componente que contiene la información relevante a la autenticación de los SH relacionados con el cuidado, así como los usuarios relacionados a cada SH. Esta es enviada al agente de Cooperación con humanos para procesos de autenticación. El contenido de este componente se muestra en la Ilustración 21. Información de Terceros.

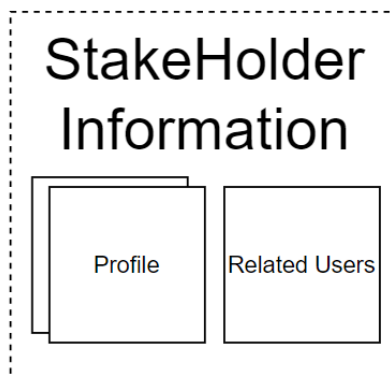


Ilustración 21. Información de Terceros



Estas creencias se pueden representar como:

$$(SH_v, SH_p, SH_{sd}, SH_{ru})$$

Donde:

- $SH_v$  es la información biométrica o de autenticación del Stakeholder.
- $SH_p$  es la información de personalización del Stakeholder, la cual puede influir en los medios utilizados para lograr la colaboración con el robot. Por ejemplo si el Stakeholder prefiere ser contactado vía WhatsApp el razonador de medios y fines preferirá utilizar un agente Proveedor de Servicio para enviar el mensaje en vez de pedirle al agente interactivo que lo haga.
- $SH_{sd}$  es la información referente a los datos socio demográficos del Stakeholder, puede incluir nombre, edad y profesión. Esto es útil para mejorar las notificaciones o la comunicación.
- $SH_{ru}$  es la información de los usuarios relacionados al Stakeholder.

### Accidentes

Corresponde a las banderas utilizadas por el agente Detector de Riesgos para activar los planes relacionados con la mitigación o activación de un riesgo. Consta de una bandera por cada riesgo o accidente a ser detectado.

### Afiliaciones

Componente utilizado para representar información relevante sobre posibles actividades o grupos a los que pertenece el usuario. Esta información puede implicar actividades o servicios específicos, los cuales han sido configurados en una misma hora para todos los usuarios. El contenido de este componente se muestra en la Ilustración 22. Información de Afiliaciones.

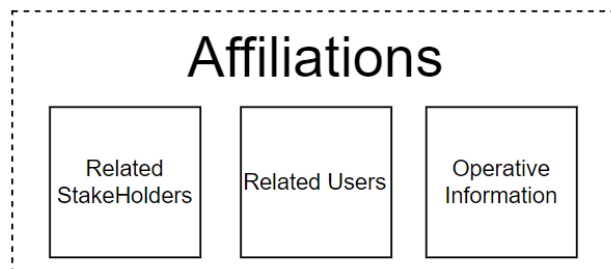


Ilustración 22. Información de Afiliaciones

Estas creencias pueden ser modeladas como:

$$(SH, Us, OI)$$

Donde:

- $SH$  son los Stakeholders involucrados en el cuidado del usuario para esta afiliación en específico.
- $Us$  son los usuarios relacionados a una afiliación.
- $OI$  es la información operativa, como la hora, el lugar y el motivo por el cual esta programado un servicio.

## Modelo del Mundo

Componente encargado de modelar el mundo físico de la interacción, almacena la información del movimiento, la ubicación y la propiocepción del robot. Esto se puede representar como:

$$(MP, A_L, OB_L, OA_L)$$

Donde:

- $MP$  es la representación de los límites del ambiente donde está ubicado el robot. Esta creencia puede cambiar a medida que el robot se mueva por el espacio.
- $A_L$  es la ubicación del agente en el espacio. Esta creencia cambia a partir de la información de desplazamiento provista por un agente *Proveedor de Servicio*.
- $OB_L$  es la ubicación de obstáculos en el espacio, la cual se puede ver modificada mediante el *active sensing del espacio*.
- $OA_L$  es la ubicación de otros agentes humanos o físicos en el espacio, la cual es modificada a partir de eventos sensoriales enviados por un agente *Proveedor de Servicio*.

