

CIS2010CP05

COBOT: Teatro robótico educativo orientado a las emociones y el realismo

Karen Juliana Celis Buitrago
Nicolás Andrés Cervantes Pineda
Pedro Martin Escobar Gómez

PONTIFICIA UNIVERSIDAD JAVERIANA
FACULTAD DE INGENIERIA
SYSTEMS ENGINEERING PROGRAM
BOGOTÁ, D.C.
2020

CIS2010CP05

COBOT: Teatro robótico educativo orientado a las emociones y el realismo

Autores:

Karen Juliana Celis Buitrago
Nicolás Andrés Cervantes Pineda
Pedro Martin Escobar Gómez

UNDERGRADUATE FINAL PROJECT REPORT PERFORMED IN ORDER TO ACCOMPLISH
ONE OF THE REQUIREMENTS FOR THE SYSTEMS ENGINEERING DEGREE

Director

Flor Ángela Bravo Sánchez

Asesor

Enrique González Guerrero

Jurados del Proyecto final

Cesar Julio Bustacara Medina

Eduardo Andrés Gerlein Reyes

PONTIFICIA UNIVERSIDAD JAVERIANA
FACULTAD DE INGENIERIA
SYSTEMS ENGINEERING PROGRAM
BOGOTÁ, D.C.
Noviembre, 2020

**PONTIFICIA UNIVERSIDAD JAVERIANA
FACULTAD DE INGENIERIA
SYSTEMS ENGINEERING PROGRAM**

President of the Pontificia Universidad Javeriana

Jorge Humberto Peláez Piedrahita, S.J.

Dean of School of Engineering

Ing. Lope Hugo Barrero Solano

Head of the Systems Engineering Program

Ing. Alexandra Pomares Quimbaya

Head of the Systems Engineering Department

Ing. Efraín Ortiz Pabón

Artículo 23 de la Resolución No. 1 de junio de 1946

“La Universidad no se hace responsable de los conceptos emitidos por sus alumnos en sus proyectos de grado. Sólo velará porque no se publique nada contrario al dogma y la moral católica y porque no contengan ataques o polémicas puramente personales. Antes bien, que se vean en ellos el anhelo de buscar la verdad y la Justicia”

AGRADECIMIENTOS

Le agradecemos a la Pontificia Universidad Javeriana por la oportunidad que nos brindaron para crecer tanto de forma personal, como profesional, así como el acogernos para ser parte de esta gran familia javeriana. Le agradecemos también a los docentes que estuvieron presentes durante esta etapa académica, y las lecciones que nos brindaron que seguramente serán útiles para nuestras vidas.

A nuestra directora, le agradecemos el esfuerzo, acompañamiento, comprensión y paciencia brindada, ya que fue un pilar fundamental en la creación y desarrollo de este proyecto, y con quien esperamos encontrarnos nuevamente en un futuro cercano.

Agradecerles también a nuestras familias, que durante todo este proceso fueron un apoyo continuo y sin ellos no habríamos logrado llegar aquí. Así mismo a nuestros compañeros de trabajo, con quienes tuvimos la oportunidad de trabajar durante nuestra estadía en la universidad y desarrollamos vínculos que permanecerán en un futuro próximo.

Por último, agradecer a los niños y personas involucradas que brindaron su tiempo para colaborar con el desarrollo de este trabajo ya que sus aportes permitieron el cumplimiento del objetivo del proyecto.

TABLA DE CONTENIDO

I-	INTRODUCCIÓN	1
II-	DESCRIPCIÓN GENERAL.....	3
1.	PROBLEMA Y OPORTUNIDAD.....	3
1.1.	<i>Contexto del problema</i>	3
1.2.	<i>Formulación del problema</i>	4
1.3.	<i>Propuesta de la solución</i>	4
1.4.	<i>Justificación de la solución</i>	6
2.	DESCRIPCIÓN DEL PROYECTO	7
2.1.	<i>Objetivo General</i>	7
2.2.	<i>Objetivos Específicos</i>	8
2.3.	<i>Entregables, estándares, y justificación</i>	8
III-	CONTEXTO DEL PROYECTO	9
1.	MARCO TEÓRICO.....	9
2.	ANÁLISIS DE CONTEXTO	14
IV-	ANÁLISIS DEL PROBLEMA	16
1.	REQUERIMIENTOS	16
2.	RESTRICCIONES	17
3.	ESPECIFICACIÓN FUNCIONAL.....	18
V-	DISEÑO DE LA SOLUCIÓN	19
1.	FUNCIONAMIENTO GENERAL DEL SISTEMA	19
1.1.	<i>Sistema COBOT:</i>	20
1.2.	<i>Sistema del robot:</i>	27
1.3.	<i>Aplicativo móvil de pruebas de COBOT:</i>	27
2.	VISTA LÓGICA DEL SISTEMA.....	29
3.	VISTA FÍSICA DEL SISTEMA	33
4.	HERRAMIENTAS DE DESARROLLO	34
VI-	DESARROLLO DE LA SOLUCIÓN	36
1.	METODOLOGÍA.....	36

1.1.	<i>Estructura de ramas</i>	36
2.	IMPLEMENTACIÓN	37
2.1.	<i>Recepción de mensajes</i>	37
2.2.	<i>Modulación de acciones</i>	39
2.3.	<i>Ejecución de acciones</i>	42
2.4.	<i>Generación de trayectorias</i>	44
2.5.	<i>Prototipo del aplicativo móvil de pruebas</i>	45
VII-	RESULTADOS	47
1.	PRUEBAS REALIZADAS.....	47
1.1.	<i>Pruebas unitarias</i>	48
1.2.	<i>Pruebas de concepto</i>	48
1.3.	<i>Prueba UAT (User Acceptance Test)</i>	52
VIII-	CONCLUSIONES	58
1.	ANÁLISIS DE IMPACTO DEL PROYECTO	58
1.1.	<i>Corto plazo</i>	58
1.2.	<i>Mediano plazo</i>	58
1.3.	<i>Largo plazo</i>	58
2.	CONCLUSIONES.....	59
3.	TRABAJO FUTURO.....	60
IX-	REFERENCIAS	61
X-	APÉNDICES	65

LISTA DE FIGURAS

Figura 1.	Proceso para realizar obras de teatro actual.....	5
Figura 2.	Proceso para realizar una obra de teatro utilizando el sistema COBOT.	6
Figura 3.	Diagrama de casos de uso para el proyecto COBOT.	19
Figura 4.	Diagrama arquitectural de COBOT.....	20
Figura 5.	Sección agente de razonamiento diagrama COBOT.....	21
Figura 6.	Pirámide de prioridad de acciones.....	22
Figura 7.	Sección agente de creencias diagrama COBOT.....	22
Figura 8.	Sección agente de acción diagrama COBOT.....	24
Figura 9.	Ecuación que representa la intensidad de los parámetros de los comandos modulables.....	25
Figura 10.	Sección aplicativo móvil diagrama COBOT.....	28

Figura 11. <i>Prototipo interfaz de envío de acciones</i>	29
Figura 12. <i>Vista lógica del sistema para el proyecto COBOT</i>	31
Figura 13. <i>Vista física del sistema para el proyecto COBOT</i>	33
Figura 14. <i>Distribución y orden de creación de ramas para desarrollo del servidor</i>	37
Figura 15. <i>Ejecución del servidor durante recepción de datos</i>	38
Figura 16. <i>Arribo de mensajes de signos de vida al servidor</i>	38
Figura 17. <i>Estructura del archivo del perfil del robot</i>	39
Figura 18. <i>Ecuación que representa la intensidad emocional del robot</i>	41
Figura 19. <i>Pruebas de recepción de mensajes de modulación en consola</i>	41
Figura 20. <i>Representación de matriz y grafo usados para el trabajo de grado</i>	45
Figura 21. <i>Pantalla principal envío de acciones</i>	46
Figura 22. <i>Gráficos demográficos de los usuarios que participaron en la prueba final</i>	53
Figura 23. <i>Fotografías tomadas a los usuarios durante la ejecución de la prueba final</i>	54

LISTA DE TABLAS

Tabla 1. <i>Entregables y estándares del proyecto COBOT</i>	9
Tabla 2. <i>Tabla de análisis sobre gestión de acciones concurrentes</i>	12
Tabla 3. <i>Requerimientos funcionales del sistema COBOT</i>	17
Tabla 4. <i>Restricciones del sistema COBOT</i>	18
Tabla 5. <i>Tipos de comandos para la modulación de acciones</i>	25
Tabla 6. <i>Definición componentes para la vista lógica del sistema en el proyecto COBOT</i>	32
Tabla 7. <i>Definición componentes para la vista física del sistema en el proyecto COBOT</i>	34
Tabla 8. <i>Herramientas de trabajo utilizadas para el desarrollo del proyecto COBOT</i>	36
Tabla 9. <i>Resultados pruebas unitarias en el sistema COBOT</i>	48
Tabla 10. <i>Edad y frecuencia de las personas encuestadas</i>	49
Tabla 11. <i>Resultados formularios</i>	50
Tabla 12. <i>Resultados ejecución momento número uno</i>	54
Tabla 13. <i>Resultados ejecución momento número dos</i>	54
Tabla 14. <i>Resultados ejecución momento número tres</i>	55
Tabla 15. <i>Resultados ejecución momento número cuatro</i>	55
Tabla 16. <i>Resultados ejecución momento número cinco</i>	55
Tabla 17. <i>Resultados ejecución momento número seis</i>	55
Tabla 18. <i>Resultados preguntas sobre ejecución de acciones</i>	56

RESUMEN

Durante los últimos años se ha experimentado y realizado proyectos en torno al teatro robótico, el cual también incurre en la educación como un medio de apoyo al aprendizaje. Un gran reto del teatro robótico es el desarrollo de robots actores expresivos y creíbles para que puedan transmitir adecuadamente a la audiencia el mensaje deseado.

Ante este reto, se desarrolló el sistema llamado COBOT, cuyo objetivo se centra en mejorar la expresión de acciones de los robots actores en las obras de teatro. Para esto se desarrolló el sistema que, partiendo del estado emocional del robot, el entorno y las capacidades físicas de este, modula las acciones emocionalmente para luego ser interpretadas. La solución fue validada con usuarios entre los 7 y los 11 años.

ABSTRACT

During recent years, projects around robotic theater had been made. They involve education as a mean of supporting learning. A great challenge of robotic theater is the development of credible and expressive robot actors so that they can adequately convey the desired message to the audience.

Faced with this challenge, the system called COBOT was developed, whose objective is to improve the expression of actions of robot actors in plays. In order to reach that goal, the system was developed, which based on the robot's emotional state, the environment and its physical capacities, modulates the actions emotionally to be afterwards interpreted. The solution was validated with users between 7 and 11 years old.

I- INTRODUCCIÓN

La robótica ha experimentado un progreso notable con el paso de los años, modificando la forma como se realizan tareas o se resuelven problemas. Tal es así que, de acuerdo con la Federación Internacional de Robótica, en el 2019 este mercado se encontró avaluado en 25 billones de dólares [1] generando un crecimiento del 32% en comparación al año anterior, con potencial de mejora para este año 2020 debido a la situación que se presenta a causa del COVID-19.

Ahora bien, dentro de los diferentes enfoques a los que se ha dedicado la robótica desde su creación, se ha incluido uno nuevo, el cual es el teatro robótico. Tal enfoque consiste en el uso de robots en obras de teatro ya sea interactuando con humanos o entre otros robots al seguir un guion [2]. Además de ello, el teatro robótico puede tomarse como un precedente para la creación de nuevas propuestas referentes a la ejecución del teatro o diferentes artes.

Una de las áreas donde se puede aplicar el teatro robótico es en la educación, debido a que sirve como apoyo en los procesos de enseñanza y aprendizaje. Para ello se requiere que el público identifique la emoción con la que los robots realizan un conjunto de acciones, haciendo uso de herramientas de software y del hardware de estos. Los robots deben interpretar los personajes de forma adecuada y acorde con lo que se establece en un guion.

La meta del proyecto es desarrollar una solución de software que implemente los comportamientos de robots al interpretar personajes de una obra de teatro. Estos podrán realizar acciones, las cuales variarán su modo de acuerdo con la emoción por la que pase el robot al ejecutarlas. Esto permitirá que niños de edades entre 7 y 11 años comprendan el mensaje que se desea transmitir y perciban dichas emociones. Este proyecto supone una solución innovadora a la estrategia del teatro robótico educativo.

En consecuencia, el presente documento describe a detalle la solución implementada. En el capítulo dos se presenta la descripción general de este proyecto donde se exponen las características y el contexto para la solución. En el capítulo tres se realiza un análisis contextual y

se recopilan las lecciones aprendidas en trabajos relacionados. En el capítulo cuatro se analiza la problemática en términos de sus requerimientos, restricciones durante el desarrollo del trabajo y las especificaciones funcionales.

En el capítulo cinco, se detalla el diseño de la solución del sistema, donde se expondrán los componentes que lo conforman, así mismo se definen las vistas en modelos arquitecturales y las herramientas utilizadas para la implementación. El capítulo seis describe el desarrollo de la solución, explicando el uso dado a la metodología SCRUM, la estructura de ramas de desarrollo de software y la exposición de las funcionalidades principales implementadas. El capítulo siete recopila toda la información de las pruebas realizadas, tanto de integración como con el público objetivo, y se analizan los resultados obtenidos. Por último, se establecen conclusiones y aspectos del trabajo por mejorar, que darán un impacto a corto, medio y largo plazo.

II- DESCRIPCIÓN GENERAL

1. Problema y oportunidad

1.1. Contexto del problema

Los robots hoy en día participan activamente en el apoyo de actividades de diferentes sectores de la sociedad, uno de ellos es la educación donde brindan herramientas para apoyar el proceso de aprendizaje fomentando el crecimiento y la creación de nuevos conocimientos, habilidades y experiencias para el educando [3]. Todo el trabajo realizado con robots es gracias a los avances diarios de la tecnología la cual busca día a día ser mejor para la sociedad.

Partiendo de la inmersión de los robots en diferentes actividades de la sociedad, en cuanto al sector del arte se han desarrollado diferentes proyectos con enfoques teatrales, donde existen robots actores que interpretan personajes tales como humanos, animales u objetos animados, y en cuyas obras el robot actúa solo o incluso con humanos u otros robots actores [4] [5]. Actualmente se busca crear una obra de teatro que logre captar y mantener la atención del espectador, con el fin de que el robot transmita el mensaje y este se entienda casi de la misma manera como si lo hiciera un actor humano [6].

En las áreas de la educación y la robótica se están explorando diferentes propuestas que buscan la integración de estos dos ámbitos, para construir herramientas innovadoras que apoyen el proceso de aprendizaje aplicando estrategias como el teatro robótico educativo. Esta estrategia es un medio de enseñanza y aprendizaje bien establecido para el desarrollo de habilidades cognitivas, sociales y emocionales, utilizado ampliamente para apoyar los procesos de enseñanza y aprendizaje de materias no técnicas y técnicas [7] [8].

En la Pontificia Universidad Javeriana se realizan proyectos como: “Quyca: prueba de concepto” financiado por la dirección de innovación donde se está desarrollando una plataforma de dramatización con robots para propósitos educativos que incluye una metodología educativa, un entorno intuitivo para la creación de guiones y robots actores inteligentes y expresivos.

vos. Así mismo, alrededor de este tema en la Universidad Javeriana se han desarrollado diferentes trabajos de pregrado y posgrado como: “Quyca-bot: creación de guiones para teatro robótico” y “Creación de herramientas de programación para drama robótico”, que buscan dar respuesta a diferentes problemáticas en torno al teatro robótico educativo.

1.2. Formulación del problema

Es importante mencionar que cuando el espectador asiste al teatro no solo va a ver un espectáculo, sino que también este transmite emociones, sentimientos e influye en el espectador mediante lo que expresa el actor [9]. Al adaptar las interpretaciones teatrales al contexto robótico la expresividad del robot es fundamental para que el actor logre crear un vínculo con la audiencia (educandos) y esta comprenda el mensaje, sentimientos y diferentes emociones.

Aunque existe una variedad de avances en teatro robótico profesional, en el educativo poco se ha desarrollado en torno a herramientas de software que faciliten la creación de obras de teatro donde los actores tengan comportamientos expresivos o creíbles. En los proyectos anteriormente mencionados se han utilizado estrategias como tele operar los robots, programarlos por separado o en la mayoría de los casos del teatro robótico tan sólo se controla el movimiento y el diálogo básico, es decir, movimientos predefinidos del robot, pero no se explora a detalle la manipulación de la expresividad al momento de ejecutar las acciones.

1.3. Propuesta de la solución

Como solución a la problemática se propuso implementar un sistema llamado COBOT, el cual, utiliza la información del contexto de la obra de teatro, un personaje y un conjunto de instrucciones con acciones a ejecutar para uno o varios actores robóticos, y hace lo siguiente:

- Modular emocionalmente las acciones: esto se logra mediante la emoción que el usuario proporciona y de las especificaciones propias del robot escogido.
- Activar acciones a modo de signos de vida: COBOT permite que el actor robótico se mantenga activo durante el tiempo cuando se encuentre en escena sin recibir acciones del usuario.

- Activar comportamientos emergentes: cuando se alcanza un umbral emocional (p. ej. mucha felicidad o mucha tristeza).
- Monitorear la correcta ejecución de las acciones: COBOT debe estar al tanto del estado de ejecución en el sistema del robot.
- Ejecutar las acciones enviadas por el usuario: dado un guion de la obra, el usuario selecciona las acciones a ejecutar, la intensidad emocional y las envía a COBOT.
- Gestionar las acciones entrantes: Se verifica la prioridad de las acciones en ejecución, validando si pueden ser interrumpidas o no dada la llegada de nuevas acciones.

