

Formato de Presentación Working Paper

Título: Desarrollo de una interfaz multiplataforma para el control basado en API NAOqi del robot Pepper.

Autores:

Sindy Paola Amaya - S.P. Amaya
Yineth Valentina Vanegas Barbosa - Yineth Vanegas
Valeria Victoria Aguilera Gil - Valeria Aguilera
Diana Alexandra Rincon Caballero - Diana Rincón
Santiago Niño Vargas - Santiago Niño
Maria Alejandra Pinzón Farfán - Alejandra Pinzón
Jorge Ivan Palacios Pardo - Jorge Palacios
Daniel Fernando Castaño Ortiz - Daniel Castaño
Juan Esteban Romero Velásquez - Esteban Romero
Heidy Nicoll Mahecha Pérez - Nicoll Mahecha

Resumen

Los robots sociales están destinados a ser usados por personas sin competencias digitales específicas. Sin embargo, algunos robots sociales como Pepper requieren de la programación en entornos específicos de software como Choreographe y NAOqi, dificultando tareas típicas como el desplazamiento y la comunicación. Se propone en este proyecto el desarrollo de una APK y de programas multiplataforma (en Linux, Windows y Android) para la interfaz de control en tiempo real del robot Pepper mediante la integración de NAOqi, ROS, aplicativos en Linux y Windows, y capas de software Android. Para lograr esto, las funciones de movimiento y habla de la API NAOqi son empaquetadas en nodos y servicios de ROS; estos nodos y servicios publicarán y obtendrán datos en la APK de Android a través de un *web service*. Escenarios comunes para el robot, incluyendo ejercicios conversacionales y coreográficos, serán construidos para validar el software programado, la configuración de ROS trabajada y la sensibilidad del robot. Finalmente, comportamientos como saludos, lenguaje corporal, desplazamientos y rutinas de entretenimiento son replicados por usuarios sin entrenamiento mediante la interfaz desarrollada.

Palabras clave Robótica, Pepper, Naoqi, Python, interfaz, control y gestión, ROS

Abstract

Social robots are intended to be used for people with no special digital skills. However, some social robots such as Pepper require programming of specific software environments including Choreographe and Naoqi, making some typical tasks like shifting and communication difficult. We propose here the development of an APK and multiplatform programs (on Linux, Android & Windows) for the interface of control on real time of Pepper robot by means of an integration of Naoqi, ROS, Linux and Windows's programs, and Android software layers. To achieve this,

movement and speech functions of Naoqi API are packet on nodes and services of ROS; these ones will publish and retrieve data on Android APK through web service. Common scenarios for the robot, including conversational and choreographic exercises, are constructed to validate correctness of the software programmed, ROS architecture setup and the characteristics of robot's responsibility. Finally, replication of built - in behaviors like greetings, body language, shifting and entertainment routines is performed by untrained users through the developed interface.

Keywords Robot, Pepper, Naoqi, Python, interface, control and management, ROS

Campos de acción

Con la presente propuesta, el Semillero de Robótica SERO, se enmarca en el campo de acción Sociedad [1] dando cabida a la investigación y a la proyección social pertinentes. De igual manera y de acuerdo al Plan Nacional de Desarrollo 2018-2022, se le apuesta a aportar en las áreas de: (2) Ciencias y tecnologías para el desarrollo sostenible y la biodiversidad protegida y (5) Educación, estrategias pedagógicas y didácticas, gracias al uso de de la inteligencia artificial y la robótica, permitiendo integrar estudiantes de la Facultad de Ingeniería Electrónica que propongan soluciones y hagan más accesible la tecnología al público en general.

Planteamiento del problema

El laboratorio de robótica de la Universidad Santo Tomás dispone del robot Pepper para realizar ejercicios de robótica social en diferentes contextos [2], los cuales abarcan actividades de investigación, educación continua, promoción y ceremonias institucionales como se aprecia en la Figura 1. Dependiendo del escenario y los objetivos específicos que se persiguen con la participación de Pepper en el mismo, el nivel de sofisticación de los programas desarrollados para el robot varían notablemente, haciendo que la programación requerida deba ser codificada desde cero en la gran mayoría de los casos. Si bien existen entornos de desarrollo para la programación de robots con diferentes alcances y potencialidades, que van desde entornos livianos hasta complejos sistemas de programación gráfica, para el caso específico del robot Pepper sólo se cuenta con el entorno gráfico Choreographe [3] y el conjunto de funciones C/Python integradas en la API NAOqi del robot [4]. Todo lo anterior trae como consecuencia que la programación de Pepper para cada una de las diferentes actividades en las que es requerido se traduzca en un esfuerzo no menor de programación y ajustes, así como en una pérdida significativa de calidad en las experiencias sociales del robot.