## Agente de Motivación

El agente de motivación recibe eventos informativos *inf* mediante la interacción 1 del agente de *Creencias* para determinar que metas deben ser activadas. Este agente es adaptado del trabajo de Bravo [3] y se muestra en la Ilustración 23. Agente de Motivación.

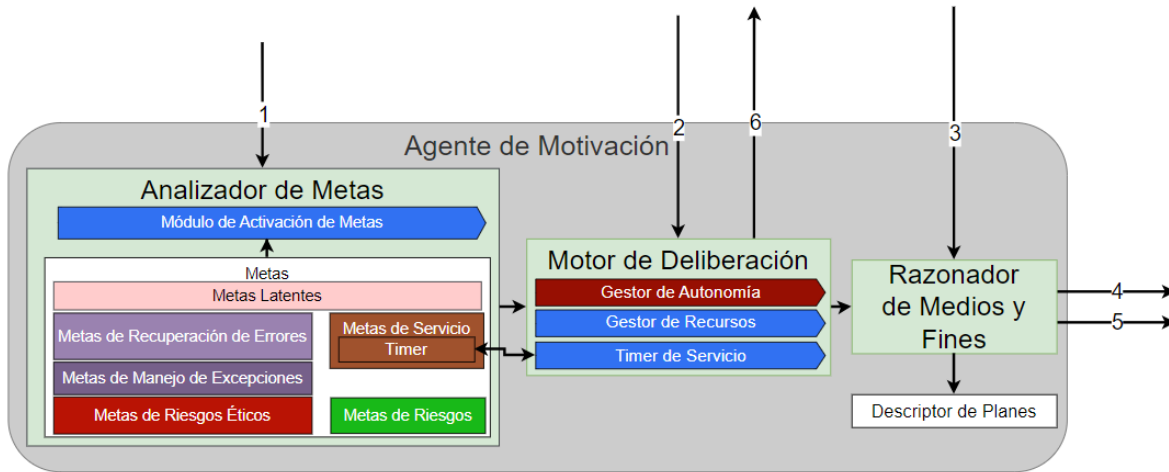


Ilustración 23. Agente de Motivación

Cuando el evento *inf*, definido como  $inf = \{BA, val\}$ , donde BA es la creencia afectada y val es el nuevo valor de la creencia, llega al agente de motivación el Analizador de Metas es el primer componente en activarse. Esto se ve representado en la interacción 1. El Analizador de Metas tiene como función iniciar el proceso de deliberación para determinar que metas pueden convertirse en intención. La salida de este módulo se representa como una tupla ( $D_{BDI}$ ) que representa las metas activas del agente.

### Analizador de Metas

Para generar esta salida, el módulo tiene la siguiente información representada en una tupla:

$$(R, G_{BDI}, G_L, AF_{G_{BDI}}, CF_{G_{BDI}}, AF_{G_L}, CF_{G_L}, W_G, RO)$$

donde  $R = \{r_1, \dots, r_n\}$  y  $r_x = \{\{BA_1..BA_k\}, \{g_1 \dots g_x\}\}$ . En este caso una regla  $r_x$  consta de un conjunto de antecedentes  $\{\{BA_1..BA_k\}\}$  que indican las creencias asociadas a una serie de consecuentes  $\{g_1 \dots g_x\}$  que indican que metas deben ser tenidas en consideración al momento de deliberar. Para lograr esto en tiempo real se propone el uso del algoritmo RETE, el cual construye una red de dependencias, representada en un árbol dirigido, conocida como "Red de Activación de Condiciones". El uso de este árbol determina que reglas deben ser consideradas y de esta manera se sabe que metas deben ser evaluadas, sin la necesidad de repetir evaluaciones innecesarias como se hacía en BDI-BESA. Para el caso de esta arquitectura las metas son específicas al caso de aplicación. Cabe resaltar que este proceso de generar y consultar el árbol indexado se hace en el Módulo de activación de Metas.