En la figura 1 se muestra una estrategia para ejecutar obras de teatro con robots a través de la teleoperación secuencial. Sin embargo, muchas veces se incluyen acciones básicas del robot como caminar, girar y decir diálogos sin dar una interpretación emocional de esas acciones. Además, cuando el actor robótico deja de recibir más instrucciones este se muestra inactivo e inmóvil o sus movimientos no son acordes con el personaje que está interpretando. Otra estrategia visible en la figura ocurre cuando se programa previamente todo el flujo de acciones que el robot va a ejecutar. Pero, esta solución impide la interacción del usuario con el sistema en tiempo de ejecución y requiere de tiempo en programar manualmente las acciones de acuerdo con las emociones del personaje.

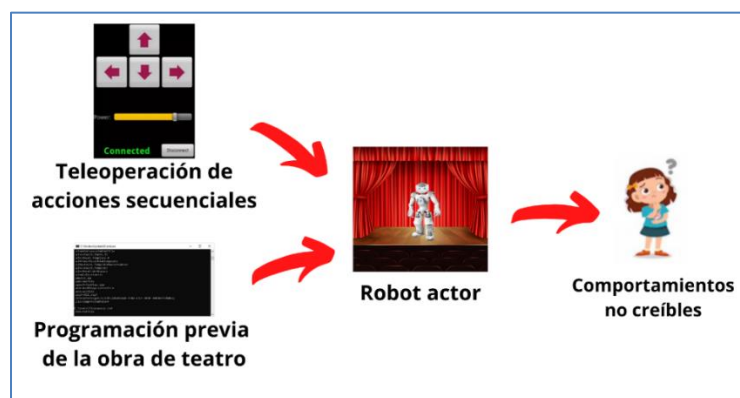


Figura 1. Proceso para realizar obras de teatro actual.

La figura 2 muestra la incorporación de COBOT el cual realiza el enriquecimiento automático de las acciones del actor agregando el componente emocional de estas, aprovechando las capacidades del robot y manteniendo al actor robótico activo durante toda la ejecución de la obra. La gestión de la concurrencia en las acciones se realiza mediante el uso de la arquitectura BDI (creencias, deseos y acciones) y de la priorización de los comandos que conforman cada acción con el propósito de ejecutar varias acciones en simultáneo.

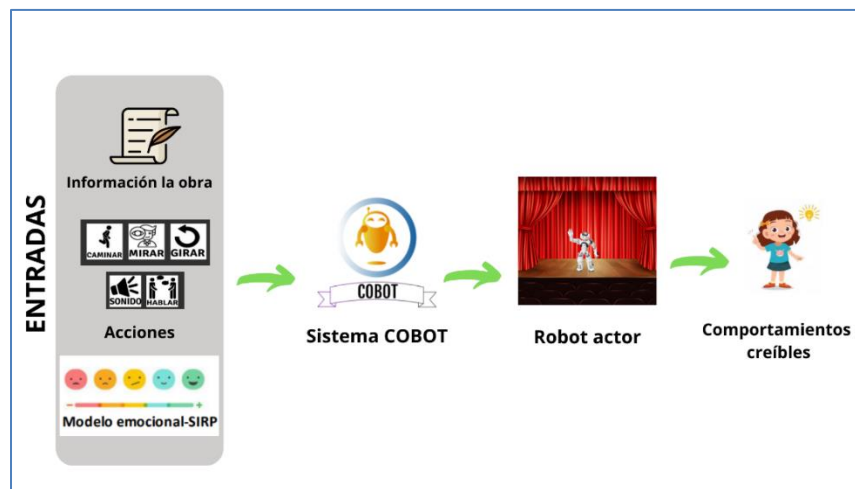


Figura 2. Proceso para realizar una obra de teatro utilizando el sistema COBOT.

1.4. Justificación de la solución

Dentro del contexto del problema se resalta la novedad y la cantidad limitada de proyectos acerca de teatro robótico, estos se ejecutan sobre un solo robot en específico o poseen un enfoque limitado en la expresión de las acciones, acotándose al envío y recepción de comandos. El sistema COBOT surge para dar flexibilidad y usabilidad en la ejecución de acciones y la expresividad que se espera que un actor tenga al interpretar un personaje en una obra de teatro.

De acuerdo con la intensidad de la emoción que el usuario defina, el sistema modula los comandos que componen cada una de las acciones para que sean ejecutados de acuerdo con dicha emoción. Así como se ejecutan acciones simples (acciones que involucran sensores y motores de un robot), COBOT consigue mantener al robot actor activo mientras el usuario no

envíe instrucciones dadas por el guion establecido al medir el tiempo de inactividad y ejecutar acciones que representen signos de vida del personaje.

Adicionalmente COBOT incluye acciones emergentes, es decir, aquellas expresiones que surgen cuando se alcanza un umbral en la intensidad de la emoción, como por ejemplo mucha felicidad. La acción emergente se plasma como comandos para el robot actor de acuerdo con sus especificaciones y se envía para que con carácter prioritario ejecute la acción por encima de los signos de vida y las acciones simples.

Todos los robots cuyo sistema operativo permita la comunicación por medio del protocolo TCP/IP [10] pueden recibir los comandos modulados por COBOT, este es el estándar utilizado para brindar flexibilidad en diversos modelos de robots. Dado lo anterior, es una solución innovadora en un campo donde hay mucho por explorar. En el ámbito de la educación se pueden realizar obras de teatro con enseñanzas educativas que de manera transversal fomentan la curiosidad por la robótica y la creatividad.

En la Pontificia Universidad Javeriana se han realizado dos proyectos: “Quyca – prueba de concepto” y “Quyca-bot: creación de guiones para dramatizaciones con robots actores para propósitos educativos” los cuales son directamente beneficiados con este proyecto debido a que se puede unificar la creación de guiones de dramatizaciones con la modulación de emociones y expresividad del actor robótico de COBOT para transmitir con propiedad la enseñanza de la obra y fomentar el uso de los robots fabricados por la Universidad Javeriana.

2. Descripción del Proyecto

En la siguiente sección se enuncia el objetivo general y los objetivos específicos para el proyecto y los entregables que lo componen, cada uno justificado con la aplicación de estándares según su enfoque.

2.1. Objetivo General

Desarrollar un sistema que implemente comportamientos creíbles en los robots actores aplicado al teatro educativo.

2.2. Objetivos Específicos

- Analizar el estado del arte sobre la gestión concurrente de acciones en robots humanoides y la expresión de emociones.
- Implementar un sistema que determine los comportamientos del robot a partir del modelo emocional SIRP y los movimientos que el robot es capaz de hacer.
- Validar la percepción de las emociones a través de una obra de teatro para niños de 7 a 11 años.

2.3. Entregables, estándares, y justificación

Entregable	Estándares asociados	Justificación
Software para teatro robótico, versión 1.0	Python Code Conventions	Las convenciones mostradas muestran las buenas prácticas que se deben seguir al momento de programar utilizando el lenguaje Python.
Informe de pruebas del prototipo	ISO/IEC/IEEE 29119	ISO/IEC/IEEE es un estándar que abarca el ciclo de vida de las pruebas de software. Se encuentra dividido en cinco partes que corresponden a las etapas de conceptos y definiciones, procesos de prueba, documentación, técnicas y pruebas por palabras clave. Se aprovecharán las herramientas y ejemplos que aporta el estándar para cualquier prueba de software.
Aplicación funcional para móvil	Android Quality Guidelines	Son estándares impuestos por Android en los cuales se enseñan tanto las buenas prácticas que se deben seguir al momento de programar usando un lenguaje Android, como las pruebas de calidad en donde se le pregunta al desarrollador cada uno de los elementos con los que debe cumplir al momento de terminado su aplicativo.
Manuales de usuario		Los manuales de usuario muestran las instrucciones y procedimientos a realizar para probar el prototipo. Se crearán: <ul style="list-style-type: none"> • Un manual para el aplicativo móvil de pruebas (ver Anexo 15: Manual de uso aplicación móvil). • Un manual para la instalación y operación del sistema COBOT (ver

		<p>Anexo 16: Manual de uso sistema COBOT.</p> <ul style="list-style-type: none"> • Un manual para la construcción e inserción del archivo de configuración de la obra dentro del sistema COBOT (ver Anexo 13: Manual de uso creación archivo de configuración de obra). • Un manual para la construcción e inserción del archivo del perfil del robot actor dentro del sistema COBOT (ver Anexo 14: Manual de uso creación archivo perfil del robot).
--	--	---

Tabla 1. Entregables y estándares del proyecto COBOT.

III- CONTEXTO DEL PROYECTO

1. Marco teórico

Con el fin de comprender los diversos conceptos que abarcan este proyecto, se realizará una definición de cada uno, comenzando por los términos más generales hacia aquellos que se derivan de estos. Para la definición de cada término, se consultaron artículos, libros y documentos relacionados extrayendo las ideas principales que conforman el concepto.

Se obtuvieron las siguientes palabras claves a partir de la revisión de la literatura y los temas importantes que componen el proyecto: Teatro, robot, teatro educativo, teatro robótico, modelado de emociones, acciones y comportamientos, acciones concurrentes, modelo BDI / BDI-CHA, comportamientos emergentes y expresión emocional. Las palabras claves se mencionan en las definiciones a continuación estableciendo la relación que tienen entre sí.

Teatro: Se refiere a la representación y recreación de aspectos de la realidad o de una historia, con el propósito de ser contemplada, estudiada o analizada [3]. Para este trabajo se acotará el enfoque del teatro a aquel en el cual los actores son robóticos y la enseñanza orientada a la educación infantil y del aprendizaje en áreas no técnicas y técnicas [6] [7] [8].

Teatro educativo: Se define como una posibilidad de implementar, mediante la metodología de la expresión, una estrategia de trabajo que vincula el teatro con la educación. Buscando así que los estudiantes desarrollen una mirada crítica hacia una propuesta artística y cultural [3]. Este proyecto se enfoca en esta rama de teatro, ya que es necesario entender a través de cuáles caminos se puede llegar al público objetivo (niños de 7 a 11 años) para transmitir un mensaje que contribuya a su proceso de aprendizaje.

Robot: Según la primera definición de la Real Academia Española, un *robot* es una “Máquina o ingenio electrónico programable, capaz de manipular objetos y realizar operaciones antes reservadas sólo a las personas” [11]. En otra definición, “el robot es un dispositivo destinado para mejorar la producción y calidad de vida” [4]. La idea clave de estas definiciones el uso de los robots para operaciones o acciones teatrales, sustituyendo al actor humano.

No obstante, se debe abarcar el motivo por el cual se decidió el uso de robots para este proyecto. Esto se debe a que los robots representan una herramienta lúdica para la enseñanza de temáticas relacionadas con ciencia, tecnología, ingeniería, arte y matemáticas en niños, lo cual es de gran ayuda al momento de apoyar su proceso de aprendizaje [5].

Teatro robótico: Se define como el uso de robots en el contexto de representaciones teatrales ya sea interactuando con humanos o entre otros robots, con el propósito de seguir los lineamientos de un guion previamente establecido [2]. Este concepto se integra con el teatro educativo para mostrar un medio que pretende promover las enseñanzas de las obras de teatro y fomentar en los niños la visión de métodos de aprendizaje innovadores y el interés por las ciencias como la robótica.

Acciones concurrentes: Las acciones concurrentes son el método para ejecutar varias acciones simultáneas. Para lograr movimiento continuo, el robot debe decidir qué nuevos movimientos realizar antes de que las acciones en ejecución terminen [12].

Actualmente existen dos aproximaciones que pueden ser usadas para exponer la arquitectura requerida en el proyecto. La primera consiste en arquitecturas reactivas que operan según los cambios del entorno. Una desventaja de estas es el poco tiempo que se tiene para decidir las

acciones que parten de las respuestas reactivas. La segunda propone realizar decisiones basadas en lógica, haciendo uso de todos los componentes del entorno y del conocimiento que tiene el robot de cómo interactuar en este. A lo largo de la literatura, se encontraron diferentes aproximaciones para gestionar acciones concurrentes, las cuales se describirán a continuación.

- **Arquitectura de control por tareas (TCA):** Este modelo provee un *framework* para el control de sistemas distribuidos de robots. Las acciones son descompuestas en tareas, las cuales son organizadas según su jerarquía. Este modelo fue diseñado para soportar comportamientos reactivos y para la coordinación de las acciones entre robots. Adicionalmente hace uso de recursos del entorno y de restricciones temporales que pueden personalizarse [12].
- **TeleoR:** Es un lenguaje de programación para agentes robóticos de mediano nivel. Este modelo utiliza reglas de conocimiento preestablecidas e información recopilada por los sensores para tomar decisiones en tiempo de ejecución. El objetivo del modelo es proveer al menos una solución a las acciones que se deben realizar, para así llegar a la completitud de su gestión. Las acciones se validan previamente para asegurarse de que no haya procesos que utilicen el mismo actuador del robot [13].
- **Control inteligente BDI:** Esta arquitectura tiene como objetivo el entendimiento y la toma de decisiones según la información representada como creencias, deseos e intenciones. Cada elemento se describe a continuación:

Creencias (*Beliefs*): Son elementos del estado mental de un agente, en este proyecto es un actor robot. Estas creencias son modificadas de acuerdo con la percepción del entorno y la comunicación con otros agentes [14].

Deseos (*Desires*): Son las alternativas de acción que tiene el agente, es decir, los objetivos que este debe lograr [14]. Según las oportunidades que tenga el agente para realizar esta acción y la acción que tenga mayor prioridad, se decidirá cuál de las alternativas será ejecutada sobre las demás.

Intenciones (*Intentions*): Son la proyección de los eventos en el futuro, es decir, es la acción por cumplir luego de ser escogida según los deseos del agente [14]. Las intenciones en este proyecto son los comandos resultantes luego de ser modulados según la emoción escogida, y la deliberación por prioridad de las acciones.

Sin embargo, con el fin de determinar cuál sería el modelo más indicado para usar en el desarrollo de este trabajo de grado, se realizó un análisis comparativo en el que se evaluaron criterios como el uso de la tele operación, así como la percepción del entorno y la entrada de acciones, que serán de vital importancia para el funcionamiento del prototipo de este proyecto. Dicho análisis se encuentra plasmado en la tabla a continuación.

Crterios / Modelo acciones concurrentes	Arquitectura de control por tareas (TCA)	TeleoR	BDI
Permite tele operación			
Si		X	X
No	X		
Percepción de entorno y entrada de acciones			
Si	X		X
No		X	
Tipo de Arquitectura basada			
Arquitectura reactiva	X		
Arquitectura lógica		X	X

Tabla 2. Tabla de análisis sobre gestión de acciones concurrentes.

Según lo observado en la tabla 2, el modelo TCA no permite la opción de uso de la tele operación. Esto, presenta un impedimento para su selección, ya que el método elegido para interactuar con el usuario final es tal criterio. Lo mismo sucede para el modelo TeleoR con el

criterio de la percepción del entorno y la entrada de acciones. Ya que, al no contar con tal característica, se convierte en un modelo incompatible con la vía deseada para desarrollar el sistema del trabajo de grado.

De acuerdo con el análisis anterior, se escoge el modelo BDI para gestionar las acciones concurrentes y describir la arquitectura del sistema en términos de los agentes del modelo. Más específicamente, se incorporará un enriquecimiento que gestiona la concurrencia de las acciones y establece una jerarquía en las metas a realizar. La arquitectura se denomina BDI-CHA.

Modelo BDI-CHA (Concurrente – Híbrido – Adaptativo): El modelo consta de una arquitectura concurrente en la línea de la investigación de los agentes híbridos hardware/software [14]. La propuesta se inspira en el pensamiento práctico humano, en el cual se delibera el comportamiento haciendo uso de los componentes del entorno, creencias e intenciones del individuo. Las siglas del modelo se componen de conceptos definidos a continuación:

El modelo es concurrente, porque añade la posibilidad de deliberar entre varios deseos, planteando las intenciones según los componentes del robot, el entorno y comunicación con otros agentes. Es híbrido, ya que puede realizarse con diferentes módulos de hardware o software, como el uso de sensores para comprender el entorno o de estructuras de datos para reconocer elementos del escenario. Y, por último, es adaptativo porque el resultado del modelo puede ser migrado y/o enviado a robots en diferentes contextos y entornos, lo que permite que el actor se exprese de acuerdo con sus posibilidades físicas [15].

Modelado de emociones: Consiste en el método para obtener el estado emocional actual del personaje. El módulo decide cómo debe afectarse este estado según eventos externos y la información de acciones previas [16]. El modelo permite gestionar la activación de comportamientos emergentes cuando se logra un umbral en la emoción modelada [15]. El modelo que será utilizado para este proyecto es el elaborado por el grupo SIRP (Sistemas Inteligentes, Robótica y Percepción) de la Pontificia Universidad Javeriana, inspirado en el modelo OCC (por

el nombre de sus autores, Ortony, Clone & Collins) [17] [18] que distingue 22 tipos de emociones diferenciadas por las situaciones psicológicas significativas que las representan. El uso de este modelo en el proyecto se define en el diseño de la solución.

2. Análisis de contexto

A partir de los conceptos mencionados anteriormente, a continuación, se van a mencionar algunos proyectos con problemáticas y soluciones similares, los cuales aportaron diferentes ideas para el diseño de la solución del presente trabajo de grado.