Entre los principales inconvenientes técnicos que la situación descrita anteriormente conlleva se encuentran la imposibilidad de control y gestión en tiempo real del robot, las continuas interrupciones y bloqueos en el computador de programación debidas a problemas de conectividad, y la alta cantidad de recursos de computación requeridos para la tarea. Este panorama afecta directamente cualquier tipo de preparación existente haciendo que en muchos casos sea preferible la utilización de las rutinas preprogramadas, las cuales son

genéricas y recomendadas solo para una interacción básica inicial con el robot. Otros problemas ligados a lo ya mencionado incluyen la necesidad de repetir expresiones, órdenes y preguntas al robot, limitaciones en la posibilidad de desplazamientos y desconexiones del mismo con sus medios de control con las personas.

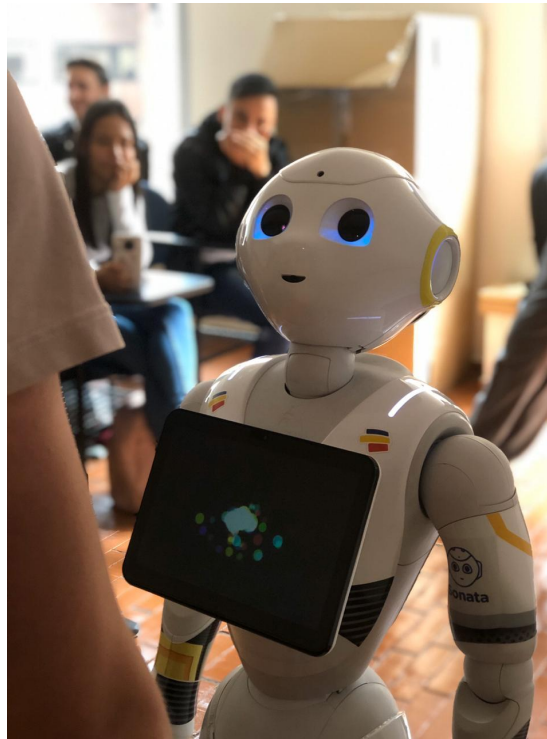


Figura 1. El robot social Pepper durante una actividad de Promoción

Dado el amplio uso que tiene Pepper alrededor del mundo, se han propuesto diversas soluciones como APKs privadas y la instalación de diversas distribuciones Linux sobre el robot. Destaca entre las soluciones abordadas en la Universidad Santo Tomás el desarrollo de una interfaz ROS para el control del robot como producto de la Alianza Sinfonía [5],[6], la cual persigue la inclusión del mismo en un ambiente de funciones distribuido; esto ha demostrado no ser una solución realmente factible y estable al problema dado que requiere de la utilización de otras herramientas de software como Docker (aumentando significativamente los recursos de cómputo necesarios), la baja e inadecuada utilización de funciones de ROS y la necesidad de control del robot de por lo menos dos operadores.

Justificación

Niños, personas de la tercera edad y público en general conforman el conjunto de usuarios comunes del robot Pepper, el cual debido a su aspecto, fluidez de movimientos e integración con algoritmos de inteligencia artificial ofrece una experiencia de interacción tecnológica altamente apreciada [7] [8]. La accesibilidad a la robótica tiene dos componentes

fundamentales: la facilidad de uso de los usuarios finales (personas sin cualificaciones técnicas especiales) y la facilidad de programación para los ingenieros de desarrollo (quienes definen la operación del robot). Estos últimos deben poder contar con instrucciones intuitivas y flexibles que les permitan obtener el mayor rendimiento posible de cada robot, por lo que una interfaz multiplataforma para el control del robot potenciará sus resultados.