Por un lado,  $G_{BDI} = \{g_1 \dots g_n\}$  representa la lista de  $n$  metas relacionadas con los objetivos del agente de robótica de asistencia social. Cada meta  $g_x$  de esta lista puede ser categorizadas según una pirámide de metas propuesta por la arquitectura BDI-CHA: Supervivencia, Obligación, Oportunidad, Facilitadoras y Life-like. Bajo la arquitectura SocialRobotBesa, una meta  $g_x$  puede tener múltiples categorizaciones entre: Meta de Recuperación de errores, Meta de Manejo de Excepciones, Meta de Riesgos Éticos, Meta de Riesgos y Meta de Servicio.  $AF_{G_{BDI}} = \{af_{g_1} \dots af_{g_n}\}$  representa el conjunto de las funciones de activación de cada meta. Cada meta  $g_x$  tiene asociada una función de activación donde  $af_{g_x} \in [0,1]$ . Cuando el valor de la función  $af_{g_x}$  es mayor a cierto umbral la meta se considera activa y continua en el proceso de deliberación. Por ejemplo, en el caso de una meta como *Contar Chiste*, la función de activación se considera activada si las emociones del usuario son negativas.  $CF_{G_{BDI}} = \{cf_{g_1} \dots cf_{g_n}\}$  representa el conjunto de las funciones de contribución de cada meta. Cada meta  $g_x$  tiene asociada una función de contribución donde  $cf_{g_x} \in [0,1]$ . Este valor indica la prioridad que se le da a la meta en relación con las demás metas de su categoría, con el fin de resolver posibles conflictos.

#### Metas Latentes

Adicionalmente,  $G_L = \{lg_1 \dots lg_n\}$  representa la lista de  $n$  metas latentes que pueden modular o afectar a las metas  $G_{BDI}$ . Una meta latente  $lg_x$  puede incluir en la activación o contribución de otras metas latentes o metas  $G_{BDI}$ . En este contexto  $AF_{G_L} = \{af_{lg_1} \dots af_{lg_n}\}$  representa la función de activación de la meta latente y  $CF_{G_L} = \{cf_{lg_1} \dots cf_{lg_n}\}$  representa la función de contribución de la meta latente. Se utilizó un sistema basado en arboles donde cada nodo es una meta latente y las hojas del árbol son metas BDI.  $W_G = \{w_{lg_1, g_1} \dots w_{lg_n, g_k}\}$  representa los pesos entre las distintas metas.

Cada arista del árbol representa un peso, que al ser multiplicado por la contribución de la meta latente, determina el incremento porcentual de la contribución de las metas normales. Para calcular el incremento porcentual de una meta normal  $m$ , se calcula la sumatoria del producto entre el peso de cada meta latente asociada  $i$  y su respectiva contribución ( $cf_{lg_i}$ ). El cálculo de la contribución de la meta BDI  $m$  ( $cf'_{g_1}$ ) a partir de su contribución original ( $cf_{g_1}$ ) y el incremento porcentual se muestra a continuación:

$$cf'_{g_1} = cf_{g_1} * (1 + \sum_{i=1}^n w_{i,m} * cf_{lg_i})$$

Cabe resaltar que este cambio en la contribución de una meta BDI por parte de la meta latente solo se da cuando la meta latente relacionada esta activa. La configuración para este caso particular se muestra en la Ilustración 24. Metas Latentes. Cabe resaltar que la meta latente se muestra en un romboide. Además, la contribución de cada meta se muestra entre paréntesis. Para el caso de las metas de servicio, se muestra su contribución original y luego la contribución después de aplicar la meta latente.



Ilustración 24. Metas Latentes

En este ejemplo, la meta latente aumenta la contribución de la meta de servicio *MusicoTerapia*, al tener un peso de 1 y una activación de 1, lo que implica un aumento porcentual del 100%. En contraste, para las demás metas, al tener un peso de -1, implica una reducción del 100%. De esta manera, cuando la meta de *MusicoTerapia* esté activa se impondrá sobre las demás metas así la preferencia del paciente sea *Cuentería*, *Momento Espiritual* o incluso si llega la hora de los medicamentos.

