

Slicer ROS 2, Arvind (Johns Hopkins)

La propuesta de Arvind conecta 3D Slicer con ROS 2 de forma directa y práctica para investigación en robótica médica. La idea central es tratar a Slicer como un nodo de ROS completo y, al mismo tiempo, exponer dentro de Slicer las capacidades de ROS como objetos de su lenguaje de modelado (MRML). Así, publicadores, suscriptores, parámetros y servicios quedan disponibles como si fueran nativos, con una integración que corre en el loop de eventos de Slicer a 50 Hz sin necesidad de hilos adicionales. La detección de tipos de mensajes se hace dinámicamente y se ofrece una consola de Python desde Slicer con tipos correspondientes a los de ROS, lo que reduce considerablemente la fricción para prototipar.

El valor aparece con los casos de uso. Cargar el URDF del robot se vuelve un clic aprovechando el robot description de robot state publisher. La visualización del robot coexiste con las imágenes médicas, lo que facilita planificar y probar procedimientos guiados por imagen en un mismo entorno. En la charla se mostraron dos ejemplos concretos: la planificación de una resección de tumor de lengua que controla un robot físico, y un escenario de cirugía de mama en el que se define una región prohibida alrededor del tumor. Si el instrumento entra en esa región, Slicer detecta la intrusión, dispara un evento en MRML y se envía retroalimentación háptica al cirujano; ese encadenamiento se resolvió con un script de Python breve. La integración con TF, el manejo de parámetros, la compatibilidad con mensajes personalizados y el uso de la infraestructura de Slicer para persistencia y callbacks completan una propuesta enfocada en ahorrar tiempo de plomería y concentrarse en el experimento clínico.

También hubo énfasis en la arquitectura: Slicer se comporta como un nodo ROS con publishers y subscribers reales; MRML organiza la información con observadores y nombres estables, de modo que la lógica reactiva es natural. La frecuencia por defecto de 50 Hz es suficiente para muchos flujos de navegación y visualización; es ajustable si se necesita más o menos carga. Aunque el énfasis estuvo en cirugía guiada por imagen, la misma estructura habilita prototipos de localización de instrumentos, sincronización con transformaciones de TF y visualización de trayectorias directamente sobre volúmenes médicos. La conclusión fue pragmática: integrar Slicer y ROS 2 de forma limpia permite pasar de la idea al prototipo clínico en días, no en meses.

Desafíos del control de posición en robots humanoides simulados, Gabriel Díaz (Ecumen, PUJ)

Gabriel puso el foco en la fragilidad inherente de los bípedos: mantener el equilibrio exige coordinación a nivel de cuerpo completo, no un conjunto de articulaciones obedeciendo cada una a su PID en aislamiento. Comparó dos enfoques en Gazebo. El

primero fue el más directo: plugins por articulación con control PID simple y límites locales. Sin una capa que vea el sistema completo y sin sincronización estricta con la física, el robot se desploma; los errores se amplifican porque cada joint intenta corregirse sin conocer lo que hacen los demás. El segundo enfoque se apoyó en ros2 control, que actúa como una capa jerárquica: respeta límites de esfuerzo y dinámica, se sincroniza con el motor físico mediante el bridge con Gazebo y coordina los modos de control (posición, velocidad, esfuerzo) de manera consistente.

En demostración, el esquema con ros2 control mantuvo al robot estable y receptivo. Un gesto deliberadamente visible, movimientos de cabeza, servía para evidenciar el seguimiento de referencias y la estabilidad del lazo. La moraleja fue directa: en humanoides, la estabilidad es el emergente de una arquitectura de control integrada; si cada articulación hace lo suyo, el sistema colapsa. Adoptar ros2 control y una jerarquía que converse con la física no es un detalle de implementación, es la condición para que el robot se sostenga y responda. Esto también implica que las pruebas en simulación deben respetar límites reales de torque y velocidad, y que los controladores se diseñan con una noción explícita del acoplamiento entre articulaciones.

ROS Blocks: robótica con bloques, Juan Andrés Ramírez (PUJ)

Juan Andrés abordó la barrera de entrada a ROS, especialmente para estudiantes que llegan sin experiencia en Linux, Python o la estructura de workspaces. Su propuesta toma la filosofía de Scratch y la lleva a ROS 2 con una plataforma web que permite construir nodos con bloques, usar publicadores, suscriptores, servicios y crear mensajes personalizados. La experiencia se centra en el navegador, con un backend en AWS que levanta un contenedor por usuario y expone un simulador básico con TurtleSim, laberintos con colisiones y utilidades para depuración. La plataforma exporta el workspace como archivo comprimido listo para compilar y ejecutar en local, de modo que lo creado con bloques se convierte en paquetes ROS 2 reales.

En pruebas con estudiantes, un 77 por ciento encontró la herramienta más intuitiva a nivel introductorio. Aunque el dominio original dejó de estar activo, el proyecto quedó abierto en GitHub y puede ejecutarse localmente con Docker y Node.js. Los pasos siguientes incluirían integrar visualizadores y simuladores como RViz y Gazebo, gestionar sesiones de usuarios en la nube y conectar robots reales. El hilo conductor es claro: reducir fricción inicial sin renunciar al estándar de ROS 2.