El proyecto también recoge el conocimiento y la experiencia al interior del Grupo de Estudio y Desarrollo en Robótica - GED sobre los diferentes problemas en el trabajo con Pepper, permitiendo la identificación de soluciones basadas en software así como el establecimiento de los requerimientos técnicos de la misma. Con el proyecto, este conocimiento y experiencia pueden ser aplicados para de esta forma y a través de la interfaz de control que se propone realizar, generar rutinas de interacción con Pepper más dinámicas, flexibles y agradables para los usuarios finales.

En adición a lo anterior, el desarrollo de competencias y habilidades STEM en jóvenes se ve particularmente impactado por la robótica, disciplina de la ingeniería en la cual los 4 campos STEM (ciencia, tecnología, ingeniería y matemáticas) se ven profundamente involucrados. Es así como tareas típicas de la robótica como el desplazamiento y la comunicación se basan en el reconocimiento de fenómenos físicos, el consiguiente modelado matemático de los mismos, el diseño conceptual de ingeniería del sistema y la implementación para el despliegue en la plataforma tecnológica [9].

Al ser desarrollado este proyecto por parte de estudiantes del semillero en robótica SERO, se pretende fortalecer las habilidades técnicas y contribuir al desarrollo de competencias necesarias hoy en día en la adopción de tecnologías, tal como lo indica el Foro Mundial Económico en su reporte de 2020 sobre los trabajos del futuro: la inteligencia artificial, específicamente Machine Learning, redes neuronales, NLP y la robótica humanoide y no humanoide serán unas de las tecnologías que más de adoptarán en el 2025 [10]. Este tipo de proyecto, permite el desarrollo de habilidades que aportan en el aspecto profesional a todos aquellos estudiantes involucrados, además que permite el fortalecimiento de competencias en áreas STEM y la profundización en el conocimiento de herramientas específicas de la robótica como ROS.

Objetivo general

Desarrollar una interfaz multiplataforma (Android, Linux y Windows) para el control y gestión en tiempo real de las funciones del robot Pepper usando la API NAOqi y la estructura de nodos y servicios de ROS.

Objetivos específicos

1. Caracterizar las funciones incluidas en la API NAOqi específicas para la plataforma robótica Pepper mediante el estudio de la documentación técnica del fabricante, de repositorios accesibles en Github/Gitlab y de pruebas de laboratorio.

2. Diseñar la arquitectura de comunicación entre la plataforma Pepper y los dispositivos de control mediante la definición de *web services* de red local de datos.
3. Implementar los paquetes, nodos, mensajes y servicios de ROS que permitan la comunicación entre las aplicaciones de Naoqi y las interfaces multiplataforma empleando la topología cliente - servidor.
4. Integrar Naoqi, ROS y las aplicaciones multiplataforma para el control y gestión en tiempo real de las funciones del robot Pepper.
5. Validar la implementación mediante la réplica en tiempo real de comportamientos humanoides predefinidos en la plataforma Pepper

Metodología

El desarrollo del proyecto se dividirá en 5 fases, asociando a cada una de ellas un serie de actividades que van a permitir alcanzar los objetivos definidos. Se seguirá una metodología empírica y cuantitativa como se enuncia a continuación:

- **Fase de reconocimiento:** la primera fase será de exploración de la API NAOqi de la plataforma Pepper que ofrece el fabricante Softbank, por parte de los integrantes de la propuesta. Se usarán las diferentes bases de datos de las cuales dispone la Universidad (IEEE, SpringerLink, ScienceDirect, Taylor&Francis) con el fin de tener puntos de referencia en cuanto a las aplicaciones e indicaciones de uso de la plataforma robótica en escenarios sociales y domésticos. También se consultarán repositorios asociados a navegación, procesamiento de lenguaje natural, poses, identificación y reconocimiento de personas y objetos, entre otros. Se busca conocer la estructura de cada uno de los módulos de programación del *framework* NAOqi.
- **Fase de comunicación:** tras la etapa de reconocimiento, es necesario lograr interactuar y acceder desde diferentes dispositivos con la plataforma robótica, por lo tanto se realizará la implementación de una red local exclusiva para el robot. Se realizarán pruebas con diferentes dispositivos como routers, computadores, celulares, entre otros. Esto ayudará a garantizar la estabilidad de la comunicación con el robot. De igual manera se harán pruebas multiplataforma para el acceso al robot.
- **Fase de implementación ROS:** con el objetivo de gestionar los módulos de la API Naoqi así como los desarrollados para realizar la comunicación desde y hacia la plataforma, se usará la arquitectura de ROS definida por paquetes, nodos, mensajes y servicios, los cuales serán programados en python y C++. Para esta fase se evaluará la posibilidad de usar ROS1 o ROS2 a partir de pruebas de programación.
- **Fase de Integración:** en esta fase se realizará la integración de los módulos del *framework* NAOqi bajo la arquitectura de ROS sobre la interfaz multiplataforma. Se llevarán a cabo varias pruebas de la integración proponiendo diferentes escenarios