#### Roles Interactivos

Para modular o cambiar los pesos del árbol de Metas, se introduce el concepto de Roles interactivos los cuales son el valor RO de la tupla del analizador de metas. Un rol interactivo (ro), se representa con la siguiente tupla:

$$(Auth_g, W, Per)$$

Donde  $Auth_g = \{auth_{g_1} \dots auth_{g_n}\}$  puede representar cambios en las autorizaciones de ciertas metas,  $W = \{w_{g_i, g_k} \dots w_{g_i, g_k}\}$  muestra los nuevos pesos del árbol de metas y  $Per$  representa los cambios en la personalidad del robot.

En general, el conjunto de metas  $D_{BDI}$  contará con metas que hayan sido indexadas por el Módulo de activación de metas, cuyo valor de activación haya sido mayor al umbral y cuyo valor de contribución compuesto las priorice de manera que puedan ser convertidas en intención.

#### Motor de Deliberación

El motor de deliberación determina tres aspectos fundamentales de las metas que pueden ser intención. En primer lugar, el motor determina posibles recursos de conflictos entre las metas activas  $D_{BDI}$  del agente. Este trabajo es realizado por el componente Gestor de Recursos. Se consideran tres posibles conflictos entre metas:

- *Conflicto de Recursos entre metas de diferente categoría:* El motor permite que la meta con mayor categoría pase a ser intención. En este caso, la meta que deja ser intención debe ser expropiada y se envía un evento de expropiación  $eexp_G$  al razonador de medios y fines.
- *Conflicto de Recursos entre metas de misma categoría:* El motor se vale del valor de contribución de cada meta y selecciona la meta de mayor contribución para volverse intención.
- *Conflicto de Recursos entre metas, pero el recurso es optativo:* Si una meta quiere ser intención pero requiere un recurso como opcional o no necesario y este recurso ya está siendo usado, el monitor de recursos permite que la meta se vuelva intención.

Para determinar si hay un conflicto de recursos, este componente funciona la lista de recursos y su estado en el agente de *Creencias* y obtiene la lista recursos necesarios de la descripción de la meta. Esto se ve representado en la interacción 2.

En segundo lugar, el motor gestiona el nivel de autonomía de metas activas  $D_{BDI}$ , cuyo conflicto de recursos haya sido resuelto. El componente Gestor de autonomía es el responsable de esto y tiene como creencias la tupla  $(Crf_{G_{BDI}}, Ref_{G_{BDI}}, Auth_g)$ , donde  $Crf_g$  representa las funciones que determinan la criticidad de la meta  $Ref_g$  representa la función de responsabilidad de la meta y  $Auth_g$  representa si la meta ha sido autorizada por un usuario relacionado con el cuidado del usuario para ser intención.

Por otro lado,  $Crf_{G_{BDI}} = \{crf_{g_1} \dots crf_{g_n}\}$  representa el conjunto de las funciones de activación de cada meta. Cada meta  $g_x$  tiene asociada una función de criticidad donde  $crf_{g_x} \in [0,1]$ . Cuando el valor de la función  $crf_{g_x}$  es alto es posible que el agente no tenga autonomía suficiente para realizar la meta. Por ejemplo, en el caso de una meta como *Hacer Ejercicio con el adulto mayor*, la función de criticidad tiene un valor alto, ya que la tarea puede ser fatal para el usuario en caso de ser realizada sin supervisión.

