

TRABAJO PRACTICO INTEGRADOR: Python y C++**A. Elementos necesarios**

Guías de clase. Prácticas con lenguajes Python y C++.

Anexo de “Referencias Básicas”.

B. Entrega y Evaluación

La fecha de entrega del informe y de la presentación grupal habitualmente se indica al principio del ciclo, pero puede ser ajustada por mutuo acuerdo en clase.

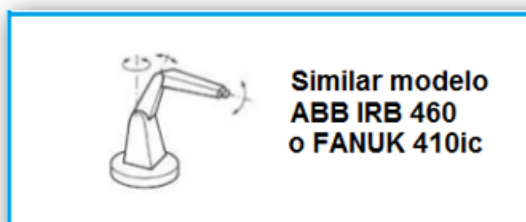
Esta presentación es de modalidad presencial de el/los autor/es del trabajo, con exposición pública y posterior coloquio, abordando aspectos teórico-prácticos sobre la implementación realizada.

C. Actividades Generales

1. A partir del enunciado de la consigna específica, de los temas especiales involucrados (interacción con teclado y/o mouse, threads, comunicaciones mediante protocolos y/o mecanismos de red bajo arquitectura cliente/servidor, gestión de archivos) y de los temas opcionales de su interés particular (interfaces gráficas de usuario, Web, manipulación de gráficos, sonido, animaciones, etc.), investigue sobre las posibilidades de los lenguajes Python y C++ para manejar los mismos.
2. Puede acordar con el profesor los detalles que no estén especificados con claridad en este enunciado.
3. Proponga e implemente una Solución Orientada a Objetos aplicando los conceptos fundamentales del ciclo: Clases, Objetos, Instanciación, Herencia, Agregación, Colecciones de Objetos, Polimorfismo, Streams de E/S, Constructores/Deconstructores, Sobrecarga de Métodos, Sobrecarga de Operadores, Control de errores mediante Manejo de Excepciones.
4. Realice un informe, siguiendo un estilo de paper similar a los ejemplos, donde:
 - a. documente el modelo OO del implementado (tanto del lado servidor como del lado cliente),
 - b. describa los detalles de instalación, configuración y ejecución particulares
 - c. detalle brevemente los conceptos teóricos usados relativos a los temas especiales u optativos incluidos (ítem 1),
 - d. Agregue los esquemas o imágenes que estime convenientes para acompañar sus descripciones.
5. Elabore una presentación breve (duración no mayor a 10 minutos) donde explique sintéticamente su solución, preferentemente basada en los gráficos y/o esquemas asociados tanto conceptuales como de la implementación realizada.
6. Elabore un video breve (duración no mayor a 5 minutos) donde muestre la funcionalidad y uso de su aplicación.
7. Realice la entrega de los archivos resultantes (código fuente del proyecto, incluyendo librerías de terceros usadas, ejecutable, informe y videos), en la actividad habilitada en el aula virtual (o envíe por correo al profesor si se indicara explícitamente).
8. Eventualmente, deberá realizar una ampliación o defensa grupal del trabajo realizado.

D. Consigna específica

1. Dominio del problema
 - Programación Orientada a Objetos para Manipulación Remota de un Robot.
2. Objetivo
 - Desarrollar un aplicativo para controlar un robot de 3 grados de libertad con efector final.
3. Robot específico



- Tipo: robot 3DF.
 - Configuración: RRR
 - Efector final: simple (pinza, minitorno o pistola de pintura a elección).
 - Dimensiones (extremas): anchura=00mm, altura=00mm, profundidad=00mm
 - Velocidades máximas: lineal=00mm/s, angular=0rad/s
4. Requerimientos generales
 - El aplicativo/sistema debe estar representado e implementado usando OO.
 - Debe diseñarse usando modelos de capas (al menos 2), de manera que la solución pueda usarse independientemente de que haya interfaz gráfica o de consola.
 - Los módulos del lado servidor deben estar desarrollados en Python
 - Los módulos del lado cliente deben estar desarrollados en C++.
 - El desarrollo debe ser operativo bajo plataforma Linux para los módulos del lado servidor¹. Para el lado cliente no es obligatorio, pero resulta preferente usar plataforma Linux².
 - Servicios del lado servidor:
 - Panel de control del robot, mediante consola (CLI) que permita realizar:
 - Conexión/desconexión al Robot usando comunicación serial.
 - Activación/Desactivación del robot.
 - Listado de los comandos de control
 - Ayuda con ejemplificación de la sintaxis requerida³ para los comandos que lo requieran.

¹ Se espera que dicho desarrollo pueda ser implantado en una SBC corriendo una distribución Linux/GNU.