En el contexto del teatro robótico se han desarrollado diferentes soluciones que atacan algunos aspectos a los cuales se pretende dar solución con el sistema COBOT. Algunas soluciones son enfocadas al teatro profesional y otras encaminadas a el ámbito educativo, como lo es, por ejemplo, el trabajo titulado "teatro interactivo con robots para la enseñanza de materias no técnicas". Este, fue desarrollado por los ingenieros Flor Ángela Bravo, Alejandra María González y Enrique González, y está orientado a un público infantil de edades entre los siete y once años. Se realizaron representaciones de diferentes ideas o conceptos relacionados con asignaturas no técnicas por medio de robots actores, con el fin de poder identificar la comunicación hombre máquina y generar un razonamiento crítico acerca de los retos y beneficios que este proyecto produce [15].

Otro trabajo relacionado con teatro robótico es "*HERB's Sure Thing: a rapid drama system for rehearsing and performing live robot theater*". Es una colaboración realizada en 2014 entre el laboratorio de robótica de la universidad Carnegie Mellon y la escuela de teatro de dicha universidad, donde se realizó un sistema que permitía el desarrollo y puesta en escena de una obra de teatro en vivo protagonizada por actores robots, resaltando los desafíos que se toparon a lo largo del desarrollo del sistema y las herramientas creadas en respuesta para solucionarlos [19].

"*Autonomous Robots as Actors in Robotics Theatre - Tribute to the Centenary of R.U.R.*" es un trabajo presentado en 2019 en la Conferencia Europea de Robots Móviles (ECMR), enfocado a desarrollar un guion para que los robots Pepper y Nao desarrollasen sus funciones en la

puesta en escena. En el evento se presentaron a los robots humanoides actuando junto con actores humanos. Se buscaba a evaluar a los robots NAO y Pepper para verificar su capacidad para actuar de forma autónoma [20].

Un trabajo análogo al problema que pretende atacar este trabajo de grado es “Enfoque teatral: diseño de comportamiento humano en robots humanoides”. Este trabajo desarrolló una interfaz que permite generar comportamientos en un robot humanoide de manera natural y similar como sería en los humanos, ayudando a distintos tipos de usuarios interesados en el diseño de las mecánicas de un robot de este tipo [21]. Este trabajo tiene similitud con el presente en el sentido que se utilizan robots para obras de teatro y su objetivo es que el público perciba la obra de teatro como si un humano lo estuviera interpretando. El trabajo se diferencia en que las obras de teatro incluyen actores humanos y en que estos fueron concentrados para un robot en específico, el Robovie-R3 desarrollado por la Advanced Telecommunications Research Institute (ATR) en Japón.

De acuerdo con lo anterior, los trabajos buscan desarrollar software a partir del teatro robótico ya sea en el campo de la educación o en el espectáculo. También buscan tener una alta interactividad con el usuario, es decir, fomentando la comunicación que tiene el usuario con la máquina al momento de usarla, desde la generación o ejecución de funciones, al intercambio existente en la puesta en escena.

En los trabajos anteriormente mencionados, se trabaja con un solo tipo de robot que puede ser o no humanoide, es decir, con características morfológicas humanas. La solución de dichos proyectos es enfocada a un robot en específico, por lo que la expresividad de este se limita a su capacidad física. COBOT pretende ser un sistema flexible que se adapta a cualquier tipo de robot, mediante la modulación de los movimientos de este según una emoción y los sensores y actuadores propios de este.

Adicionalmente, en el enfoque hacia un público objetivo infantil existen pocas referencias en teatro robótico, se destacan los avances realizados en la Pontificia Universidad Javeriana por el grupo de investigación SIRP – Sistemas inteligentes, Robótica y Percepción en la línea de

investigación de robótica educativa [22]. Este grupo planteó e implementó el desarrollo tecnológico Quemes, el cual se enfoca actualmente en la sensibilización y adquisición de conocimiento de estudiantes.

IV- ANÁLISIS DEL PROBLEMA

En la presente sección se describirán los requerimientos del sistema, sus restricciones y las especificaciones funcionales que posee. Esta información surge del análisis del estado del arte, explicado en la unidad anterior y de los apartados 1 y 2 del capítulo III.

1. Requerimientos

Para proponer los requerimientos de COBOT se realizaron reuniones con la directora del trabajo de grado, ya que al ser la *product owner*, guio el levantamiento de los requerimientos del sistema. Primero se obtuvo una lista de historias de usuario, las cuales fueron sometidas a un proceso de priorización ([ver Anexo 11: Historias de usuario y planning póker](#)). Los resultados presentaron las funcionalidades que posteriormente serían transformadas en requisitos funcionales, como se puede apreciar en la tabla 2. ([ver Anexo 8: documentación requerimientos funcionales](#)) ([ver Anexo 9: documentación requerimientos no funcionales](#)).

ID	Descripción del requerimiento
RF-1	El sistema debe proveer el medio de comunicación para el intercambio de información con el usuario.
RF-2	El sistema debe permitir la carga de la configuración de la obra de teatro del usuario.
RF-3	El sistema debe permitir la carga de acciones simples provenientes del usuario.
RF-4	El sistema debe permitir la carga de acciones emergentes provenientes del usuario.
RF-5	El sistema debe disparar la ejecución de signos de vida para el personaje en momentos de inactividad.
RF-6	El sistema debe leer la información del perfil del robot seleccionado.
RF-7	El sistema debe crear el modelo del mundo de la escena seleccionada.
RF-8	El sistema debe producir el enriquecimiento de las acciones a partir de los comandos que la componen, localizados en el perfil del robot.

RF-9	El sistema debe calcular los valores de los parámetros que conforman cada comando a partir de su rango de valores y la intensidad de la emoción.
RF-10	El sistema debe calcular el recorrido que debe realizar el robot en acciones que implican desplazamiento.
RF-11	El sistema debe recopilar los comandos que conforman una acción emergente localizados en el perfil del robot.
RF-12	El sistema debe modular los comandos que conforman los signos de vida, localizados en el perfil del robot.
RF-13	El sistema debe enviar los comandos resultantes al robot para su ejecución.
RF-14	El sistema debe gestionar la prioridad de los mensajes entrantes del usuario.
RF-15	El sistema debe recibir el estado de ejecución del robot.
RF-16	El sistema debe retornar el estado de ejecución de los mensajes entrantes al usuario.

Tabla 3. *Requerimientos funcionales del sistema COBOT.*

2. Restricciones

Así como se realizó un levantamiento de requerimientos, fue necesario realizar lo mismo con las restricciones, debido a que estas presentan las limitaciones a nivel de diseño, desarrollo y pruebas del sistema. Para que el sistema funcione completa y correctamente, se recopilaron las siguientes restricciones según su tipo:

Tipo de restricción	Descripción de la restricción
Hardware	<ul style="list-style-type: none"> • Computador de 8GB de RAM, procesador Core i3 sexta generación como mínimo para el desarrollo del sistema. En esta restricción influye los requerimientos de las herramientas IDE utilizadas para el desarrollo. • El computador debe contar con una tarjeta de red que le permita conectarse a una red local.
Software	<ul style="list-style-type: none"> • Sistema operativo Windows 7 o superior. Sin embargo, para evitar tener que aplicar permisos de envío y recepción de sockets en el <i>Firewall</i> se recomienda utilizar la versión Windows 10 Pro. • Python 2.7.18 para la programación del sistema, debe estar agregado al Path principal en las variables de entorno del sistema. • Instalación de las API, entornos de simulación del robot tales como Choregraphe 2.8.6.23. La versión del programa es la última accesible por el proveedor Softbank Robotics, para las pruebas con el robot NAO. • Para añadir las librerías de la API naoqi, debe agregarse la dirección en el computador a las librerías en la variable <i>PYTHONPATH</i> de las variables de entorno del sistema.

Usuario	<ul style="list-style-type: none">• La operación del sistema COBOT debe ser realizada por usuarios con conocimiento en el lenguaje de programación Python y el manejo de archivos tipo JSON.• La operación del sistema de pruebas puede ser realizada por usuarios con poco o nulo conocimiento en programación, tales como niños.
Herramientas	<ul style="list-style-type: none">• El protocolo de comunicación del sistema COBOT con el aplicativo móvil de pruebas debe ser mediante el protocolo TCP/IP.• El protocolo de comunicación del sistema COBOT con el robot actor debe ser mediante el protocolo TCP/IP.• La información de entrada al sistema (configuración de obra, acciones) debe estar agrupada en formato JSON y siguiendo con las indicaciones del manual de elaboración.• La información del perfil de cada robot debe ser escrita en formato JSON y siguiendo con las indicaciones del manual de elaboración.

Tabla 4. Restricciones del sistema COBOT.

3. Especificación Funcional

En esta sección, se describirán las especificaciones funcionales del sistema a alto nivel, haciendo uso de diagramas de casos de uso o de procesos, para comprender la forma como se distribuyen los requerimientos de diseño y la interacción entre los actores con las funcionalidades.

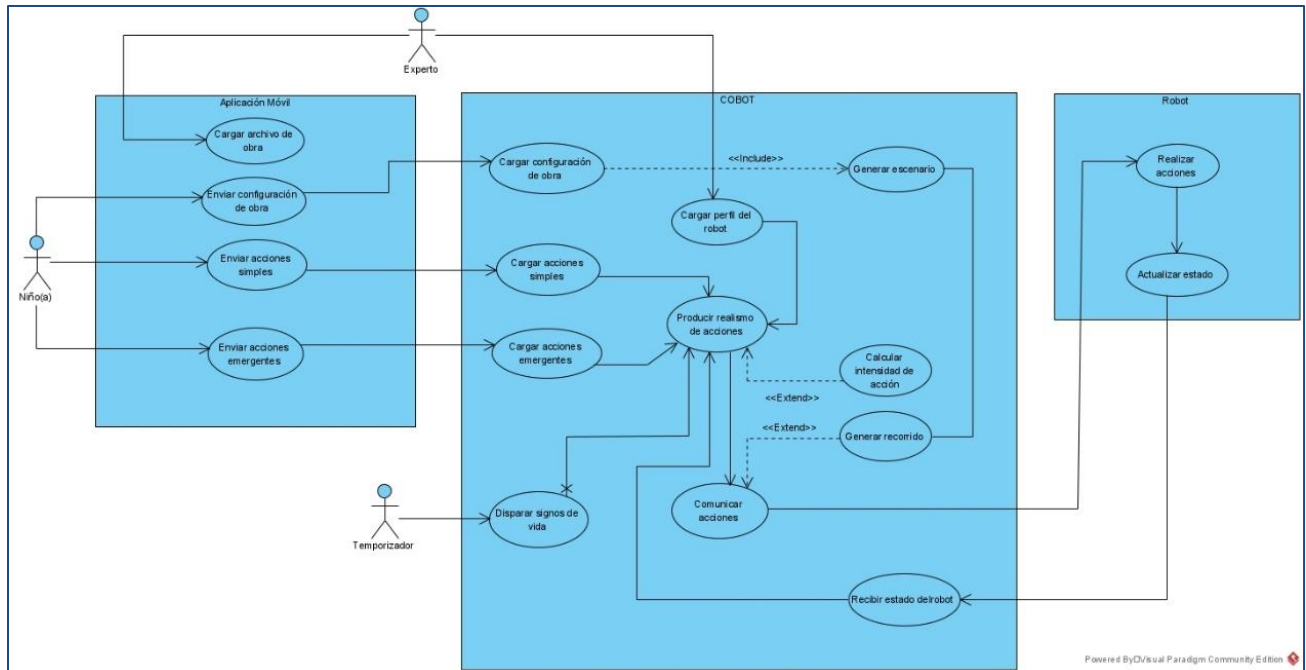


Figura 3. Diagrama de casos de uso para el proyecto COBOT.

Para revisar a detalle las funcionalidades presentes en el diagrama de casos de uso según el actor que las realiza ([ver Anexo 7: Documentación casos de uso](#)).

V- DISEÑO DE LA SOLUCIÓN

En la siguiente sección se describe el diseño realizado para el sistema. Primero, se expondrá el funcionamiento general, explicando cada uno de los agentes que lo conforman. También se muestran las pruebas de diseño realizadas con el público objetivo para establecer la interfaz del sistema. Luego, se presentan las vistas lógica y física del sistema diseñado.

1. Funcionamiento general del sistema

En esta sección se describe el funcionamiento general del sistema. La figura 4, muestra la representación de la arquitectura de COBOT, el aplicativo de pruebas y el sistema del robot actor, en términos de los agentes que interactúan entre sí.

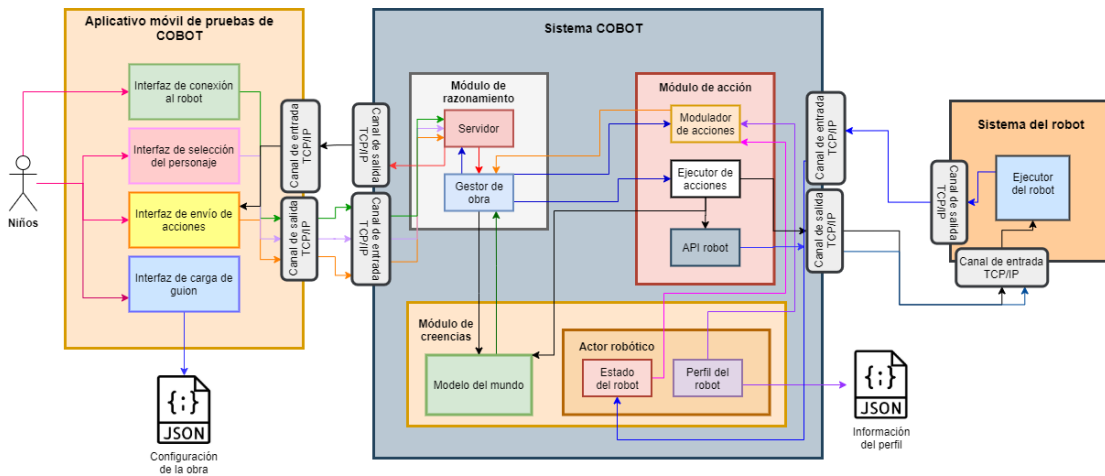


Figura 4. Diagrama arquitectural de COBOT.

1.1. Sistema COBOT:

El sistema COBOT contiene la implementación de los agentes que permiten la modulación emocional de las acciones, la gestión de la concurrencia y la comunicación tanto del proveedor de la información de entrada como del robot actor y la gestión de su estado. A continuación, se presentan los elementos del sistema:

Agente de razonamiento:

Este agente es el encargado de gestionar la conexión y transmisión de la información de la obra de teatro y los robots actores (ver figura 5). Para lograr esta función, se diseñaron dos elementos que permiten la comunicación con los sistemas proveedores de los datos de entrada, su persistencia y el control de las peticiones del usuario al seguir el guion de una obra de teatro.

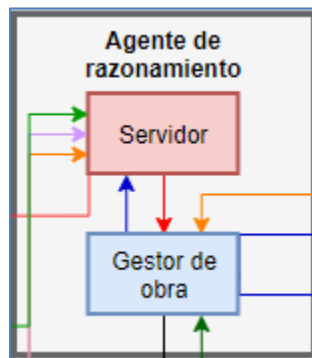


Figura 5. Sección agente de razonamiento diagrama COBOT.

- **Servidor:** gestiona la información de cada uno de los usuarios que se conectan al sistema. Este bloque es responsable de informar al usuario sobre el resultado de la ejecución de las peticiones realizadas. La información que maneja el servidor se describe a continuación:
 - Información de conexión con los usuarios: El servidor debe mantener los datos de los usuarios que se conecten al sistema, según el protocolo de comunicación.
 - Información de las peticiones realizadas: El servidor gestiona la información por robot. Cada robot tiene sus datos almacenados en su gestor de obra respectivo.
 - Tiempo de inactividad de los robots: El servidor centraliza los procesos concurrentes del sistema, manejando los eventos que se disparan y las peticiones del usuario. Para ello debe realizar procesos en paralelo para atender cada usuario al momento de ser requerido.
- **Gestor de obra:** gestiona la información específica de un robot actor y el estado en el cual este se encuentra. El gestor de obra almacena los datos de las acciones entrantes según su categoría:
 - Acciones simples: Son aquellas acciones que están escritas en el guion de la obra de teatro. Estas acciones varían el modo de expresión según la emoción que presente el robot durante el fragmento de la escena.
 - Acciones emergentes: Son comportamientos que surgen cuando se alcanza un umbral emocional. Este tipo de acciones se diseñaron con el fin de transmitir a la audiencia que el robot tiene un estado emocional a un nivel elevado, y así apoyar la validación de la percepción de tal emoción.
 - Signos de vida: Son comportamientos que expresa el robot en momentos de inactividad durante la obra de teatro. Para evitar momentos donde el robot está completamente detenido ya que no recibe acciones para ejecutar, fueron diseñadas acciones que permiten que el robot permanezca activo. Se diseñaron con el fin de que la audiencia no pierda la atención con el robot, y que a su vez este exprese como se siente durante la obra de teatro además de lo que se establece en el

guion. Además, estas acciones dependerán del personaje de la obra que el robot esté interpretando, por lo que no todos los personajes tendrán los mismos signos de vida (p. ej. un robot que interprete un árbol moverá sus hojas y un robot que interprete un perro moverá su cola).

Además, el gestor de obra conoce el momento en el que el robot se encuentra durante toda la ejecución de la obra, por lo que puede gestionar la ejecución de acciones cuando reciba una con mayor prioridad. Para la validación de la prioridad, fue diseñada la pirámide de objetivos (que se puede apreciar en la figura 6) basada en la pirámide propuesta por el modelo BDI-CHA.



Figura 6. Pirámide de prioridad de acciones.

Agente de creencias:

Este agente se encarga de brindar al sistema la información del entorno e información sobre la forma como cada robot actor interpreta las acciones de la obra de teatro.