De igual manera,  $Ref_{G_{BDI}} = \{ref_{g_1} \dots ref_{g_n}\}$  representa el conjunto de las funciones de activación de cada meta. Cada meta  $g_x$  tiene asociada una función de responsabilidad donde  $ref_{g_x} \in [0,1]$ . Cuando el valor de la función  $ref_{g_x}$  es alto es posible que el agente no tenga autonomía suficiente para realizar la meta. Por ejemplo, en el caso de una meta como *Dar Medicamento al adulto mayor*, la función de responsabilidad tiene un valor alto, ya que si los medicamentos son dados erróneamente un ser humano debe asumir la responsabilidad en ese error. Por último,  $Auth_{G_{BDI}} = \{auth_{g_1} \dots auth_{g_n}\}$  representa el conjunto de las autorizaciones de cada meta. Cada meta  $g_x$  tiene asociada un valor de autorización donde  $auth_{g_x} \in [True, False]$ . Cuando el valor de  $auth_{g_x}$  es verdadero la meta es considerada para volverse intención. En general, se consideró que si el nivel de autonomía era alto (criticidad y responsabilidad bajas), el agente puede actuar autónomamente. Si el nivel de autonomía era medio, el agente debía solicitar autorización a un tercero experto, de manera que actuaba semi autónomamente. Finalmente, si el nivel de autonomía era bajo (la meta BDI ha sido desactivada por un tercero), se considera que la meta no debe realizarse.

En tercer lugar, el motor de deliberación gestiona el tiempo de las metas de servicio. Este tipo de metas puede estar sujeto a un marco de tiempo particular y el motor de deliberación hace de *Timer* para que la meta sepa que esta próxima a finalizar.

Una vez finalizada la deliberación, se obtiene una tupla  $(I_{bdi})$ , la cual representa las intenciones del agente. El componente de motor de deliberación también se comunica con el agente de explicabilidad para reportar los cambios en las intenciones del agente  $(I_{bdi})$ , como se ve en la interacción 6.

### Razonador de Medios y Fines

El razonador de medios y fines ejecuta planes a partir de las intenciones de  $(I_{bdi})$  del agente. El módulo esa en la capacidad de ejecutar tareas que requieren de realizar cálculos o procesar datos. Así mismo, puede haber tareas que requieran de la colaboración de los agentes de Cooperación, del agente de Acción o del agente de Creencias. De esta manera, la salida de este módulo está representada como:

$$(T_{RA}, T_{UB}, T_{CA}, T_{CH}, T_E, PS)$$

Donde:

- $T_{RA}$  son las tareas del plan que involucran una acción del robot, como lo puede ser extender los brazos o bailar.
- $T_{UB}$  son las tareas relacionadas con actualizar las creencias del agente de Creencias. Esto puede ser actualizar el estado del mundo o el estado de los recursos.
- $T_{CA}$  son las tareas relacionadas con la interacción con otros agentes. Esto podría ser solicitar a un robot aspiradora que limpie la casa o solicitar información a un agente en un wearable del usuario. Esto se ve representado en la interacción 5.
- $T_{CH}$  son las tareas de cooperación con otros seres humanos, lo que puede involucrar solicitar la intervención de un cuidador en un adulto mayor o incluso pedirle a un familiar que le llame. Esto se ve representado en la interacción 6.
- $T_E$  son las tareas de explicabilidad, donde se le solicita al agente de explicabilidad que envíe los registros al agente interactivo.
- $PS$  es el conjunto de estados de planes en ejecución. En este se lleva registro de las tareas en ejecución y las acciones relacionadas a esta. Una vez el agente de acción envía un evento de finalización  $ef_a$ , se actualiza el estado de la tarea relacionada a la acción  $a$ .

Cuando este componente recibe un enveto de expropiación  $eexp_G$ , lo convierte en una serie de eventos de expropiación  $eexp_a$  que son transmitidos al agente de acción. Adicionalmente, el estado de las tareas y del plan se deja como interrumpido. En este momento, los planes se interrumpen de manera definitiva. Sin embargo, para trabajo futuro podría considerarse la posibilidad de permitir que los planes sean interrumpidos temporalmente.