² Existen diferencias de portabilidad/compatibilidad en caso de usar Windows como plataforma de desarrollo. Por ejemplo, si se usara Qt5 no hay problema, mientras que con protocolo RPC no hay garantías (si bien hay herramientas tipo rpcgen desde la librería de wingnu32).

³ Es viable asumir que el robot acepta órdenes del tipo G-code a través de su comunicación serial.

- [modo manual]: Movimiento, de cada uno de los vínculos (de manera independiente) especificando la velocidad, distancia o ángulo a recorrer y el sentido.
 - [modo manual]: Actividad del efector final (propias de la herramienta particular elegida) especificando velocidad, tiempo de operación y sentido de operación (si correspondieran).
 - [modo manual]: Movimiento a su “posición de origen”.
 - [modo automático]: Carga y ejecución de un archivo conteniendo una secuencia de trabajo específica.
 - Servidor XML-RPC, capaz de ejecutar las mismas operaciones anteriores solicitadas en forma remota.
 - La aplicación de consola debe implementar herencia desde la clase Cmd
- Servicios del lado Cliente. La interfaz principal debe ofrecer:
 - Panel de control del robot, con funcionalidades similares a las de la interfaz de consola.
 - Reporte mostrando la conexión, el estado de actividad del Robot, el instante de tiempo (acumulado) desde el inicio de actividad, el instante de inicio de cada orden, las coordenadas de cada articulación del robot y, de estar disponible, la velocidad del efector final.
 - [opcional] GUI de control y monitoreo de parámetros.
 - [opcional] Despliegue visual del robot, usando vista 3D o proyecciones. En cualquiera de los casos puede usarse una estructura simplificada del robot, ejes/planos y escala (a su criterio).
 - [opcional] Tabla o curvas que muestren la velocidad de cada uno de los vínculos.
 - [opcional] Aprendizaje de trayectoria con almacenamiento de las órdenes correspondientes en un archivo de texto (legible) que permita reproducir la misma.
 - [opcional] Streaming de video mostrando el comportamiento del Robot usando una cámara remota.
 - [opcional] Efectos de sonido que indiquen el arranque/parada del robot, los cambios de posición del efector final y los momentos de actividad del mismo.
 - [opcional] Generación de alarmas visuales/auditivas cuando se intenten desplazamientos fuera del espacio de trabajo (aproximado).
- El efector final no agrega grados de libertad especiales, más allá de algún desplazamiento fijo en las coordenadas asumidas para el punto de operación.
- El aplicativo debe operar de manera completa con control desacoplado (aunque resulta mejor valorada aquella propuesta que avance en el control acoplado).
- Definir constantes apropiadas para los parámetros extremos (límites de trabajo).
- Otros datos constructivos del robot o plataforma son indicados durante el desarrollo

E. Anexo: Esquema⁴ del Informe (ver ejemplos)

1. Carátula
 - a. Identificación de materia
 - b. Identificación del trabajo
 - c. Identificación de alumno/s
 - d. Identificación de docente/s
 - e. Identificación de carrera
 - f. Identificación de facultad/universidad
 - g. Identificación de ciclo lectivo
2. Introducción
 - a. Descripción breve del problema
 - b. Descripción detallada, con imágenes de robots del tipo asignado y ejemplos de implementación real si las hubiere.
 - c. Identificación (listado) de los Temas Especiales de Python/C++ investigados
3. Marco teórico (desarrollo de cada uno de los Temas Especiales)
 - a. Conceptos/fundamentos
 - b. Esquema o gráfica descriptiva con detalles de aspectos estructurales o funcionales
 - c. Descripción de componentes software (librerías, frameworks, aplicaciones, etc.) asociados específicamente al problema/solución.
 - d. Descripción/ejemplificación paso a paso del modo de instalación/vinculación/uso, aplicado a la solución específica
4. Desarrollo
 - a. Diagrama de clases de la solución propuesta (debe ser consistente con la implementación)
 - b. Diagrama de secuencia de la solución propuesta (debe ser consistente con la implementación)
 - c. Diagrama de actividad (a nivel de aplicación) de la solución propuesta (debe ser consistente con la implementación)
 - d. Screenshots mostrando la ejecución del aplicativo, para cada una de las interfaces de entrada/salida propuestas
5. Comentarios y Conclusiones
 - a. Comentarios sobre las dificultades enfrentadas y su resolución, justificaciones a la solución propuesta (de diseño, de implementación, de instalación, etc.)
 - b. Análisis de ventajas/desventajas de los componentes usadas
 - c. Posibles extensiones a la solución dada
6. Referencias
 - a. Bibliografía/Sitios web consultados
 - b. URLs a los recursos software (tutorial, video, ejemplo, herramienta, etc.) que se hayan usado.

⁴ El formato propuesto es un acercamiento al usado en tesinas de grado, posgrado y papers de investigación