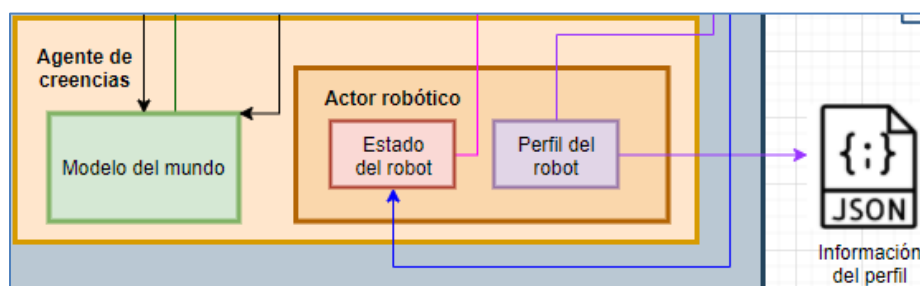


Figura 7. Sección agente de creencias diagrama COBOT.

- **Modelo del mundo:** Es el bloque que almacena la información del lugar donde se ubican los robots en el escenario de la obra. Para todos los robots que participen en la misma

escena este modelo del mundo es compartido. Sin embargo, el punto de vista y la ubicación de cada robot es diferente.

- **Actor robótico:** El actor robótico contiene la información necesaria para que el sistema module las acciones del robot y gestionar el estado resultante de la ejecución de estas. Sus bloques internos son:

- **Estado del robot:** El estado del robot es el bloque que permite validar la finalización de la ejecución de las acciones, su interrupción o la ubicación del robot en el modelo del mundo. Este bloque es utilizado al momento de ejecutar las acciones para asegurarse de que se realicen correctamente para luego informar tal estado al usuario.
- **Perfil del robot:** El perfil incluye toda la información para el mapeo de las acciones del guion de la obra en términos de los componentes de hardware y software del robot. Esta información del mapeo es lo que permite la modulación de acciones según la emoción, ya que este se divide según el eje emocional (dos emociones opuestas tales como felicidad y tristeza componen un eje emocional). Es uno de los bloques más complejos en elaboración debido a que se deben construir las acciones adaptadas al hardware y software del robot propietario del perfil, y establecer el valor de los parámetros de cada movimiento para que logre expresar la emoción requerida.

Para elaborar el perfil, este bloque lee la información del mapeo desde un archivo cuya estructura permite organizar la información a modo de objetos con atributos. En la figura se observa la presencia de tal archivo con la información del perfil. El propósito de diseñar este bloque es para permitir que se puedan modular las acciones según como se exprese un robot dados sus componentes.

Agente de acción:

El agente de acción tiene como meta realizar la modulación de las acciones entrantes de acuerdo con la información del robot y posteriormente realizar su ejecución y gestión del estado de este. La figura 8 muestra el contenido del agente:

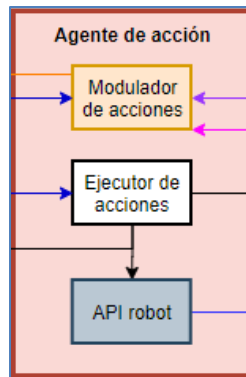


Figura 8. Sección agente de acción diagrama COBOT.

- Modulador de acciones:** La modulación de acciones consiste en ajustar las acciones ingresadas según el mapeo de estas, distribuido en comandos. Se diseñaron diferentes tipos de comandos que responden a lo que se pretende ejecutar, estos comandos se definen en la siguiente tabla:

Tipo de comando	Descripción
Comando de mapeo general	Estos comandos son aquellos únicos y esenciales para que se ejecute la acción (p. ej. mover los motores de las piernas para caminar). Dentro del perfil del robot se tiene el rango de valores de los parámetros que componen el comando, es decir, según las especificaciones del robot este ejecutará las acciones de diferente forma y velocidad según este rango. Para este tipo de comandos, el rango de valores que se tiene para cada uno depende del eje emocional y la emoción seleccionada.
Comando de mapeo secundario	Estos comandos definen el enriquecimiento de las acciones que se seleccionan. Este enriquecimiento es propio de cada robot según sus especificaciones, como por ejemplo el uso de leds en el cuerpo o un movimiento de brazos específico. Este mapeo se modula de la misma forma que el mapeo general, pero se ejecuta independientemente ya que varias acciones tienen comandos y parámetros de enriquecimiento similares. Por tanto, se referencia al mismo objeto del archivo del perfil evitando que se repita información. Para este tipo de comandos el rango de valores que se tiene para cada uno depende del eje emocional y la emoción seleccionada.
Comando sin modulación	Estos comandos no requieren modulación debido a que sus posibles valores no pueden aplicarse como variables en una ecuación matemática o representan un archivo externo que se debe reproducir o cargar. Estos comandos se almacenan en el perfil como un objeto in-

	dependiente que es referenciado en el mapeo general. La modulación entonces consiste en leer el archivo y convertir el contenido en un objeto en memoria. Estos comandos a diferencia de los dos anteriores no dependen del eje emocional ni de la emoción seleccionada, ya que es una opción del usuario que debe cumplirse de cualquier manera con cualquier estado emocional.
Comando de opción seleccionada en el aplicativo de pruebas	Este comando hace parte de los comandos sin modulación, sin embargo, se clasifica aparte debido a que toma su valor de acuerdo con la opción seleccionada en el aplicativo móvil. En el aplicativo se les permite a los usuarios escoger una opción según la acción, por ejemplo, el destino al cual quiere que el robot llegue caminando o el diálogo que quiere que el robot reproduzca. Este comando se busca en el perfil del robot para traducir la opción seleccionada a un valor que sea entendible por el sistema (p. ej. ir hacia la fachada de la casa significa desplazarse hacia el nodo número 2).

Tabla 5. Tipos de comandos para la modulación de acciones.

El modulador de acciones aplica a los comandos provenientes del mapeo general y secundario la ecuación de modulación de acciones propuesta para este proyecto. Dicha ecuación parte del uso del resultado de aplicar el modelo emocional del grupo *SIRP* de la Pontificia Universidad Javeriana, con el cual se simula el resultado que da cuando se estima la emoción del robot. Este modelo da como resultado un coeficiente que permite modular los parámetros de cada comando, este va desde -1 hasta 1.

El valor del modelo emocional es enviado al servidor como dato de entrada cuando recibe la información de la emoción y las acciones a ejecutar. Gracias a este coeficiente se puede calcular la intensidad que deben tener los parámetros de los comandos modulables. El cálculo de este valor se realiza aplicando la siguiente ecuación:

$$V = \left(\left(\frac{I_e + 1}{2} \right) * (MAXR_{xe} - MINR_{xe}) \right) + MINR_{xe} \quad (1)$$

Figura 9. Ecuación que representa la intensidad de los parámetros de los comandos modulables.

La variable I_e representa la intensidad emocional en la escala del modelo *SIRP* proveniente del usuario, y los valores $MAXR_{xe}$ y $MINR_{xe}$ representan los valores máximo y mínimo que el parámetro permite ejecutar. El parámetro y su rango provienen del perfil del robot escogido y dependen de la emoción seleccionada. El resultado de aplicar la ecuación es el valor del parámetro en función de la intensidad emocional.

Una vez moduladas todas las acciones, luego se debe validar que no existan dos o más parámetros modulados que utilicen el mismo recurso de hardware o software del robot. Esto se realiza con el fin de evitar problemas de concurrencia cuando se ejecuten las acciones. Para ello, a cada parámetro se le establecerá una prioridad global, la cual servirá de punto de comparación para escoger qué parámetro permitir, así solo uno hará uso del componente requerido del robot.

- **Ejecutor de acciones:** Es el bloque encargado de conectarse al robot y enviar los comandos modulados para que sean ejecutados en su sistema. El bloque es llamado por el gestor de cada robot, el cual le proporcionará tales comandos y la información del robot para establecer la conexión. Además, este bloque se encarga de verificar la necesidad de desplazamiento del robot a través del modelo del mundo, es decir, se encarga de solicitar la generación de la trayectoria que debe seguir el robot. La ejecución de las acciones fue diseñada de dos maneras:

Cuando el robot cuenta con un API para ejecutar las acciones: Para que el robot seleccionado en la propuesta de este trabajo (robot NAO) ejecute los comandos modulados, este hace uso del API que permite organizar los comandos de forma comprensible para su sistema operativo. Se elaboró un bloque denominado **API robot** ([ver figura 8](#)) que utiliza las herramientas de la interfaz y envía los comandos al robot directamente para su ejecución, por lo que la labor de organización de los parámetros es realizada desde el sistema COBOT y no desde el sistema operativo del robot.

Cuando el robot no cuenta con un API para ejecutar las acciones: Este método de ejecución de acciones consiste en enviar el conjunto de comandos modulados al robot, y que este tenga en su sistema operativo un componente implementado que reciba tales comandos y los organice para su ejecución según el modelo del robot y el lenguaje de su sistema. El método no está contemplado dentro del alcance de este proyecto, por lo que supone un valor agregado al resultado de la ejecución de acciones, permitiendo que los comandos sean enviados a cualquier modelo de robot cuyo sistema operativo permita recibir los comandos desde el protocolo de integración. Es importante destacar que cada

vez que se desee agregar un nuevo robot que contenga un API, este deberá ser programado para cada robot.

Al final de la ejecución de las acciones se debe recibir el estado de ejecución de estas para informar al usuario e iniciar el momento de inactividad.

1.2. Sistema del robot:

El sistema del robot es robot encargado de ejecutar las acciones y realizar los movimientos o expresiones resultantes de la modulación y el envío de los comandos. Dentro de este sistema, según el tipo de robot y la presencia de API, tiene un bloque encargado de recibir los comandos y traducirlos al lenguaje del sistema operativo. En cualquier caso, este sistema se comunica con COBOT para indicarle el resultado de ejecución de las acciones y su nueva ubicación en el modelo del mundo para ser actualizada.

1.3. Aplicativo móvil de pruebas de COBOT:

Se diseñó un aplicativo móvil para las pruebas con los usuarios finales, ya que se requiere de una herramienta con la que ellos puedan interactuar y controlar al robot según las acciones establecidas en el guion. El sistema permite enviar la información de entrada que COBOT requiere para su funcionamiento y conexión con los robots. En el alcance de este proyecto no está contemplado el uso de una interfaz para el seguimiento del guion de la obra, por lo que este componente es un valor agregado de gran importancia para las validaciones finales, ya que permite realizar las verificaciones del sistema con niños de una forma interactiva, y porque anteriormente se propuso ingresar las acciones y la emoción desde la consola del sistema.

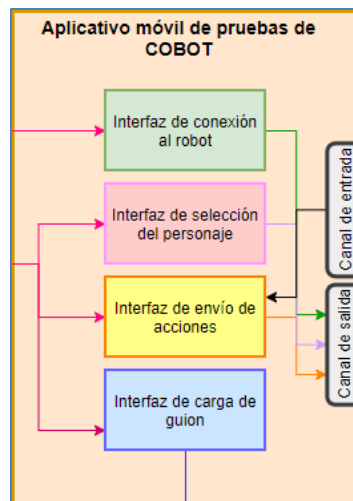


Figura 10. Sección aplicativo móvil diagrama COBOT.

Las interfaces diseñadas para el aplicativo de pruebas son:

- **Interfaz de conexión al robot:** La interfaz solicita al usuario la información de conexión de red para el robot, con el fin de que el servidor pueda establecer el canal de comunicación con él y enviar los comandos a ejecutar.
- **Interfaz de carga de guion:** Esta interfaz permite que el usuario cargue un archivo correspondiente a la información de la obra. Este archivo debe ser elaborado por un experto, el cual plasma el guion de la obra en el archivo y lo sube al almacenamiento del dispositivo.
- **Interfaz de selección de personaje:** La pantalla permite escoger el personaje de la obra de teatro cargada.
- **Interfaz de envío de acciones:** La interfaz central que reúne los datos de entrada que requiere el sistema COBOT para ejecutar las acciones del guion. A continuación, se describe el diseño de esta interfaz del aplicativo.

Se realizaron diseños de prototipos para este sistema, los cuales fueron evaluados por personas con edades dentro del rango del público objetivo, con el fin de obtener su retroalimentación respecto a cómo desean ver el sistema cuando se realicen las validaciones y lo que se espera que puedan realizar ([ver Anexo 20: Prototipos de papel versión 1.0](#)) ([ver Anexo 21: Prototipos de papel versión 2.0](#)) ([ver Anexo 23: Resultados pruebas aplicación móvil](#)).



Figura 11. Prototipo interfaz de envío de acciones.

La figura 11 muestra las primeras aproximaciones que se diseñaron para el aplicativo de pruebas, en este caso mostrando la interfaz de usuario para el envío de acciones. Esta interfaz permite realizar lo siguiente:

- Seleccionar las acciones que puede hacer el robot en la escena.
- Seleccionar la emoción y la intensidad de esta.
- Cambiar de escena en tiempo de ejecución.
- Visualizar información del personaje seleccionado.

En el prototipo final, expuesto en la fase de implementación, se removió el componente de estado del personaje o 'mood' y fue reemplazado por las acciones emergentes que se manifiestan cuando se selecciona una emoción y un nivel elevado de esta. Los resultados de las validaciones del prototipo con el usuario final se pueden ver a detalle en el anexo 23.

2. Vista lógica del sistema

En esta sección se explica la vista lógica del sistema COBOT ([ver Anexo 6: Diagramas de diseño proyecto COBOT](#)), el cual, relaciona los sistemas mediante el modelo Cliente / Servidor operando con una estructura de tres capas sin sistemas intermedios o Middleware, ya que no se maneja información personal sobre el usuario y no se requiere de un sistema de seguridad intermedio para el control de las peticiones al servidor.

La estructura se divide de la siguiente manera:

- **Capa de lógica de presentación:** Esta capa contiene la implementación del aplicativo móvil del usuario, el cual funciona como el cliente que envía peticiones al servidor. Para este proyecto se cuenta con un aplicativo de pruebas escrito en el lenguaje Android [23]. Esta aplicación se aloja en un dispositivo móvil.
- **Capa de lógica de negocio COBOT:** Esta capa contiene la implementación del sistema COBOT, sus módulos del modelo BDI-CHA se encuentran alojados en los componentes de Negocio COBOT e integración COBOT. Esta capa hace referencia al servidor que recibe las peticiones de los usuarios y almacena la información en memoria durante la ejecución de este. El servidor no conoce las direcciones de los clientes, pero los clientes deben conocer la dirección IP y puerto del servidor para poder conectarse [23].
- **Capa de lógica de negocio ROBOT:** Esta capa hace referencia al robot que ejecuta los comandos modulados. En la estructura de capas fue agregado el sistema del robot porque a este puede agregársele el código que desempaquete los comandos en caso de no tener un API en el sistema del servidor.

En la siguiente figura, se muestran los componentes que conforman el sistema y sus conexiones entre sí.

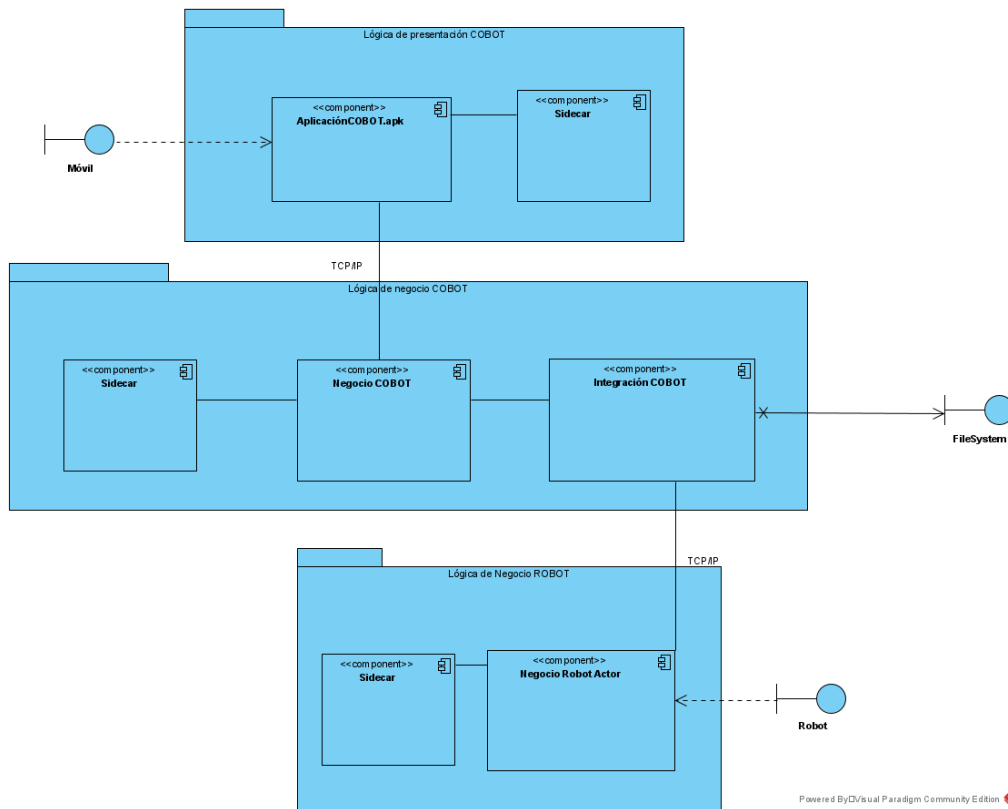


Figura 12. Vista lógica del sistema para el proyecto COBOT.