Exápodo, puesta en marcha, Andrés Camilo Torres Cajamarca y Felipe Chávez Delgadillo (Universidad Nacional)

El trabajo sobre el exápodo inició con el modelado de una sola pata, empleando parámetros de Denavit Hartenberg y trayectorias parabólicas para el paso, buscando contactos suaves y controlables. La validación dinámica se realizó en Simulink con Simscape Multibody, importando los modelos CAD, y se integró con ROS 2 para comandar articulaciones y observar el comportamiento global.

La arquitectura de control se dividió en nodos de propósito general, interacción con el usuario, cinemática inversa, coordinación, y nodos específicos para simulación y robot físico. El cierre de lazo incluye la lectura de posiciones articulares y la actualización del coordinador para avanzar sobre la trayectoria planificada. Durante las pruebas, el robot mostró caminatas funcionales, pero la velocidad disminuía con el tiempo por latencias y carga computacional en la Raspberry Pi, exacerbadas por nodos en Python. La conclusión práctica: el diseño de la arquitectura y la distribución del cómputo son tan decisivos como el modelado del paso.

Exápodo, autonomía y visión, Juan Camilo Gómez, Andrés Torres Cajamarca, Emilia Angélica Villanueva Cerna, Julián Andrés González Reina (UNAL)

Sobre la base de la puesta en marcha, el equipo introdujo percepción y control distribuido para dar autonomía al exápodo en un entorno estructurado. Con 18 articulaciones Dynamixel, reorganizar la alimentación y limitar corrientes fue un requisito para mantener marcha consistente. La visión por cámara se transmitió por RTSP a un computador principal, que ejecuta los nodos de percepción y planificación; la Raspberry Pi mantiene la ejecución de bajo nivel, motores y gripper, conectada por Wi-Fi.

La detección por color entrega centroides de objetos y del propio robot, con una flecha para estimar orientación, y el control aplica una máquina de estados que alterna giros y avances hasta posicionarse para el agarre. Se optó por acciones en lugar de servicios para poder interrumpir o ajustar objetivos en tiempo real conforme cambia la percepción. El gripper, construido en TPU con superficies de silicona, aporta seguridad y adaptabilidad. En demostración, el robot selecciona objetos por color y ejecuta la tarea de clasificación; la principal limitación sigue siendo la potencia, dieciocho motores demandan corrientes elevadas, lo que condiciona la autonomía.

Pushing the Boundaries of Robotics Automation with Robots, John Alberto Betancur

John trazó una línea que va de proyectos estudiantiles a liderar un departamento de autonomía en Silicon Valley. La empresa, de raíces latinoamericanas y antes conocida por Kiwibot, opera robots móviles para logística y atención al público, con despliegue en múltiples regiones y una infraestructura que permite administrar flotas desde un centro de operaciones. Presentó sus plataformas móviles de reparto, un humanoide en desarrollo y un robot tipo perro diseñado para interacción y seguridad.

La parte técnica combina percepción redundante, mapeo con extracción semántica, navegación en interiores y exteriores y un ecosistema de software que soporta casos de uso diversos. La reflexión final subrayó que la robótica hecha desde Latinoamérica es posible si se apuesta por el aprendizaje continuo, el pensamiento crítico y la toma de riesgos.

Control de navegación agrícola con Pure Pursuit, de ROS Melodic a ROS 2 Humble, Edna Carolina Moriones (UTS)

Edna situó el problema en un contexto global de robots agrícolas y, a la vez, en la realidad colombiana de pendientes pronunciadas y alta demanda energética. La trayectoria del proyecto pasó de ROS Melodic en Raspberry Pi a ROS 2 Humble, migrando hacia plataformas Jetson para percepción y delegando control bajo en microcontroladores con micro ROS. El vehículo integra controladores BLDC con PIDs internos y un juego de nodos que fusionan IMU y GPS.

El controlador de seguimiento de trayectorias se basó en Pure Pursuit, parametrizado con ganancias y un rango de lookahead ajustado a velocidades objetivo. En lugar de seguir curvas suaves idealizadas, el sistema consume puntos sucesivos, incorporando el error que realmente ocurre en campo. En simulaciones sobre topografías reales, el desempeño mostró tasas de éxito elevadas en recorridos cortos y retos mayores en trayectorias más largas o circulares. El aprendizaje central fue pragmático: Pure Pursuit funciona bien para navegación básica; cuando se acoplan otras tareas, dosificación, manipulación, un MPC cobra sentido. Como próximos pasos, más validación en terreno y uso de gemelos digitales.