### Agente de Acción

El agente de *Acción* es responsable de ejecutar las acciones del robot y de enriquecerlas según su estado emocional y sus creencias del usuario. Este agente está compuesto de dos módulos: el módulo *Modulador de Acciones* y el módulo *Ejecutor de Acciones*. Cabe resaltar que la descripción de este agente fue adaptada del trabajo de Bravo [3].

Las creencias de este agente se representan con la siguiente tupla:

$$\{A, S, EMMB, PR\}$$

Donde  $A = \{a_1..a_n\}$ , es el conjunto de acciones descritas por primitivas secuenciales,  $S = \{a_{1,t}: \{p_1: state \dots p_n: state\} \dots a_{n,t}: \{p_1: state \dots p_n: state\} \dots\}$ , representa el estado de las primitivas enviadas por el agente hacia los *Proveedores de Servicio*, cabe resaltar que  $a_{1,t}$  representa una acción asociada a una tarea t. Adicionalmente, *EMMB* es una copia del estado emocional del agente y PR es un conjunto de todos los perfiles activos de los usuarios. Estos dos últimos elementos son transmitidos en la interacción 2 por un evento informativo *einfn* generado por el agente de *Creencias*.

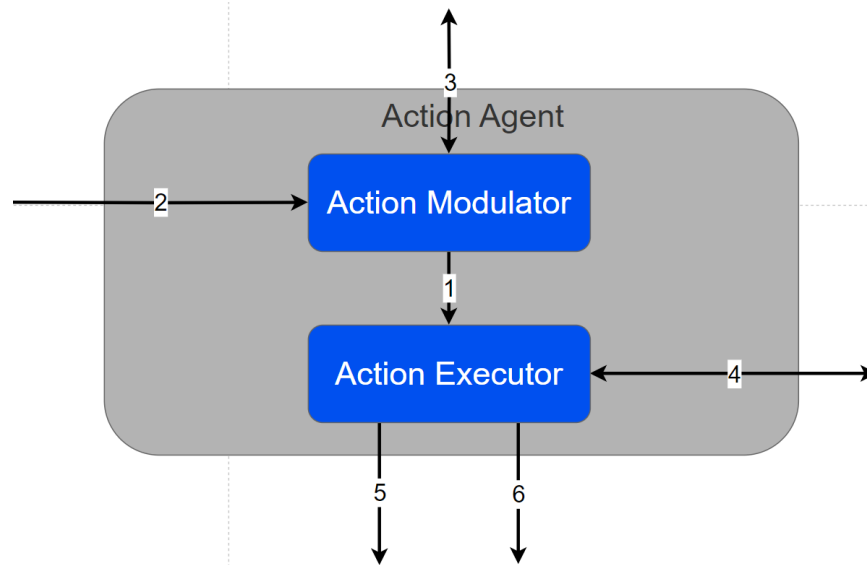


Ilustración 25. Agente Interactivo

El propósito del módulo *Modulador de Acciones* es el de enriquecer las acciones del robot de asistencia al agregar interpretación emocional a partir del estado emocional actual del robot y criterios de modulación a partir del perfil del usuario. además tiene la función de traducir las acciones a primitivas del sistema. Una acción  $a_x$ , es enviada por la interacción 3 por el *Razonador de Medios y Fines* y puede estar compuesta de varias primitivas como se muestra a continuación.

$$a_x = \{p_1, p_2, p_n\}$$

Una vez la acción ha sido convertida a primitivas, cada primitiva puede ser modulada emocionalmente o según el perfil del usuario. De esta manera, la salida de este módulo se puede describir como el conjunto  $P = \{(p_1, \{pp_1..pp_k\}), (p_n, \{pp_1..pp_k\})\}$ , donde  $pp_k$  es el parámetro de modulación n de la primitiva  $p_n$ .

El proceso de modulación es dependiente del contexto aplicación de la arquitectura y puede soportar cantidad indefinida de parámetros, así como múltiples tipos de variable. Cabe mencionar que es posible que existan primitivas que no deban ser moduladas. De igual manera, el cómo se quiera determinar el valor del parámetro es dependiente a la implementación.