A continuación, se muestra una tabla con la definición de los componentes de la vista lógica:

Nombre del componente	Descripción
Aplicación COBOT	Aplicación de pruebas que es operada por el experto y los niños. Esta aplicación es el cliente que envía al servidor los datos de la obra de teatro y las acciones a ejecutar.
Negocio COBOT	Contiene la lógica de negocio del sistema, la cual se encarga de gestionar los mensajes del cliente y modular las acciones entrantes del mismo. Así mismo, el servidor se encarga de gestionar los procesos concurrentes que se disparen a causa de la configuración enviada. Este componente almacena los agentes vistos en las figuras de la vista general, los cuales son: <i>gestor de obra</i> , <i>action modulator</i>
Integración COBOT	Este componente se encarga de enviar la información de los comandos modulados hacia el robot, este opera de dos formas:

	<ul style="list-style-type: none"> • Cuando la API del robot está en el lenguaje del servidor, se hace uso de esta para el mapeo de las acciones a términos comprendidos por el robot. • Cuando el robot no cuenta con una API, se envían los comandos por medio del protocolo TCP/IP, para que el robot las ejecute. <p>En este componente se almacenan los agentes <i>action executor</i>, <i>servidor</i> y el agente ejecutor del API del robot NAO, ya que cada uno envía mensajes a sistemas externos.</p>
Sidecar	El componente transversal aloja las entidades que son necesarias para la carga y persistencia de la información alrededor de toda la arquitectura del sistema. En la aplicación móvil y el robot también están presentes algunos componentes del sidecar, ya que contiene la estructura de los comandos, y de la información de envío.
Negocio Robot Actor	Es el sistema instalado en el robot, el cual se encarga de ejecutar los comandos enviados. Para el caso de robots que no cuentan con una API, debe existir este componente para desempaquetar los comandos y gestionarlos con respecto a los componentes del robot. En el caso de contar con una API, este componente ya recibe los comandos en términos de su sistema, por lo que se encarga de ejecutarlos.

Tabla 6. Definición componentes para la vista lógica del sistema en el proyecto COBOT.

3. Vista física del sistema

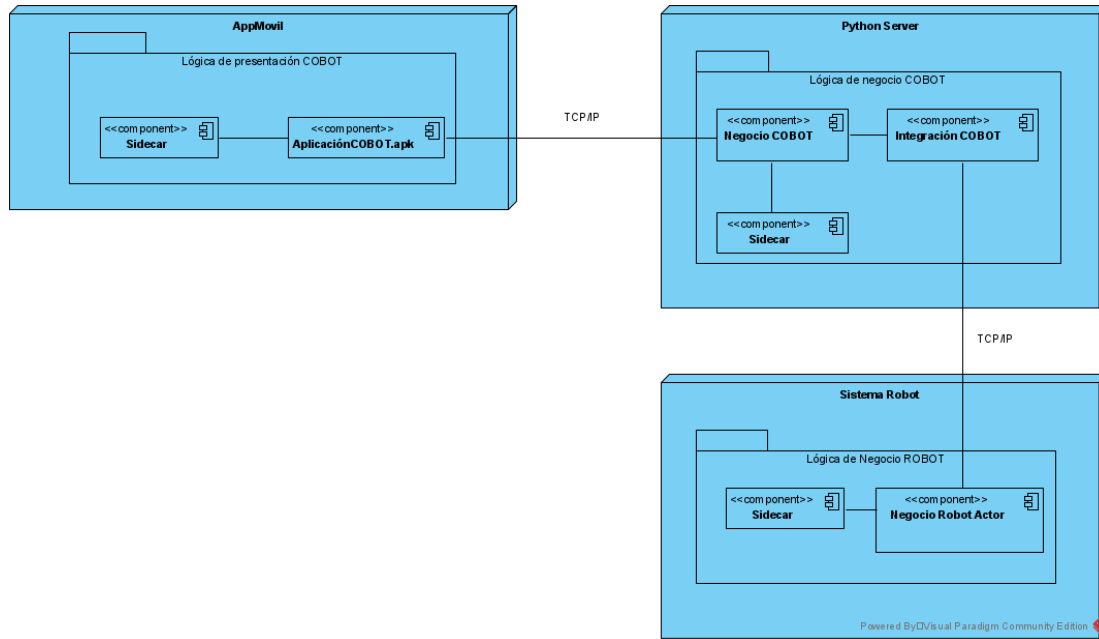


Figura 13. Vista física del sistema para el proyecto COBOT.

A continuación, se muestra una tabla que explica el contenido de la figura, con sus respectivas conexiones:

Nombre del nodo	Características
App Móvil	<p>Software</p> <ul style="list-style-type: none"> - Aplicación móvil para dispositivos con sistema operativo Android 6.0 mínimo. <p>Hardware</p> <ul style="list-style-type: none"> - 500 MB RAM, 60 MB de almacenamiento interno. - Conexión a internet con Wi-Fi 802.11ac/3G/4G.
Python Server	<p>Software</p> <ul style="list-style-type: none"> - Programa de consola para la visualización del estado del programa, recepción y envío de peticiones a clientes. - Python 2.7.18. - Para el uso del API del robot NAO: Instalación de pynaoqi-python2.7-2.8.x.x-win32-vs2013. <p>El servidor es programado en el lenguaje de programación Python, debido a que el robot utilizado para las pruebas posee un API en este lenguaje, para más información sobre la herramienta utilizada para la implementación ver la <i>tabla 8</i>).</p>
Sistema Robot	<p>Software para otros robots</p> <ul style="list-style-type: none"> - Independiente de cada robot.

	<ul style="list-style-type: none"> - Conexión a internet con Wifi IEEE 802.2. <p>Software utilizado para el proyecto:</p> <ul style="list-style-type: none"> - Conexión a internet con Wifi IEEE 802.2. - Robot NAO V5 H20 utilizado en simulador Choregraphe 2.8.6.23
--	--

Tabla 7. Definición componentes para la vista física del sistema en el proyecto COBOT.

4. Herramientas de desarrollo

En esta sección se enuncian las herramientas utilizadas para el desarrollo del sistema, los lenguajes de programación escogidos y los estándares o protocolos de comunicación escogidos para la integración del sistema COBOT, los robots y el aplicativo de pruebas. La siguiente tabla lista y define las herramientas de trabajo utilizadas, así mismo la razón por la cual algunas de estas cambiaron con respecto a las escogidas en el documento PMP del presente proyecto por causas sociales o restricciones encontradas.

Nombre de la herramienta	Tipo de herramienta	Razón de su uso
Python	Lenguaje de programación	Se decidió utilizar este lenguaje, ya que, para las pruebas de ejecución de comandos con el robot de la NAO de la Pontificia Universidad Javeriana, sólo está disponible actualmente la API de Python. La versión del intérprete debe ser 2.7.18. Es importante aclarar que originalmente se pretendía utilizar el lenguaje de programación Java en NetBeans IDE, pero por causa de la restricción de la API se cambió a Python, en el cual uno de los miembros del grupo tiene conocimiento. La herramienta para programar en este lenguaje será Pycharm, que hace parte de los creadores del entorno de Android Studio conocido por dos de los miembros del grupo.
Android Studio	Desarrollo de Software	Es una herramienta de desarrollo de aplicaciones para dispositivos móviles, utilizada para crear la aplicación de pruebas. Esta aplicación se realizó a modo de entregable extra, para poder probar el funcionamiento del sistema y la interacción que pueden tener los niños al utilizarlo.
Choregraphe	Desarrollo / Simulación	Es una herramienta propia de la empresa Softbank Robotics desarrolladores del robot NAO

		utilizado para las pruebas del sistema. El programa ofrece un simulador del robot con casi todas las características del robot real, y adicionalmente ofrece un entorno de desarrollo para programar las acciones que ejecuta el robot de forma intuitiva con el uso de cajas que se ubican en un plano.
Github	Manejo de versiones, control del código fuente	Es un repositorio para almacenar los archivos de código y pruebas del prototipo, donde se crean ramas de trabajo para dividir de manera eficiente el desarrollo en sus diferentes etapas según los hitos a alcanzar durante los sprints. La herramienta fue utilizada para mantener al sistema y a la aplicación de pruebas en el mismo repositorio.
Visual Paradigm 16.2	Diseño, construcción de diagramas	Es un software de diseño de arquitectura de software que posee todas las herramientas necesarias en su edición comunitaria para elaborar diagramas de vista lógica, física, de procesos y otros que requieran elementos gráficos especiales para representarlos. Se utilizó este programa en reemplazo de Bizagi Modeler, debido a que se requería la elaboración de diagramas de arquitectura más que de procesos, por lo que se trasladó su elaboración a este programa, además, todos los integrantes tienen conocimiento en él (ver Anexo 6: Diagramas de diseño proyecto COBOT).
Draw.io	Diseño, construcción de diagramas	El programa es una aplicación de Google Online para diseñar y crear diagramas que pueden utilizar estándares UML para su elaboración. Se utilizó la aplicación para elaborar los diagramas del funcionamiento general del sistema, así como otros diagramas para visualizar la arquitectura durante las reuniones con los miembros y el <i>product owner</i> .
Microsoft Teams	Documentación y administración del proyecto	La plataforma para trabajo compartido de Microsoft 365, permite gestionar las reuniones semanales dadas por la metodología escogida, las reuniones con el <i>product owner</i> . Adicionalmente funciona como repositorio para la documentación, archivos de pruebas y manejo del cronograma de actividades realizado (ver Anexo 12: Informe de reuniones), ya que permite subir

		cualquier tipo de documento y visualizarlo o descargarlo posteriormente.
--	--	--

Tabla 8. Herramientas de trabajo utilizadas para el desarrollo del proyecto COBOT.

VI- DESARROLLO DE LA SOLUCIÓN

Este capítulo describe la forma como fue implementado el sistema COBOT, haciendo uso de la metodología planteada para esta fase del proyecto. El contenido se apoya por capturas de la consola del servidor y las pantallas con las que el usuario final interactúa o visualiza en tiempo de ejecución.

1. Metodología

La metodología escogida fue Scrum, esta fue comparada con otras metodologías para proyectos de desarrollo de software en el documento *Project Management Plan* y obtuvo la mayor puntuación según los criterios establecidos para su escogencia. Gracias a esta metodología se monitoreó el avance del proyecto en cada *sprint* y se detectaron riesgos y nuevas restricciones que afecten el desarrollo y el estado final esperado.

Los *sprints* tuvieron una duración de 8 a 15 días, cuyos objetivos consisten en implementar secciones completas del sistema y avances en los documentos adicionales y diagramas de diseño.

1.1. Estructura de ramas

Cada etapa de la implementación fue realizada en ramas utilizando la plataforma *GitHub*, la cual permite controlar el versionamiento del código fuente y organizarlo según las necesidades del grupo. La siguiente figura muestra la distribución y orden en el que se crearon las ramas para el desarrollo del servidor:

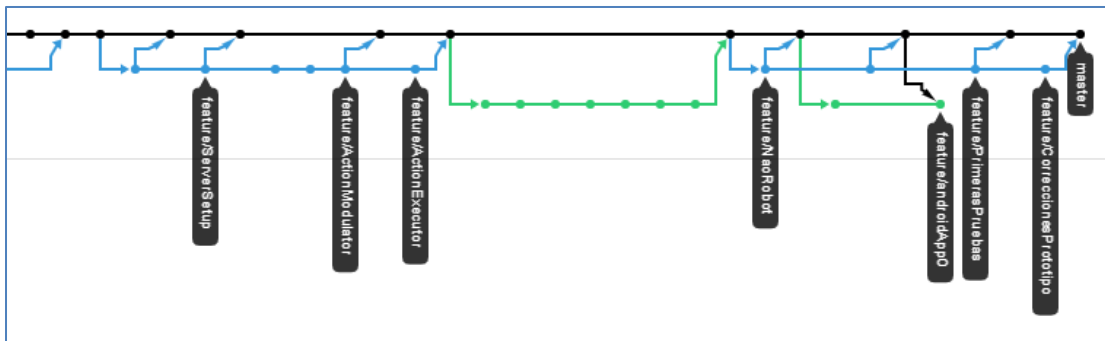


Figura 14. Distribución y orden de creación de ramas para desarrollo del servidor.

Esta figura muestra una parte de la implementación, cuyo orden va de acuerdo con el cronograma propuesto para el semestre.

2. Implementación

La implementación del sistema completo se compone del aplicativo móvil de pruebas y el servidor en un computador. La intención de desarrollar el aplicativo es con el fin de que se pueda interactuar con el sistema de forma intuitiva y sencilla al momento de realizar las pruebas. Este componente es adicional a lo propuesto para la solución, pero es beneficioso para la validación de los resultados.

A continuación, se describen las funcionalidades más importantes en el sistema COBOT.

2.1. Recepción de mensajes

La recepción de mensajes consta del procedimiento para recibir la información de configuración de la obra y las acciones seleccionadas por el usuario. Para lograr la conexión con el usuario, se implementó un servidor que utiliza el protocolo TCP/IP para crear una conexión de tipo cliente/servidor. Se tiene un listado de las direcciones IP de los clientes que se conectan al servidor y paralelo a este un listado del temporizador de cada cliente. Al llegar una petición se valida que la IP no exista y se instancia el objeto que almacena toda la información del robot actor de un usuario. El elemento que gestiona la información de la obra de teatro y la comunicación con los demás componentes es el *Gestor de Obra*.

```
Enviar datos al servidor de IP 192.168.100.44 y puerto 1235
-----Servidor escuchando-----

-----Conexión recibida-----
Iniciando un nuevo gestor de obra para el usuario con IP 192.168.100.178

-----Servidor escuchando-----

Información recibida para el robot nao: 127.0.0.1
-----Conexión recibida-----
Iniciando un nuevo gestor de obra para el usuario con IP 192.168.100.179

-----Servidor escuchando-----

Información recibida para el robot nao: 192.168.0.12
```

Figura 15. Ejecución del servidor durante recepción de datos.

La figura muestra la ejecución del servidor al momento de recibir datos desde dos dispositivos móviles distintos, instanciando el gestor de obra para cada uno y almacenando la información de conexión con el robot suministrada en el mensaje. El servidor soporta un máximo de 10 clientes concurrentes para los cuales crea un hilo de ejecución propio que recibe los mensajes de cada uno y establece un tiempo de espera de 240 segundos. Cuando se alcanza el tiempo de espera de un cliente este se desconecta liberando su espacio.

```
-----Servidor escuchando-----
Información recibida para el robot nao: 192.168.0.12

-----Conexión recibida-----
Accediendo al gestor de obra del usuario con IP: 192.168.100.179

-----Servidor escuchando-----
Cargado signo de vida Mirar alrededor

Cargado signo de vida Mirar la hora
Cargado signo de vida Pensar
El momento del gestor se ha establecido en INACTIVO
Iniciando timer para el gestor de 192.168.100.179
```

Figura 16. Arribo de mensajes de signos de vida al servidor.

La figura anterior describe el momento cuando llegan mensajes al servidor con los signos de vida del personaje seleccionado en el aplicativo de pruebas. Cada signo de vida se almacena como instancia de objeto en el *gestor de obra* del cliente, para luego establecer el primer momento de ejecución de acciones que es *INACTIVO*. Cuando este momento se establece se inicializa el conteo del temporizador para el gestor del cliente de tal forma que se ejecuten acciones durante la espera o inactividad del usuario del aplicativo. Cabe mencionar que se escogió este evento de mensajes como el primer caso que activa el temporizador ya que es el momento donde se recibe la información del personaje.

2.2. Modulación de acciones

La modulación de acciones consiste en que para cada acción se consulta su mapeo en el *perfil del robot* según el tipo seleccionado. El agente encargado de modular las acciones es el *action modulator*, el cual carga información del robot leyendo un archivo en el computador llamado *perfil del robot*. El *perfil del robot* es un archivo de extensión *JSON* que almacena todos los comandos y sus parámetros. Cada comando debe tener la siguiente información:

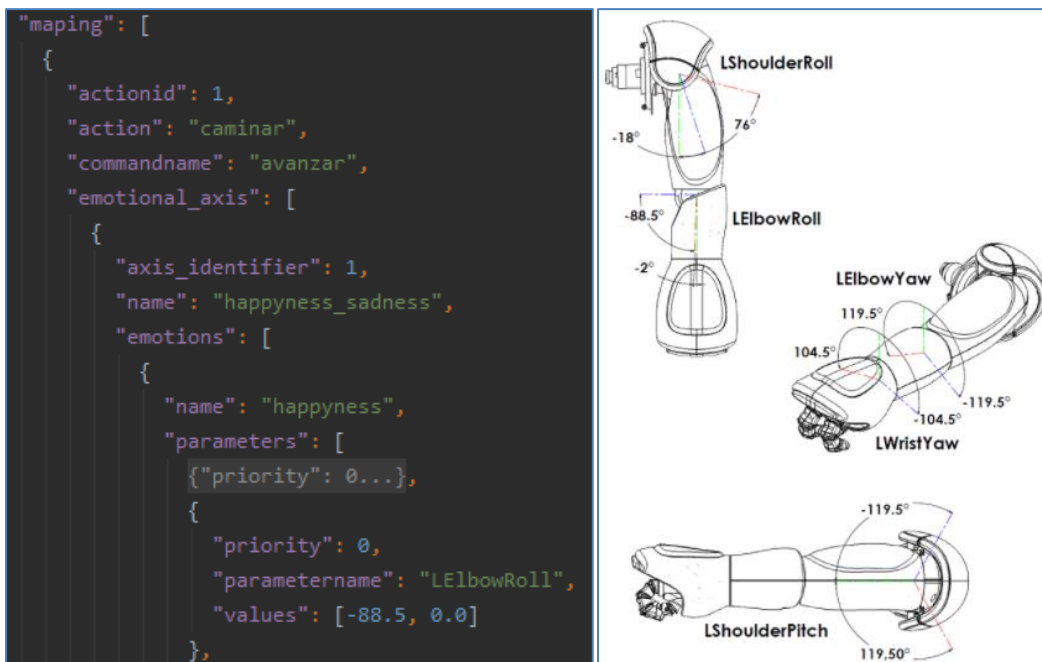


Figura 17. Estructura del archivo del perfil del robot.