SEFIR, plataforma experimental de baja fricción para robótica holonómica, Santiago Romero, Luis Enrique Coquier, Carlos Eduardo Cortés (PUJ)

SEFIR busca emular la microgravedad en el plano mediante una falda de aire que separa el robot del suelo apenas un milímetro, reduciendo la fricción a niveles mínimos. El empuje se logra con válvulas neumáticas que dirigen chorros de aire en traslación y rotación, y el control de posición se implementó con ROS 2 usando un controlador PD que modula la apertura de válvulas por PWM.

Para posicionamiento preciso, el sistema se apoya en captura de movimiento con cámaras Vicon y marcadores reflectivos; como alternativa de menor costo, combinan LiDAR e IMU con fusión por filtrado. La simulación en Gazebo permitió modelar la dinámica de impulsos de las válvulas y cerrar el ciclo de diseño antes de pasar al laboratorio. En pruebas, el robot realiza movimientos de paso y mantiene errores contenidos en orientación y posición. El plan a corto plazo apunta a independencia del suministro de aire, mayor control en ejes ortogonales y navegación con SLAM.

Orión en Acción, plataforma abierta para interacción humano robot, Daniel Felipe López (PUJ, Triple RAS Javeriana)

Orión es un robot social diseñado para ser reproducible: estructura modular impresa, electrónica accesible y un stack ROS 2 que combina navegación, SLAM, control de actuadores y módulos de interacción. La arquitectura separa control de motores en microcontroladores, percepción con sensores de bajo costo y procesamiento de voz con pipelines que usan reconocimiento, síntesis y modelos de lenguaje ligeros en local o más potentes en un equipo externo.

El desarrollo avanzó en iteraciones rápidas, primero en simulación, luego en hardware, con la intención de que estudiantes y laboratorios puedan replicar el proyecto. En demostraciones, el robot cartografía espacios interiores, mantiene diálogos sencillos y acompasa gestos a estados emocionales detectados por visión. La contribución no es solo el robot, sino el método y los recursos abiertos que facilitan su adopción.

Sistemas multiplataforma de transcripción y respuesta en tiempo real con modelos de lenguaje para ROS, David Cuevas (Sinfonía Uniandes)

Este trabajo integra un pipeline de voz en tiempo real con un agente basado en modelos de lenguaje para orquestar comportamientos en robots como Pepper o NAO. La transcripción continua se resuelve con detección de actividad de voz y modelos de reconocimiento robustos; el agente interpreta intenciones, invoca acciones disponibles, cambiar color de ojos, ejecutar gestos, hablar, y coordina secuencias compuestas, devolviendo también un resumen de lo realizado. La arquitectura es portable: puede usar un modelo en la nube o variantes locales y expone el control a través de ROS 2 para integrarse con el resto del sistema. La fortaleza está en la naturalidad de la interacción y la capacidad de recomponer acciones a partir de instrucciones abiertas.

Rosline, asistente de WhatsApp para robótica, Miguel Ángel González (RAS Colombia, Ecumen)

Rosline permite controlar robots ROS 2 y consultar el estado del sistema desde WhatsApp con lenguaje natural. El backend interpreta mensajes, descubre tópicos y servicios activos y publica comandos cuando corresponde. En la demostración, se gestionaron múltiples robots por nombre y se respondieron consultas sobre la red ROS. La elección de WhatsApp elimina fricción del lado del usuario y abre posibilidades de teleoperación y soporte remoto, siempre que se agreguen salvaguardas de seguridad, whitelists y control de latencia. El proyecto usa un modelo de lenguaje económico y herramientas de orquestación para mantener el sistema liviano.

Del genoma al robot, lenguajes compartidos de sistemas vivos y artificiales, Laura Zuluaga (Uniandes)

Laura propuso mirar la robótica desde la vida misma: hongos que resuelven redes óptimas, enjambres que cooperan, tejidos que se regeneran, exoesqueletos que devuelven movimiento. El mensaje se mueve entre inspiración y responsabilidad. La bioinspiración no es una moda, sino un atajo hacia soluciones que se adaptan, que ahorran energía y que conviven con el entorno. Pero también exige ética: comprender implicaciones, anticipar usos y cuidar el para qué de lo que diseñamos. Fue una invitación a rehacer el vínculo entre técnica y propósito.

Políticas de aprendizaje por refuerzo para cuadrúpedos con Isaac Sim e Isaac Lab, Lugo

La charla conectó el aprendizaje por refuerzo con simulación de alta fidelidad para entrenar cuadrúpedos que se estabilizan y se desplazan robustamente. Con entornos masivos en paralelo, perturbaciones durante el entrenamiento y aleatorización del dominio, las políticas aprenden a responder a superficies y empujes sin necesidad de un modelo exacto del sistema. Isaac Sim y su formato USD permiten escenas físicas detalladas, fricción realista, iluminación y generación de datos sintéticos, lo que acelera la transferencia a robots reales. La tendencia que asoma es la hibridación: políticas guiadas por visión y modelos del mundo que encapsulan tokens de movimiento, capaces de generalizar más allá de los escenarios vistos. La idea no es desplazar el control clásico, sino reservarlo para donde aporta, y dejar que el aprendizaje absorba la complejidad que antes hacía inviable el modelado.