dentro del Laboratorio de Robótica y de acuerdo a los comportamientos predefinidos en Pepper.

- **Fase de validación:** la última etapa está dedicada a los ajustes necesarios sobre la interfaz multiplataforma partiendo de varios experimentos propuestos, los cuales están asociados con el escenario del laboratorio de robótica y los comportamientos predefinidos del robot Pepper. Se busca evaluar el desempeño de la integración de acuerdo al control y la gestión de la plataforma.

Resultados esperados

Con el desarrollo del presente proyecto se busca desarrollar una interfaz multiplataforma que permita la programación del robot Pepper de una manera intuitiva, eliminando los problemas de conectividad que se presentan actualmente con otros programas comerciales como Choregraphe. De igual manera se busca fortalecer las competencias en programación en estudiantes del programa de ingeniería electrónica.

Dentro de los productos comprometidos se presentan los siguientes, los cuales se relacionan en la Tabla 1:

Tabla 1. Productos comprometidos propuesta Semillero SERO 2023

Cantidad	Producto	Tipo
1	Generación de nuevo conocimiento	Artículo Tipo D
2	Productos de Apropiación social del conocimiento	- Participación en evento científico - Producción de contenido digital audiovisual, sonoro o recursos gráficos

Referencias Bibliográficas

- [1] O.-L. Ostos-Ortíz. 'Campos de acción USTA'. Universidad Santo Tomás. Working Paper No 197529, 2020.
<https://repository.usta.edu.co/bitstream/handle/11634/27354/Campos%20de%20acci%C3%B3n%20USTA.pdf?sequence=1&isAllowed=y>
- [2] A. K. Pandey, R. Gelin, και A. Robot, 'Pepper: The first machine of its kind', IEEE Robotics & Automation Magazine, τ. 25, τχ. 3, σσ. 40–48, 2018.

- [3] A. Subedi, D. Pandey, and D. Mishra, “Programming Nao as an Educational Agent: A Comparison Between Choregraphe and Python SDK,” in The Proceedings of the International Conference on Smart City Applications, 2021, pp. 367–377.
- [4] C. Yata, T. Ohtani, and M. Isobe, “Conceptual framework of STEM based on Japanese subject principles,” International Journal of STEM Education, vol. 7, no. 1, pp. 1–10, 2020
- [5] Repositorio Github Alianza SinfonIA.
<https://github.com/carlosquinterop/SinfoniaPepperTeam>
- [6] C. A. Quintero, E. Rojas, and F. Pérez, “2019 SinfonIA Pepper Team Description Paper,” 2019. <https://repository.usta.edu.co/handle/11634/2358>
- [7] M. Çetin και H. Ö. Demircan, ‘Empowering technology and engineering for STEM education through programming robots: A systematic literature review’, Early Child Development and Care, τ. 190, τχ. 9, σσ. 1323–1335, 2020.
- [8] E. C. Camacho, C. Higuera, J. G. Guarnizo, Y. Suarez, N. Garzon, και N. Bonifaz, ‘STOx’s Team description paper 2018’. <https://repository.usta.edu.co/handle/11634/22364>
- [9] S. Robotics, “NAOqi-Developer guide,” Línea]. Disponible en internet: http://doc.aldebaran.com/2-8/index_dev_guide.html, 2020
- [10] W. E. Forum, “The future of jobs report 2020,” Retrieved from Geneva, 2020.