La figura 17 muestra la estructura del archivo del *perfil del robot* que permite representar el rango de movimiento de actuadores y motores de un robot, tales como los que se observan en los brazos del robot NAO, cuyos movimientos tienen un límite de acuerdo con el ángulo que se toma. Estos valores varían según la morfología de cada robot. Además, como se puede observar los parámetros se clasifican según la emoción permitiendo que un comando pueda ser ejecutado de varias formas diferentes de acuerdo con esta. La estructura general del archivo de comandos debe tener lo siguiente:

- **Nombre del comando e identificador numérico:** Estos datos se utilizan para hallar la acción seleccionada por el usuario en el perfil y clasificar todos los parámetros según el comando que los representa. Puede haber acciones que tengan el mismo comando, como por ejemplo caminar y correr que corresponden al comando *avanzar*.
- **Mapeo de comandos secundarios** (opcional).
- **Mapeo de comandos sin modulación** (opcional).
- **Parámetros:** Cada uno debe tener:
 - **Nombre:** Es el valor utilizado para identificar el parámetro respecto a los demás, y con él se detectan parámetros que se repiten en dos o más acciones seleccionadas.
 - **Rango de valores:** Este rango consta principalmente de valores numéricos. En comandos no modulados o aquellos que referencian archivos externos, se almacenan los nombres de estos.
 - **Prioridad:** Es un valor entre 0 y 2, que sirve para reconocer la importancia del parámetro en la acción seleccionada. Esta prioridad es un componente importante para decidir cuál parámetro conservar cuando se seleccionan varias acciones que comparten el parámetro (p. ej. caminar o hablar utilizan los leds del robot como enriquecimiento para la acción, por lo que debe escogerse uno de ellos solamente).

Luego de realizar la lectura del archivo y cargar los comandos de las acciones seleccionadas en memoria, el *action modulator* aplica a los comandos provenientes del mapeo general y

secundario la ecuación de modulación de acciones propuesta para este proyecto. En el aplicativo de pruebas se tiene un selector de intensidad emocional en el rango de 0 a 100, el cual al ejecutar las acciones se convierte al rango del modelo emocional cuyo rango va de -1 hasta 1. La conversión se aplica realizando la siguiente ecuación:

$$\text{Intensidad emocional} = \frac{(\text{valor seleccionado} - 50)}{50} \quad (2)$$

Figura 18. Ecuación que representa la intensidad emocional del robot.

Este valor es enviado al servidor como dato de entrada cuando recibe la información de la emoción. Posterior a ello, se aplica la ecuación propuesta en la sección del diseño en el presente documento para cada parámetro modulable

Luego de modular todos los parámetros de los comandos, estos se muestran en la consola del servidor para así evidenciar en qué se ha mapeado la acción seleccionada:

```
Cargando el eje emocional happiness_sadness de valor 0.82
Cargando la acción de id: 1, valor: 6
El momento del gestor se ha establecido en EJECUTANDO ACCIONES SIMPLES
Iniciando modulación de acciones simples y carga de comandos
Cargando mapeo de acciones simples...
-- Comandos y parámetros modulados--
('Comando: avanzar', ', Nombre del parametro: speed', ', valor(es): 0.955')
('Comando: avanzar', ', Nombre del parametro: sound', ', valor(es): SuspirosFelices.wav')
('Comando: avanzar', ', Nombre del parametro: ledcolor', ', valor(es): [232.04999999999998, 255.0, 0.0]')
('Comando: avanzar', ', Nombre del parametro: avanzar', ', valor(es): 6')
('Comando: avanzar', ', Nombre del parametro: HeadPitch', ', valor(es): -14.55')
```

Figura 19. Pruebas de recepción de mensajes de modulación en consola.

Como se observa en la figura 19, se obtuvieron los comandos para la acción *caminar* cuyos parámetros fueron modulados a partir de la intensidad emocional 0.82. Luego de este proceso se priorizan los parámetros modulados con el fin de impedir que se ejecuten parámetros del mismo nombre que involucran los mismos actuadores en el robot, causando problemas de concurrencia e impidiendo la ejecución del resto de comandos. El parámetro que tiene el mismo nombre que el comando (en el caso de la figura es *avanzar*) representa la opción se-

leccionada por el usuario en el aplicativo. En este caso representa el nodo destino en el desplazamiento, para poder calcular la trayectoria del robot. Finalmente, los comandos son retornados al *gestor de obra* para iniciar la ejecución de estos.

2.3. Ejecución de acciones

La ejecución de acciones consiste en el envío de los comandos modulados al robot para que sean ejecutados en su sistema. Para este proyecto la ejecución de comandos se implementó de dos maneras diferentes en el agente llamado *action executor*.

Por un lado, para las pruebas de integración y con usuarios se utilizó el robot NAO en el simulador Choregraphe, el cual ofrece un API llamada *naoqi* en varios lenguajes de programación incluido Python. Se implementó un agente que recibe los comandos del robot NAO y los convierte en comandos legibles por el sistema operativo del robot. Este agente es implementado en el servidor ya que utiliza el API en el mismo lenguaje para que le envíe los comandos al robot directamente para su ejecución.

El robot NAO ejecuta sus comandos al instanciar un objeto Proxy que contiene la información necesaria para organizar los comandos en forma entendible para el sistema del robot. Este Proxy envía los comandos correspondientes a una acción utilizando el protocolo TCP/IP. Los tipos de Proxy del API de NAO utilizados fueron:

- **ALMotion:** Proxy encargado de enviar los comandos que representan el movimiento de los motores del robot NAO. Este Proxy es utilizado también para el enriquecimiento de acciones tales como el movimiento de brazos y cabeza y es considerado el componente del API más importante ya que se encarga de todos los movimientos del robot. Dependiendo del tipo de acción que se requiera, el movimiento puede solicitar datos del perfil del robot como: ángulos de rotación, tiempos de rotación, velocidad de rotación, entre otros.
- **ALRobotPosture:** Proxy encargado de establecer una postura del robot según el listado de posturas que ofrece el robot. Es utilizado para establecer una postura segura previa a cada ejecución de acciones de tal forma que se evite en lo posible caídas o golpes tanto en el entorno simulado como el real. Este Proxy también permite variar la velocidad con

- la que el robot cambia de postura lo que ayuda a tener una transición de movimientos más fluida.
- **ALAudioPlayer:** Proxy encargado de la carga y reproducción de archivos de sonido. Estos sonidos son ubicados en un directorio del servidor llamado *Sounds* y este a su vez se divide en los sonidos que representan diálogos y los sonidos de enriquecimiento para acciones que no involucran directamente los parlantes del robot.
 - **ALLeds:** Proxy encargado de manejar el color de los leds en el cuerpo de NAO. Para este proyecto se pretendió utilizar el color de los leds como enriquecimiento de las acciones, pero en el simulador no se puede observar el color de ninguno. Por tanto, este proxy se utiliza para mostrar el color en formato RGB que el robot físico mostraría.
 - **Otros proxys:** Se propuso utilizar el proxy ALTextToSpeech para que el robot leyera con su voz un texto correspondiente a un fragmento del guion, de esta manera no era necesario cargar archivos de sonido para los diálogos de la obra de teatro. Sin embargo, dado que no se pudo utilizar el robot físico, el simulador no ofreció la funcionalidad por lo que se utilizó ALAudioPlayer para los diálogos como medida de contingencia. No obstante, fue implementado el ejecutor para el proxy ALTextToSpeech de todas formas para futuras implementaciones que puedan usar el robot físico.

Por otro lado, para permitir el envío de comandos hacia otros modelos de robot que no tengan un API en el lenguaje del servidor, se implementó el empaquetado y envío de los comandos por medio de sockets TCP/IP. El protocolo fue utilizado porque permite enviar objetos en el formato JSON y así mismo es utilizado por el API de NAO y en el servidor para recepción de mensajes del cliente. El formato JSON permite estructurar el contenido del mensaje de forma ordenada y serializable en diferentes lenguajes de programación de robots que lo admiten tales como Arduino o C++. Esta aproximación permite la adaptabilidad del modelo BDI-CHA.

Como se mencionó en las restricciones del sistema se requiere que el robot tenga una tarjeta de red que le permita conectarse a una red local y que el lenguaje de programación de su sistema permita desempaquetar objetos JSON provenientes de un socket cliente. Para este

proyecto se propuso utilizar el robot Quyca de la Pontificia Universidad Javeriana. Sin embargo, dadas las condiciones sociales que impidieron de llevar los robots a las residencias de los miembros del grupo y que el modelo a utilizar no fuera terminado y entregado por el ingeniero encargado de tal labor, las pruebas de esta implementación consistieron en visualizar el envío de comandos por medio del cliente socket.

2.4. Generación de trayectorias

Para las acciones que requieren que el robot se desplace de un punto a otro, el agente *action executor* hace uso del agente *world model* siguiendo el flujo mostrado en la figura 4 del diseño de la solución del problema. Para la generación de la trayectoria se utilizó el algoritmo de búsqueda de caminos mínimos *Dijkstra*. Este algoritmo recibe como parámetros de entrada el punto de origen y destino del desplazamiento y retorna el listado de nodos (puntos específicos en el escenario a los cuales los robots pueden desplazarse) que corresponden al camino más corto para que el robot vaya a su destino.

El algoritmo utiliza un grafo que representa el escenario de la obra, instanciado como una matriz de adyacencias de n filas por n columnas, cuya celda representa la existencia de una conexión entre un nodo con otro. Se crea un listado de distancias por cada nodo del grafo y se calcula la distancia mínima para llegar desde el nodo origen hasta cualquier otro nodo mediante comparaciones entre los valores de estos. Al final de la ejecución se busca el camino cuyo destino corresponda al ingresado como parámetro y se devuelven los índices de los nodos que conforman la ruta.

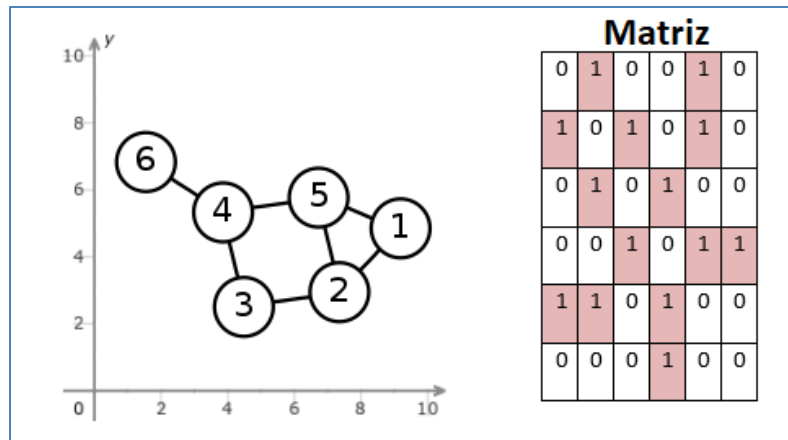


Figura 20. Representación de matriz y grafo usados para el trabajo de grado.

Como se observa en la figura 20, cada fila y columna de la matriz de adyacencias representa un nodo del grafo, por lo tanto, en la casilla de la fila i y una columna j se almacena el valor 1 si estos dos nodos están conectados y 0 si no lo están (p. ej. la fila 1 columna 2 indica que el nodo 1 tiene conexión directa con el 2). No obstante, para que el robot conozca exactamente la distancia que separa un nodo de otro, estos contienen la información de su ubicación con respecto a un plano cartesiano de dos dimensiones. Por lo tanto, al momento de calcular el listado de nodos que corresponden a un camino se calcula la distancia euclidiana de un nodo con el siguiente para finalizar con el proceso.

Para la implementación del ejecutor del robot NAO, este se desplaza caminando hacia adelante y rotando su cuerpo cuando llega a un nodo intermedio. Dado lo anterior, fue necesario implementar un sistema de rotación dado un ángulo inicial con respecto al eje de las abscisas. Al finalizar la ejecución de acciones de desplazamiento, se retorna la nueva posición del robot que corresponde a su posición de destino y el ángulo al cual se encuentra orientado luego de recorrer el escenario.

2.5. Prototipo del aplicativo móvil de pruebas

La aplicación móvil se desarrolló para realizar las pruebas finales del sistema COBOT. La versión fue puesta a prueba en las pruebas de usuario. A continuación, se muestra la pantalla principal para el envío de las acciones y la cual es con la que el usuario interactuará durante

la obra. Para ver el resto de las pantallas de la aplicación ([ver Anexo 15: Manual de uso aplicación móvil](#)).

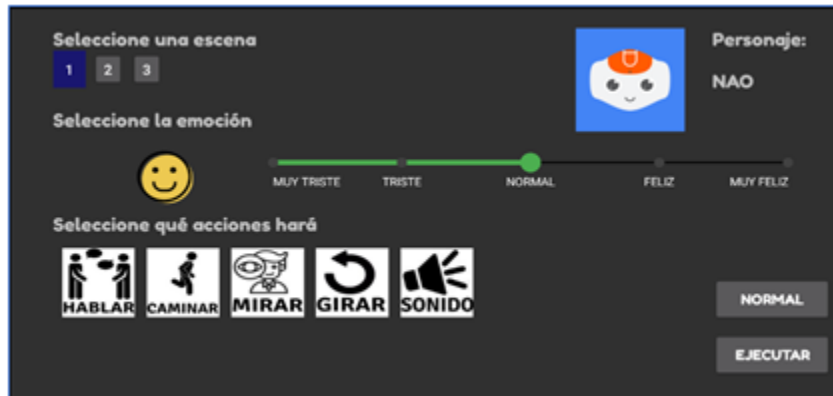


Figura 21. Pantalla principal envío de acciones.

El aplicativo agrupa las acciones a ejecutar según la escena de la obra. Cada una de las acciones al seleccionarse despliega las opciones que esta puede aplicar (p. ej. el destino de desplazamiento o el diálogo que debe decir el robot). La interfaz adicionalmente bloquea aquellas acciones que no se pueden ejecutar al mismo tiempo, tales como caminar y girar, ya que comparten casi todos los actuadores.

En la interfaz se implementó un sistema interactivo para seleccionar la intensidad de la emoción según un rango de 0 a 100 que es adaptado al rango del modelo emocional del grupo SIRP como se explicó en la implementación de la modulación de acciones. Tal selección de intensidad despliega en el botón de acciones emergentes la acción que se puede ejecutar al llegar a un número específico. Esta interfaz permite seleccionar una o más acciones para su ejecución, impidiendo directamente que acciones que usen el mismo actuador o sensor se seleccionen al mismo tiempo.

Al seleccionar todas las acciones y la emoción se selecciona el botón ejecutar, el cual envía este conjunto en formato JSON utilizando el protocolo TCP/IP, y también despliega en ventanas de diálogo temporales el estado de ejecución de las acciones.

VII- RESULTADOS

Esta sección muestra los resultados obtenidos de las pruebas al prototipo realizadas según la metodología de verificación definida en el *Project Management Plan*.

1. Pruebas realizadas

Se realizaron tres tipos de pruebas para verificar el funcionamiento de este y el cumplimiento de los objetivos de este proyecto, los cuales se definen a continuación:

- **Pruebas unitarias:** Se realizaron pruebas unitarias a los componentes del sistema COBOT con el fin de verificar que se cumplan las funcionalidades en diferentes escenarios con diferentes entradas al sistema. La verificación también muestra el porcentaje de cubrimiento que tienen las pruebas con respecto al código fuente.
- **Pruebas de concepto:** Estas pruebas se realizaron con múltiples usuarios de diferentes edades de manera anónima, ellos resolvían un formulario donde se mostraban videos de acciones individuales que realizaba el robot NAO. Estas pruebas se realizaron en dos ocasiones diferentes para corregir el prototipo del formulario y la expresividad de las acciones según los resultados de la primera para posteriormente volver a evaluarse. Estas pruebas fueron previas a las pruebas de aceptación de usuario y evalúan la percepción de la emoción según los comportamientos verbales y no verbales del robot.
- **Pruebas UAT (User Acceptance Test):** La prueba final consistió en realizar una obra de teatro en compañía de un niño de la edad objetivo (7 a 11 años) con el fin de evaluar la percepción de las emociones y la expresividad del robot al agrupar todas las acciones en una obra de teatro tele operada por él. Para esta prueba también se evaluó la usabilidad del sistema de pruebas, es decir, la aplicación móvil ([ver Anexo 23: Resultados pruebas aplicación móvil](#)) y se realizan preguntas durante y después de la obra.

1.1. Pruebas unitarias

Las pruebas fueron realizadas para cada agente del sistema COBOT, los resultados se resumen en la tabla de cubrimiento de las pruebas realizadas y se otorga una calificación de acuerdo con su porcentaje. Un cubrimiento superior al 70% es una meta razonable para las pruebas de software [24].

Nombre del agente	Porcentaje de cubrimiento	Puntaje
Gestor de obra	97%	4.8
Action Modulator	93%	4.6
Action Executor	80%	4.0
Perfil del Robot	93%	4.6
Modelo del mundo	99%	4.9

Tabla 9. Resultados pruebas unitarias en el sistema COBOT.

1.2. Pruebas de concepto

El objetivo de estas pruebas fue identificar la percepción del usuario con respecto a la ejecución del nivel emocional de diferentes acciones. Se implementó la siguiente prueba con un tipo variable de público, es decir, cualquier edad desde los 7 hasta los 60 años.