Respecto al ejecutor de acciones, este tiene como función el enviar las primitivas a los agentes *Proveedor de Servicio* como se muestra en la interacción 6. El módulo toma como entrada el conjunto P, luego por cada primitiva  $p_i$  la agrega al conjunto S con el estado iniciado. Los estados de las primitivas se muestran a continuación:

Estado	Descripción
Iniciado	Representa que la primitiva ha sido registrada en el módulo ejecutor de acciones.
En ejecución	Representa que la primitiva ha sido enviada al agente <i>Proveedor de Servicio</i> específico.
Finalizado	Representa que la primitiva ha sido finalizada por al agente <i>Proveedor de Servicio</i> específico.
Interrumpido	Representa que el agente de <i>Motivación</i> ha enviado un evento de expropiación que ha interrumpido la primitiva.

Tabla 16. Estado de las Primitivas

Cuando todas las primitivas  $p_i$  asociadas a una acción  $\alpha_{1,t}$  han finalizado, el componente ejecutor de acciones envía un evento de finalización  $ef_\alpha$  de la acción al *Razonador De Medios Y Fines*, como se muestra en la interacción 3. Además, los eventos de expropiación  $exp_\alpha$ , llegan a través de la interacción 4. Es necesario mencionar que existen ciertas primitivas del sistema, como la primitiva *say*, que involucran al agente *Interactivo* y no al agente *Proveedor de Servicio* directamente. Esto se ve representado en las interacciones 5 de la Ilustración 25. Agente Interactivo.

## Cambiando de Robot

Para cambiar de robot bajo una implementación de la arquitectura es necesario cambiar ciertas configuraciones y instanciar algunas interfaces de interacción. En primer lugar, será necesario cambiar los adaptadores de los agentes *Proveedor de Servicio* y cada uno de sus servicios, de manera que las peticiones de servicio puedan ser traducidas correctamente a la interfaz de servicio del robot.

Por otro lado, será necesario modificar las creencias de los recursos físicos del usuario de manera que el sistema pueda tener en cuentas las verdaderas capacidades del hardware.

## Cambiando de Contexto de Aplicación

En caso de cambiar proceso de aplicación, será necesario adaptar múltiples puntos de cambio dentro de la arquitectura de control para adaptarse al nuevo tipo de usuario. Em primer lugar, es necesario adaptar el modelo emocional del usuario para reflejar las emociones importantes en el contexto de la interacción. Asi mismo, es pertinente adaptar el contexto de conversación en las creencias para manejar nuevos tipos de interacción verbal. Por último, es posible extender la información enviada al agente interactivo de manera que los nuevos eventos puedan ser procesados sin problema.

## Agregar una Meta de Servicio

Para agregar una meta de servicio al sistema es necesario que esta extienda de la clase padre de metas de servicio. Asi mismo, será necesario crear una clase hija de la clase *ServiceContext*. Con esto los datos necesarios para ejecutar los servicios estarán presentes en las creencias.



