**UNIVERSIDAD NACIONAL DE INGENIERÍA**

**FACULTAD DE ELECTROTECNIA Y COMPUTACIÓN**

ANEXO 2

ABC DEL SISTEMA OPERATIVO PARA ROBOTS(ROS)

Elaborado por: Br. Yeser Alfredo Morales Calero

Tutor: MSc. Alejandro Alberto Méndez Talavera

26/08/ 2019

**Índice de Contenido\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

1. INTRODUCCIÓN[…………………………………………………….……….……….12](#Objetivos)
   1. Porque ROS[……….....................…………………………….…….…..…….12](#ObjetivoGeneral)
   2. Que se obtiene al utilizar ROS[………….……………………..….….………12](#ObjetivosEsp)
2. [MARCO TEÓRICO……………………………………..……….……………...…….13](#MarcoTeor)
   1. [Tipos de Laboratorios……………………………………...…………….…..….13](#TiposDeLAB)
      1. [Laboratorios Físico…………………..…………………………….…..….13](#LABFisico)
      2. [Laboratorios Virtual………………………………….…………….…..….14](#LABVirtual)
      3. [Laboratorio Remoto online………………………………….…….…..….14](#LABRemoto)
      4. [Realidad Virtual 3D………………………………………….…….…..….14](#LABVirtual3D)
   2. [Modelo de utilización del laboratorio virtual para robótica industrial………16](#ModeloLAB)
   3. [Sistema operativo para robot (ROS)……………….…………………………..17](#ROS)
      1. [Conceptos básicos de ROS…………………………...…………………18](#ConceptosROS)
      2. [Lenguajes de programación soportados…………………………….….20](#LanguageC)
      3. [Aplicaciones y librerías……………………………………………….…..21](#AppROS)
      4. [Herramientas de ROS…………………………………………………....21](#HerramROS)
   4. [Interfaz gráfica de usuario (GUI) ……………………………………………...21](#GUI)
      1. [Interacción dentro la GUI…………………………………………………22](#GUIInteraccion)
      2. [Visualización de datos……………………………………………………22](#GUIVisualizacion)
         1. [Visualización 3D de robots……………………………………….22](#GUIVisualizacion3D)
         2. [Gráficos de dispersión XY………………………………………..22](#GUIVisualizacionXY)
      3. [Programación de Rutinas de robots…………………………………….23](#ProgrammingRobots)
      4. [Información de los fundamentos de Robótica Industrial………………23](#InfoGUI)
   5. [Modelos de robots en el LVR…………………………………………………...24](#ModelosLVR)
      1. [Modelado de brazos robóticos de diferentes DOF……………..…..…24](#ModelosbrazosDOF)
   6. [Entornos de simulación de robots……......…..........................………………25](#EntornosSimu)
      1. [Software de compañías basadas en CAD ………………….………….26](#EntornosCAD)
      2. [Software proveído por los fabricantes de robots…………….….……..26](#EntornosFab)
      3. [Softwares de programación de varios fabricantes…………………….26](#EntornosvariousFab)
      4. [Software Simuladores de Robots de propósito general………………27](#EntornosGNRL)
         1. [Simuladores de propósito general comerciales………………..27](#EntornosGNRLComer)
         2. [Simuladores de propósito general de uso libre………………...28](#EntornosGNRLOpen)
   7. [Modelación Cinemática y Dinámica…….…………………………………......29](#KinemDyn)
      1. [Cinemática Directa e Inversa ………………………...………..………..29](#FKIK)
      2. [Espacio de trabajo del robot……………………………………………..29](#Workspace)
   8. [Control de movimiento en el entorno…………………………….…………….30](#CTRLMOV)
      1. [PID loop……………………………...……………………….……………30](#PIDLoop)
   9. [Hardware externo………………………………………………..…………….…30](#HARDext)
      1. [Joystick control…………………………………………………………....31](#Joystick)
   10. [Manual de uso del laboratorio virtual……………………….…………..31](#Manual)
3. [HIPÓTESIS Y VARIABLES……………………………………….…………………32](#Hipot)
4. [DISEÑO METODOLÓGICO…………………………………………………………33](#DesignMet)
5. [CRONOGRAMA DE EJECUCIÓN………………………………….………………36](#Cronograma)
6. [BIBLIOGRAFÍA…………………………………………………………..……………37](#Biblio)

[ANEXOS……………………………………………………………………..…………….42](#anexos)

# **Introducción**

La comunidad de robótica ha hecho un progreso impresionante en los últimos años, con el desarrollo de robots móviles, drones, humanoides. Aún más impresionante, la comunidad también ha desarrollado algoritmos que ayudan a esos robots a funcionar con niveles cada vez mayores de autonomía. Debido a este rápido progreso, los robots aún presentan algunos desafíos significativos para los desarrolladores de software. ROS, fue desarrollado por la comunidad para superar algunas de estas dificultades. La descripción oficial de ROS es de un sistema meta-operativo de código abierto para su robot. Proporciona los servicios que se esperaría de un sistema operativo, incluida la abstracción de hardware, el control de dispositivos de bajo nivel, transmisión de mensajes entre procesos y gestión de paquetes.

La anterior descripción es precisa, y enfatiza correctamente que ROS no reemplaza, sino que funciona junto con un sistema operativo tradicional, pero puede hacer que se pregunte cuáles son las ventajas reales para el software que usa ROS. A continuación se enumera algunos problemas específicos en el desarrollo de software para robots que ROS puede ayudar a resolver

Los diferentes temas presentado en este documentados estarán basados en mi corta experiencia de desarrollo de ROS con c++, se le recomienda al interesado a revisar la información con Python

# **Objetivo:**

En este Documento a la persona interesada se le facilitara la información y respuestas fundamentales del middleware ROS, esto con el fin de motivar a los estudiante de tecnologías de la información y electrónica de la Universidad Nacional de Ingeniería el uso de ROS al realizar proyectos.

Dos nodos como aplicación del LVR (Algoritmo).

Nacimiento y origen de ROS

**Requisitos**

Microcontrollers

Microsoft

Explicando Conceptos de ROS

Estructura, Arquitectura.

Desarrollo

1. Desarrollo de nodos básicos.
2. Entorno de trabajo (Puesta a punto para iniciar un proyecto)
   1. Proceso de instalación de ROS requerimientos.
   2. Instalación de IDE
   3. Instalación de Gazebo.
3. Ejemplos de aplicación de ROS
   1. Creando tu primer proyecto modelos
      1. Nodos
      2. URDF
      3. RVIZ
   2. Simuladores
   3. Librerías
   4. Plugins

**Introducción**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

La robótica es uno de los campos de la tecnología que (evoluciona) se desarrolla velozmente y encuentra nuevas áreas de aplicación