La razón por la cual se amplió la edad del público objetivo para esta prueba fue debido a las condiciones sociales del año en curso (2020), las cuales impidieron establecer contacto con entidades escolares por lo que se evaluaron personas de diferentes edades. Por lo tanto, se desarrolló un formulario masivo donde se realizaron diferentes preguntas. A continuación, se observa la frecuencia de cada rango de edad, como se observa en la tabla 9 en total la cantidad de usuarios encuestados fue de 68 personas. Para ver las pruebas por caso de uso ([ver Anexo 24: Resultados pruebas funcionalidades](#))([ver Anexo 25: Pruebas de casos de uso](#)).

Fue necesario dividir el formulario en tres formularios más pequeños debido a su extensión ([ver Anexo 10: Protocolo pruebas de concepto](#)).

Edad	Frecuencia
Entre 7 y 11 años	21
Entre 12 y 18 años	8
Entre 19 y 26 años	18

Entre 27 y 59 años	21
Mayor de 60 años	0
Total	68

Tabla 10. Edad y frecuencia de las personas encuestadas.

Para realizar la medición de las expresiones de cada video, se les solicitó a las personas calificar utilizando la siguiente escala de Likert [25] para las acciones simples de caminar, mirar y girar:

- Extremadamente feliz / triste, cuyo valor es 5 puntos.
- Muy feliz / triste, cuyo valor es 4 puntos.
- Ligeramente feliz / triste, cuyo valor es de 3 puntos.
- Nada feliz / triste, cuyo valor es de 2 puntos.
- Otro, es decir, el robot está realizando otra acción con otra emoción, cuyo valor es de 1 punto.

Así mismo, para las acciones emergentes y signos de vida se presentó la siguiente escala para identificar si el usuario estaba de acuerdo con la acción y la emoción que se le presentó en el video:

- Lo expresa totalmente, cuyo valor es 5 puntos
- Si lo expresa, cuyo valor es 4 puntos.
- Ligeramente lo expresa, cuyo valor es de 3 puntos.
- No lo expresa, cuyo valor es de 2 puntos.
- Otro, El robot está expresando otra acción con otra emoción, cuyo valor es de 1 punto.

Los puntajes se ponderan de acuerdo con la cantidad de personas que respondieron según la escala. El objetivo descrito para los resultados de los formularios en la fase de verificación es que más del 50% de los encuestados perciban la emoción que expresa el robot en los videos. Los resultados del diligenciamiento de los formularios fueron los siguientes:

Acción y emoción	Pregunta	# Personas	Puntaje ponderado redondeado
Caminar muy triste	¿Qué tan triste se encuentra NAO cuando camina?	35	3.25

Caminar muy feliz	¿Qué tan feliz se encuentra NAO cuando camina?	35	2.91
Caminar triste	¿Qué tan triste se encuentra NAO cuando camina?	35	3.11
Caminar feliz	¿Qué tan feliz se encuentra NAO cuando camina?	35	2.8
Mirar muy triste	¿Qué tan triste se encuentra NAO cuando mira hacia un lado?	22	2.86
Mirar muy feliz	¿Qué tan feliz se encuentra NAO cuando mira hacia un lado?	22	2.5
Mirar triste	¿Qué tan triste se encuentra NAO cuando mira hacia un lado?	22	3.41
Mirar feliz	¿Qué tan feliz se encuentra NAO cuando mira hacia un lado?	22	2.5
Girar muy triste	¿Qué tan triste se encuentra NAO cuando gira?	21	3.52
Girar muy feliz	¿Qué tan feliz se encuentra NAO cuando gira?	21	3.1
Girar triste	¿Qué tan triste se encuentra NAO cuando gira?	21	3.57
Girar feliz	¿Qué tan feliz se encuentra NAO cuando gira?	21	3.14
Celebrar	¿Crees que NAO expresa que se encuentra extremadamente feliz?	68	3.88
Llorar	¿Crees que NAO expresa que se encuentra extremadamente triste?	68	4.2
Mirar alrededor feliz	¿Crees que NAO expresa que se encuentra mirando feliz alrededor?	68	3.36
Mirar alrededor triste	¿Crees que NAO expresa que se encuentra mirando triste alrededor?	68	3.72
Pensar feliz	¿Crees que NAO expresa que se encuentra pensando feliz?	68	2.94
Pensar triste	¿Crees que NAO expresa que se encuentra pensando triste?	68	3.49

Tabla 11. Resultados formularios.

Análisis de los resultados para las acciones simples: Los resultados de los formularios enviados muestran que las expresiones del robot tanto para felicidad como para tristeza en las

acciones simples son ligeramente notorias. Esto quiere decir que el robot por sí sólo sin apoyo de diálogos o sonidos y dadas sus restricciones morfológicas, logra transmitir a más del 50% de los encuestados ya que la puntuación ponderada supera el intermedio (2.5 puntos).

Se identificó en las respuestas del formulario que algunas personas encuestadas no percibieron la expresión que el robot realizaba por factores como:

- Gran parte de los encuestados en las recomendaciones de mejora solicitaron que el robot tuviera expresiones faciales tales como sonrisa, llanto, etc. A pesar de que se les indicó al comienzo la incapacidad de NAO para realizar tales expresiones dada su morfología, ellos consideraron que hacían falta.
- Las pruebas de las acciones simples se realizaron sin apoyo de sonido ya que en una prueba previa se identificó el sesgo de los usuarios dados los sonidos, es por esto por lo que se quiso evaluar las acciones no verbales moduladas según el nivel de tristeza o felicidad sin el recurso auditivo. Por lo tanto, para algunos usuarios fue extraño o incompleto y propusieron agregar sonidos a las acciones.

Análisis de los resultados para las acciones emergentes: Con respecto a las acciones emergentes mostradas: celebrar y llorar, estas dos fueron las que mejor percepción tuvieron por los usuarios, ya que estas expresiones se apoyan de efectos de sonido y de movimientos más expresivos. La puntuación ponderada fue aproximadamente de 4, lo que en la escala indica que se encontraba muy feliz o triste en el caso correspondiente. En las observaciones dadas reinciden en solicitar expresiones faciales a pesar de que no sea posible por el tipo de robot utilizado. Sin embargo, los usuarios también aportaron comentando que puede mover los brazos con más velocidad o fluidez.

Análisis de los resultados para las acciones de signos de vida: Los signos de vida mostrados fueron dos: mirar alrededor y pensar. Existe un contraste en los dos signos de vida debido a que el primero tiene una calificación redondeada a 4, lo cual indica que se percibe el estado emocional del robot, pero la acción de pensar no. Esto se debe a que los usuarios no notaron una clara diferencia entre la forma como se expresaba el robot al

estar feliz o triste. Las recomendaciones dadas para el trabajo futuro fueron realizar diferentes tipos de acciones de signos de vida y nuevamente el refuerzo de las expresiones faciales.

En una vista general los resultados aportaron a identificar qué aspectos son posibles para mejorar tales como el movimiento de brazos, que fue el más común entre las recomendaciones.

1.3. Prueba UAT (User Acceptance Test)

La prueba final se realizó con ocho usuarios entre 7 y 11 años de manera presencial, a los cuales se les presentó el guion de una obra de teatro llamada “NAO: un rescate galáctico” ([ver Anexo 18: Obra pruebas finales](#)). Se les indicó que, con ayuda de la aplicación móvil, enviaran las instrucciones del guion utilizando la interfaz en la cual se mostraban las acciones y el nivel emocional que debía tener el robot actor NAO para que las ejecutara en tiempo real. Al inicio de la prueba se le solicitó a cada usuario sus datos demográficos preguntas ([ver Anexo 5: Datos recolectados prueba final](#)), luego en tiempo de ejecución y al finalizar la prueba se le hicieron diferentes. Las preguntas se realizan con el fin de conocer su opinión frente a las acciones que realizaba el robot con el nivel emocional que el usuario indicaba en la aplicación. Para conocer paso a paso la metodología implementada en esta prueba ver ([ver Anexo 17: Protocolo Pruebas Finales](#)).

Cabe resaltar el propósito inicial era realizar la prueba final con un número mayor de usuarios se pretendía hacerla en un colegio con niños entre 7 y 11 años, pero la propuesta fue detenida a causa de las situaciones sociales del presente año. Entonces, se optó por buscar niños y niñas de la edad objetivo que vivieran cerca de las residencias de los miembros del grupo, para luego obtener la autorización firmada por parte de los padres de familia de los menores y por tanto ellos puedan participar en la interacción manual con la aplicación móvil y el robot virtual.

Datos demográficos

Es de destacar que los usuarios con mayor participación fueron menores de 9 años, también, el género que predominó fue el masculino y el grado escolar fue tercero de primaria.

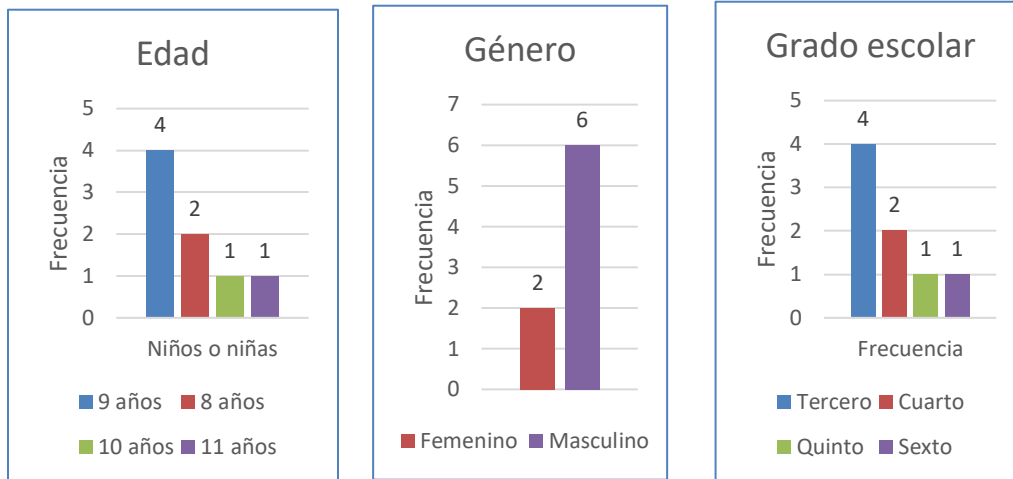


Figura 22. Gráficos demográficos de los usuarios que participaron en la prueba final.

Registro fotográfico de los usuarios en ejecución

Las siguientes fotografías fueron tomadas en tiempo de ejecución de la prueba final, donde se observa a diferentes usuarios interactuando con la aplicación móvil y el robot virtual.

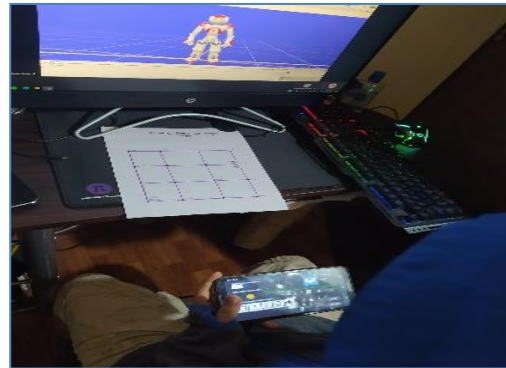
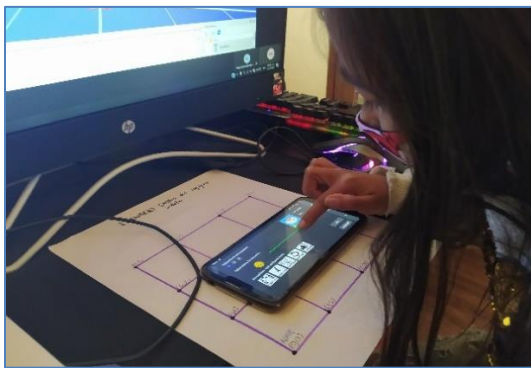


Figura 23. Fotografías tomadas a los usuarios durante la ejecución de la prueba final.

Recolección de datos al finalizar la ejecución

En ejecución se realizaron diferentes preguntas que se dividieron en 6 momentos acordes al guion de la obra ([ver Anexo 18: Obra pruebas finales](#)), donde se consideraba evaluar la percepción del usuario, específicamente con cada una de las acciones propuestas más algún nivel emocional (0-100) entre muy triste y muy feliz.

La escala Likert [25] [26] utilizada para las opciones de respuesta en las siguientes preguntas fue:

- Totalmente de acuerdo, cuyo valor son 5 puntos.
- De acuerdo, cuyo valor son 4 puntos.
- Ni de acuerdo ni en desacuerdo, cuyo valor son 3 puntos.
- En desacuerdo, cuyo valor son 2 puntos.
- Totalmente en desacuerdo, cuyo valor es 1 punto.

Los resultados se recopilan en la siguiente tabla:

Momento #1: NAO se encuentra al lado de su nave, comienza a caminar triste con un nivel de 25 hacia la coordenada (1,1), al mismo tiempo mira hacia el lado derecho para ver el entorno	
Pregunta	Puntuación ponderada
¿NAO camino triste hacia la coordenada (1,1)?	4.38
¿NAO miro hacia su alrededor triste?	4.13
¿NAO mostró adecuadamente el nivel de tristeza que indicaba el guion?	4.0

Tabla 12. Resultados ejecución momento número uno.

Momento #2: Y en ese instante de profunda tristeza, NAO comienza a llorar	
Pregunta	Puntuación ponderada
¿NAO mostró adecuadamente el nivel de tristeza que indicaba el guion?	4.25

Tabla 13. Resultados ejecución momento número dos.

Momento #3: Cuando NAO escucho la voz de sus amigos dejo de estar triste y comenzó a caminar feliz con un nivel de 60 hacia la coordenada (2,1) para poder llegar a su destino final	
Pregunta	Puntuación ponderada

¿NAO camino feliz hacia la coordenada (2,1)?	4.75
¿NAO mostró adecuadamente el nivel de felicidad que indicaba el guion?	4.13

Tabla 14. Resultados ejecución momento número tres.

Momento #4: Luego, NAO camina muy feliz con un nivel de 80 hacia la coordenada final para por fin encontrarse con sus amigos.	
Pregunta	Puntuación ponderada
¿NAO camino muy feliz hacia la coordenada final?	4.75
¿NAO mostró adecuadamente el nivel de felicidad que indicaba el guion?	4.5

Tabla 15. Resultados ejecución momento número cuatro.

Momento #5: NAO espera un tiempo, pero no recibe respuesta de sus amigos. NAO se gira hacia atrás triste con un nivel de 20 para ver si localiza a su equipo detrás suyo.	
Pregunta	Puntuación ponderada
¿NAO giró triste hacia atrás?	4.5
¿NAO mostró adecuadamente el nivel de tristeza que indicaba el guion?	4.25

Tabla 16. Resultados ejecución momento número cinco.

Momento #6: De repente se ilumina una luz en el suelo, esa es la luz de la nave del equipo de NAO, él empieza a celebrar diciendo “¡Yupi!”.	
Pregunta	Puntuación ponderada
¿NAO mostró adecuadamente el nivel de felicidad que indicaba el guion?	5.0

Tabla 17. Resultados ejecución momento número seis.

Análisis de los resultados para los momentos en ejecución: En general todos los momentos tuvieron calificaciones ponderadas superiores a 4.0, lo cual quiere decir que están de acuerdo con la expresión de las emociones justo después que el robot las ejecutara sin excepción. Además, en las preguntas abiertas realizadas ellos comentan que para mejorar las expresiones del robot este se podría desplazar más rápidamente de un lugar a otro y también mejorar el movimiento de los brazos, al igual que las personas encuestadas en la prueba de concepto ([ver Anexo 4: Datos recolectados prueba de concepto](#)). No obstante, en estas pruebas también fue común que recomendaran las expresiones faciales que el robot actualmente no puede realizar.

Recolección de datos al finalizar la prueba:

Al finalizar la prueba se hicieron las siguientes preguntas a los niños, con el fin de dar una calificación final al resultado de la ejecución de acciones y dar retroalimentación sobre su experiencia y recomendaciones para trabajos futuros. Los resultados ponderados según la escala de Likert utilizada para la recolección de datos en ejecución son los siguientes:

Pregunta	Puntaje ponderado redondeado
¿NAO interpretó adecuadamente las emociones y las acciones que se indicaban en el guion?	4.38
¿Se diferenció cuando NAO estaba un poco feliz o muy Feliz?	4.75
¿Se diferenció cuando NAO estaba un poco triste o muy triste?	4.5
¿NAO expresó su nivel más alto de tristeza llorando?	4.0
¿NAO expresó su nivel más alto de felicidad al celebrar?	4.38
¿NAO realizó acciones cuando estaba quieto, es decir, cuando no le enviabas acciones?	4.38
¿Te gustaría ver a NAO representando diferentes obras de teatro?	4.75

Tabla 18. Resultados preguntas sobre ejecución de acciones.

Análisis de los resultados para los momentos al final de la prueba: Los resultados obtuvieron puntuaciones de un promedio de 4.45, lo que implica que la mayoría estuvo de acuerdo con la expresión emocional del robot durante toda la obra y les pareció entretenida y enriquecedora la experiencia ([ver Anexo 22: Resultados prueba final usuarios](#)). Para todas las pruebas con cada uno de los niños se obtuvo un consentimiento informado ([ver Anexo 2: Consentimientos informados diligenciados](#)) y se grabó la pantalla y el audio durante toda la ejecución de la prueba. Las observaciones para esta prueba son las siguientes:

- La ejecución en ocasiones tardaba varios segundos debido a que previo a la ejecución de las acciones enviadas desde el aplicativo, se disparaban las acciones que corresponden a signos de vida. Estas acciones pueden interrumpirse, sin embargo, el robot virtual puede correr el riesgo de caerse cuando se interrumpe un movimiento durante su ejecución, por lo que se optó por esperar a la finalización de cada una para evitar este problema. En dos ocasiones el robot perdió conexión

dada una caída al interrumpir las acciones, por lo que se tuvo que reiniciar el simulador para que vuelva a verse el robot como antes y que el niño no perdiera interés en lo que veía.