## Bibliografía

- [1] M. A. Kohl, K. Baum, M. Langer, D. Oster, T. Speith, y D. Bohlender, «Explainability as a Non-Functional Requirement», en *2019 IEEE 27th International Requirements Engineering Conference (RE)*, Jeju Island, Korea (South): IEEE, sep. 2019, pp. 363-368. doi: 10.1109/RE.2019.00046.
- [2] A. Traldi, F. Bruschetti, M. Robol, M. Roveri, y P. Giorgini, «Real-Time BDI Agents: a model and its implementation». arXiv, 2 de mayo de 2022. doi: 10.48550/arXiv.2205.00979.
- [3] F. Bravo, «DRAMABOT, A COGNITIVE MULTIAGENT ARCHITECTURE FOR IMPLEMENTING EDUCATIONAL DRAMA TECHNIQUES USING ROBOT ACTORS», Pontificia Universidad Javeriana, Bogota DC, 2019.
- [4] M. Torres Moreno, O. Garavito, y J. OCAMPO, «Metodología de Pruebas para Sistemas Multi-Agentes (SMA) integrada a AOPOA», ene. 2008.
- [5] A. González, R. Angel, y E. González, «BDI concurrent architecture oriented to goal management», en *2013 8th Computing Colombian Conference (8CCC)*, ago. 2013, pp. 1-6. doi: 10.1109/ColombianCC.2013.6637540.
- [6] S. Elbeleidy, T. Mott, y T. Williams, «Practical, Ethical, and Overlooked: Teleoperated Socially Assistive Robots in the Quest for Autonomy», en *2022 17th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, mar. 2022, pp. 577-587. doi: 10.1109/HRI53351.2022.9889573.
- [7] J. P. Boada, B. R. Maestre, y C. T. Genís, «The ethical issues of social assistive robotics: A critical literature review», *Technology in Society*, vol. 67, p. 101726, nov. 2021, doi: 10.1016/j.techsoc.2021.101726.
- [8] A. Ortony, G. Clore, y A. Collins, «The Cognitive Structure of Emotion», *The Cognitive Structure of Emotion*. Accedido: 18 de abril de 2021. [En línea]. Disponible en: [https://www.researchgate.net/publication/202304316\\_The\\_Cognitive\\_Structure\\_of\\_Emotion](https://www.researchgate.net/publication/202304316_The_Cognitive_Structure_of_Emotion)
- [9] J. Alcalá Vásquez, «Reconocimiento multimodal del estado emocional de un niño en un contexto educativo», 2017, doi: 10.11144/Javeriana.10554.39889.
- [10] F. Kaptein *et al.*, «A Cloud-based Robot System for Long-term Interaction: Principles, Implementation, Lessons Learned», *J. Hum.-Robot Interact.*, vol. 11, n.º 1, p. 8:1-8:27, oct. 2021, doi: 10.1145/3481585.
- [11] R. D. Benedictis, A. Umbrico, F. Fracasso, G. Cortellessa, A. Orlandini, y A. Cesta, «A dichotomic approach to adaptive interaction for socially assistive robots», *User Model User-Adap Inter*, nov. 2022, doi: 10.1007/s11257-022-09347-6.
- [12] M. Maroto Gómez, Á. Castro-González, J. Castillo, M. Malfaz, y M. Salichs, «An adaptive decision-making system supported on user preference predictions for human-robot interactive communication», *User Modeling and User-Adapted Interaction*, pp. 1-45, abr. 2022, doi: 10.1007/s11257-022-09321-2.
- [13] A. A. De la Peña, «RoboAct: Modelo de control autónomo y cooperativo para el Teatro Robótico». Accedido: 23 de abril de 2020. [En línea]. Disponible en: <https://repository.javeriana.edu.co/bitstream/handle/10554/15082/PenaSantanaAndresArmandodela2014.pdf?sequence=1&isAllowed=y>
- [14] S. Chen, R. Arkin, J. Borenstein, A. R. Wagner, y A. Melo Cruz, «Case-based robotic architecture with multiple underlying ethical frameworks for human-robot interaction», en *Value-Sharing between Humans and Robots*, CLAWAR Association Ltd., jul. 2022, pp. 115-120. doi: 10.13180/icres.2022.18-19.07.005.

- [15] P. Bremner, L. A. Dennis, M. Fisher, y A. F. Winfield, «On Proactive, Transparent, and Verifiable Ethical Reasoning for Robots», *Proceedings of the IEEE*, vol. 107, n.º 3, pp. 541-561, mar. 2019, doi: 10.1109/JPROC.2019.2898267.
- [16] J. J. Paez Rodríguez, «Human-Robot Scaffolding - Arquitectura BDI para el desarrollo de habilidades de solución de problemas», Pontificia Universidad Javeriana, 2019.
- [17] B. Steunebrink, M. Dastani, J.-J. Ch, y J. Meyer, «The OCC model revisited», ene. 2009.