- Uno de los sujetos participantes que hacía la prueba no percibió las acciones de signos de vida, debido a que estas se ejecutaron casualmente antes del envío de acciones que ella hizo desde el aplicativo, por lo que pensó que era un movimiento que se hace cada vez que va a realizar acciones entrantes. Para solucionar esto es posible cambiar el intervalo de tiempo en el que se disparan los signos de vida para así estimar mejor los tiempos y evitar esperas.
- A todos los niños se les explicó las restricciones que hay en cuanto a los movimientos del robot, es decir, que este no puede realizar acciones de la misma manera o con la misma velocidad que un ser humano. Sin embargo, en las recomendaciones para mejorar movimientos mencionaron aspectos como:
 - Llorar: Que el robot llore cuando está muy triste.
 - Saltar: En un momento de mucha felicidad que salte de alegría.
 - Caminar como humano: Que la forma de caminar sea más parecida a la forma como camina un ser humano.
 - Mover los brazos: Utilizar los brazos como enriquecimiento de las acciones.

De estas recomendaciones se pueden tomar en cuenta para futuros trabajos el movimiento de los brazos, ya que, el resto las restricciones son meramente morfológicas lo que implica que para corregirlas hay que agregar o eliminar partes físicas del robot o programar un simulador diferente que contenga estas nuevas funcionalidades.

VIII- CONCLUSIONES

1. Análisis de impacto del proyecto

El impacto del desarrollo de un proyecto como COBOT se centra en la posibilidad de brindar nuevas formas de incluir a la robótica en el ámbito educativo, bien sea para el aprendizaje o para el entretenimiento.

1.1. Corto plazo

Se considera importante que, a corto plazo, se puedan mejorar los resultados que tuvo el proyecto, o reforzar otros temas de este, como lo pueden ser la conexión realizada con los robots o los movimientos realizados para el enriquecimiento de las acciones. Además de ello, se espera que las condiciones sociales que impidieron trabajar con los robots de manera presencial cambien para poder realizar la implementación y las pruebas en un contexto más cercano al público objetivo.

1.2. Mediano plazo

Se espera que el proyecto sea una base para futuros proyectos que se enfoquen en el teatro robótico, con el fin de aportar a la investigación y uso de los robots dentro de este nuevo ámbito, y que pueda ser usado por personas que les interese el comportamiento de los robots dentro de este tema. Gracias a la implementación del proyecto puede avanzarse en trabajos futuros para facilitar la elaboración de guiones y realizar pruebas con diferentes robots que tengan morfologías distintas. Además, puede integrarse este trabajo con otros realizados por la Universidad Javeriana para ofrecer un producto más completo y con varios años de trabajo.

1.3. Largo plazo

A largo plazo, se espera que el proyecto tenga un gran impacto sobre la inclusión que podría tener la robótica dentro de la vida diaria, dado el crecimiento que ha venido teniendo estos años. Esto, generaría que dicho trabajo se convirtiera en un precedente acerca de lo que puede implicar la robótica, ya sea tanto dentro del teatro, como de la educación, en caso de que allí se usen estrategias referentes a lo mencionado en este proyecto. El contexto del problema evidencia el componente innovador de este proyecto y el bagaje de implementaciones

que conviertan el sistema en uno mucho más robusto y completo, agregando más robots actores a las obras de teatro.

2. Conclusiones

Diseño del sistema: El diseño realizado fue adecuado ya que permitió realizar la implementación del sistema sin cambios drásticos. Se realizó cumpliendo con el modelo BDI-CHA y el gestor de obra permitió hacer la gestión concurrente de acciones.

Implementación del sistema: A partir de la Implementación realizada con el robot NAO se realizó la modulación de las acciones utilizando la mayor cantidad de actuadores que permitiesen la ejecución de las acciones con expresividad. Sin embargo, se debe que mejorar la expresividad de las intensidades emocionales, para esto es necesario tener una gran cantidad de pistas verbales y no verbales, puesto que la claridad de la expresión de las acciones depende de la complejidad del perfil del robot.

Pruebas de concepto: Los resultados de las pruebas realizadas de las expresiones con el robot actor NAO fueron satisfactorias, ya que, a pesar de que en los formularios se percibieron de manera ligera los niveles emocionales, las recomendaciones aportaron para la prueba final. No obstante, en la prueba final estos niveles emocionales fueron identificados en mayor medida porque se representaron en una obra de teatro completa y con una guía personalizada.

Validaciones finales: A pesar de las condiciones sociales del presente año la validación planeada con la obra de teatro se realizó de manera correcta con el público objetivo y de manera presencial. En estas validaciones comprobamos que el sistema es atractivo para los usuarios finales y lograron entender el propósito de la obra de teatro. En las recomendaciones hace falta que el robot pueda expresar con su rostro la emoción que siente y desplazarse con mejor fluidez.

Alcance del proyecto: En un principio en el alcance del proyecto se había establecido trabajar sólo con el robot NAO, pero se ofreció la posibilidad de enviar los comandos modulados a diferentes tipos de robots que cumplan con el requisito de comunicación, también se diseñó

e implemento una interfaz gráfica como lo es el aplicativo móvil el cual es un sistema independiente.

Además de realizar acciones simples para una obra de teatro, se enriqueció el sistema COBOT diseñando e implementando acciones emergentes y signos de vida para mantener activo al robot en la ejecución de la obra de teatro, lo anterior supone un valor agregado al producto final propuesto al comienzo del trabajo ya que incrementa la expresividad del robot.

3. Trabajo futuro

Respecto al trabajo futuro, es importante que se logre automatizar la creación de los archivos tanto de la configuración de la obra, como del perfil del robot por medio de interfaces de usuario. Esto, debido a que, si alguna persona sin conocimientos de programación desea realizar dicha tarea en un futuro próximo, pueda realizarla de la forma más accesible posible. Además de ello, algo que se puede mejorar en caso de realizarse un trabajo futuro, es ampliar la variedad de modelos de robots con el que el usuario puede interactuar, ya que se implementó el envío de las acciones a cualquier robot que cumpla con los requerimientos.

Otro aspecto importante es la participación de más robots en obras de teatro, es decir, que las obras tengan más personajes y que estos conformen una obra más compleja. En este proyecto se trabajó con el robot NAO y se pretendió utilizar el robot Quycá de la Universidad Javeriana para las pruebas de recepción de comandos modulados. No obstante, no fue posible dadas las restricciones sociales actuales y también porque el prototipo del robot no fue terminado por el agente externo encargado. Sin embargo, este trabajo abre la posibilidad de integrarse con otros trabajos de la Universidad Javeriana que utilicen diferentes robots o que ofrezcan nuevas funcionalidades.

Por último, algo a mejorar en un trabajo futuro es la gama de emociones que el sistema puede manejar. Ya que, si bien el hecho de representar dos emociones (felicidad y tristeza) constituyeron todo un proceso de implementación complejo, gracias al desarrollo de este proyecto de grado, es posible ampliar el rango de emociones que podrían coexistir dentro del sistema.

IX- REFERENCIAS

- [1 International Federation of Robotics, «Executive Summary World Robotics 2020 Service Robots,» 2020.
- [2 A. D. Crovo Pérez, «Pontificia Universidad Javeriana,» Mayo 2018. [En línea]. Available: <https://repository.javeriana.edu.co/bitstream/handle/10554/38786/Daniel%20Andr%c3%a9s%20Crovo%20P%c3%a9rez.pdf?sequence=2&isAllowed=y>. [Último acceso: 23 Septiembre 2020].
- [3 T. Motos, «El teatro en la educación secundaria: fundamentos y retos,» *Creatividad y sociedad*, nº 14, pp. 3-5, 2009.
- [4 L. P. C. B. R. A. Antonio Barrientos, Fundamentos De Robótica, Madrid: McGRAW HILL/INTERAMERICANA DE ESPAÑA, S.A, 1997.
- [5 C. Angulo, «Universitat Politècnica de Catalunya,» 13 Enero 2017 . [En línea]. Available: <https://www.upc.edu/latevaupc/usos-y-beneficios-robotica-las-aulas/#:~:text=En%20educaci%C3%B3n%20pueden%20diferenciarse%20dos,de%20manera%20l%C3%B3gica%20y%20crear>. [Último acceso: 23 Septiembre 2020].
- [6 P. Baldwin y K. Fleming, Teaching literacy through drama: Creative approaches, 2003.
- [7 T. Motos, «Las técnicas dramáticas: procedimiento didáctico para la enseñanza de la lengua y la literatura,» *Enseñanza: Anuario Interuniversitario de Didáctica*, Vols. %1 de %210-11, nº ISSN 0212-5374, pp. 75-94, 1993.
- [8 M. Navarro, «Drama, creatividad y aprendizaje vivencial: algunas aportaciones del drama a la educación emocional,» *Cuestiones Pedagógicas: Revista de Ciencias de La Educación*, vol. 18, p. 161–172, 2006.
- [9 M. Cañete, «Los artistas, la cultura y los medios El teatro como comunicación,» *Escritos en la Facultad*, vol. 126, pp. 18-20, 2017.

- [1 IBM, «IBM Knowledge Center,» [En línea]. Available:
0] https://www.ibm.com/support/knowledgecenter/es/ssw_aix_72/network/tcpip_protocols.html. [Último acceso: 30 Octubre 2020].
- [1 Real Academia Española, «<https://dle.rae.es/robot>,» Diccionario de la lengua española,
1] 2019. [En línea]. Available: <https://dle.rae.es/robot>.
- [1 R. Simmons, «Concurrent Planning and Execution for autonomous robots,» IEEE, Nueva
2] York, 1988.
- [1 K. C. P. Robinson, «Concurrent Task Programming of Robotic,» School of ITEE, University
3] of Queensland, Brisbane, Londres.
- [1 R. A. E. G. Alejandra González, «BDI CONCURRENT ARCHITECTURE,» 2013.
4]
- [1 F. A. Bravo, E. González y A. M. González, «ACM Digital Library,» Septiembre 2017. [En
5] línea]. Available: <https://dl.acm.org/doi/10.5898/JHRI.6.2.Bravo>. [Último acceso: 10
Abril 2020].
- [1 M. Mateas, «An Oz-Centric Review of Interactive Drama and Believable Agents,» *Lecture
6] Notes in Computer Science*, vol. 1600, pp. 297-328, 1999.
- [1 A. O. Gerald Clore, «Psychological Construction in the OCC Model of Emotion,»
7] Cambridge University Press, 1988.
- [1 B. Steunebrink, M. Dastani y J. J. Meyer, *The OCC model revisited.*, 2009.
8]
- [1 G. Zeglin, A. Walsman, L. Herlant, Z. Zheng, Y. Guo y M. C. Koval, «HERB's Sure Thing: A
9] rapid drama system for rehearsing and performing live robot theater,» de *International
Workshop on Advanced Robotics and its Social Impacts*, Evanston, IL, USA, 2014.
- [2 K. L. P. F. K. Z. Petrovic D, «Autonomous Robots as Actors in Robotics Theatre - Tribute
0] to the Centenary of R.U.R,» de *European Conference on Mobile Robots (ECMR), Mobile
Robots (ECMR)*, 2019.
- [2 S. O. K. Y. Y. C. T. H. O. & I. H. Nishiguchi, «Theatrical approach: Designing human-like
1] behaviour in humanoid robots,» *Robotics and Autonomous Systems*, pp. 158-166, 2020.

- [2 Pontificia Universidad Javeriana, «Pontificia Universidad Javeriana,» [En línea].
2] Available: <https://www.javeriana.edu.co/investigacion/quemes>. [Último acceso: 29 Septiembre 2020].
- [2 «TutorialsPoint; Unix Socket - Client Server Model,» [En línea]. Available:
3] https://www.tutorialspoint.com/unix_sockets/client_server_model.htm. [Último acceso: 10 9 2020].
- [2 S. Cornett, «bullseye.com,» Bullseye Testing Technology, 2013. [En línea]. Available:
4] <https://www.bullseye.com/minimum.html#:~:text=Summary,higher%20than%20for%20system%20testing..>
- [2 S. McLeod, «simplypsychology; Likert Scale Definition, Examples and Analysis,» [En
5] línea]. Available: <https://www.simplypsychology.org/likert-scale.html>. [Último acceso: 1 11 2020].
- [2 «QuestionPro; ¿Qué es la escala de Likert y cómo utilizarla?,» [En línea]. Available:
6] <https://www.questionpro.com/blog/es/que-es-la-escala-de-likert-y-como-utilizarla/>. [Último acceso: 2 11 2020].
- [2 G. Hoffman, R. Kubat y C. Breazeal, «A hybrid control system for puppeteering a live
7] robotic stage actor,» *IEEE International Symposium on robot and human interactive communication*, pp. 354-359, 2008.
- [2 L. Takayama, D. Dooley y W. Ju, «Expressing thought: Improving robot readability with
8] animation principles,» *ACM/IEEE International Conference on Human-robot Interaction*, pp. 69-76, 2011.
- [2 H. Knight, «Eight lessons learned about non-verbal interactions through robot theater,»
9] *Social Robotics*, pp. 42-51, 2011.
- [3 A. Bruce, J. Knight, S. Listopad, B. Magerko y I. Nourbakhsh, «Robot improv: Using drama
0] to create believable agents,» *IEEE International Conference on robotics and automation*, vol. 4, pp. 4002-4008, 2000.
- [3 «Corporateit,» 23 Abril 2019. [En línea]. Available:
1] <https://www.corporateit.cl/index.php/2019/04/23/idc-mercado-de-robots-tendria-un-crecimiento-del-21-este-2019/#:~:text=IDC%3A%20Mercado%20de%20robots%20tendr%C3%ADa%20un%20cre>

cimiento%20del%2021%25%20este%202019,-
Los%20robots%20para&text=Para%202019%2C%20.

- [3 Google, «Android Developers,» Google, [En línea]. Available:
2] <https://developer.android.com/docs/quality-guidelines/core-app-quality>. [Último acceso: 20 Agosto 2020].
- [3 G. A y V. P., *Humanoid Robotics: A reference*, Dordrecht: Springer, 2019.
3]
- [3 E. González y F. Á. Bravo, «Robots Para Colombia,» Pontificia Universidad Javeriana, [En
4] línea]. Available: <http://robotsparacolombia.com/quyca.html#testimonials>. [Último acceso: 25 Septiembre 2020].
- [3 M. Bratman, *Intention, Plans, and Practical Reason*, Cambridge, 1987.
5]
- [3 A. Rao y M. Georgeff, «BDI-agents: from theory to practice,» de *Proceedings of the First*
6] *Intl. Conference on Multiagent Systems*, San Francisco, 1995.
- [3 J. Gosling, B. Joy, G. Steele, G. Bracha y A. Buckley, *The Java™ Language specification*,
7] 2000.
- [3 G. Van Rossum, «Python programming language,» de *USENIX annual technical*
8] *conference*, 2007.
- [3 Object Management Group, «The Unified Modeling Language,» Object Management
9] Group, [En línea]. Available: <https://www.uml-diagrams.org>. [Último acceso: 26 Octubre 2020].
- [4 Á. Montes, «Semana,» 24 Mayo 2008 . [En línea]. Available:
0] <https://www.semana.com/vida-moderna/articulo/robotica-colombiana/92865-3/>.
[Último acceso: 10 Noviembre 2020].
- [4 Semana, «Semana,» 6 Junio 2020. [En línea]. Available:
1] <https://www.semana.com/tecnologia/articulo/robots-y-desempleo-asi-sera-el-impacto-de-la-tecnologia-tras-la-pandemia/680888/>. [Último acceso: 10 Noviembre 2020].

[4 International Federation of Robotics, «IFR presents World Robotics report 2020,»
2] Frankfurt, 2020.

X- APÉNDICES

Anexo 1: [Carta consentimiento informado.](#)

Anexo 2: [Consentimientos informados diligenciados.](#)

Anexo 3: [Cuestionario usabilidad.](#)

Anexo 4: [Datos recolectados prueba de concepto.](#)

Anexo 5: [Datos recolectados prueba final.](#)

Anexo 6: [Diagramas de diseño proyecto COBOT.](#)

Anexo 7: [Documentación casos de uso.](#)

Anexo 8: [Documentación nuevos requerimientos funcionales.](#)

Anexo 9: [Documentación nuevos requerimientos no funcionales.](#)

Anexo 10: [Protocolo pruebas de concepto.](#)

Anexo 11: [Historias de usuario y planning póker.](#)

Anexo 12: [Informe de reuniones.](#)

Anexo 13: [Manual creación archivo configuración de obra.](#)

Anexo 14: [Manual creación archivo perfil del robot.](#)

Anexo 15: [Manual de uso aplicación móvil.](#)

Anexo 16: [Manual de uso sistema COBOT.](#)

Anexo 17: [Protocolo Pruebas Finales.](#)

Anexo 18: [Obra pruebas finales.](#)

Anexo 19: [Product Backlog.](#)

Anexo 20: [Prototipos de papel versión 1.0.](#)

Anexo 21: [Prototipos de papel versión 2.0.](#)

Anexo 22: [Resultados prueba final usuarios.](#)

Anexo 23: [Resultados pruebas aplicación móvil.](#)

Anexo 24: [Resultados pruebas por funcionalidad.](#)

Anexo 25: [Pruebas casos de uso.](#)