**Universidad Peruana de Ciencias Aplicadas**

**UPC**

“Año del diálogo y la reconciliación nacional”



Curso de Diseño de Experimentos en Ingeniería de Software

Profesor:

Luis Martin Canaval Sanchez

Sección: CC71

Alumnos:

Mariano Alonso De Rivero Burbank 201510450

Sergio Sugahara Cruzalegui 201513410

Renzo Adriano Ravelli Altamirano 201616446

José María Ansset Rojas Guimarey 201623243

Monterrico, noviembre 2019

# Índice

Contenido

[Índice 2](#_Toc25265823)

[Objetivo del Estudiante 3](#_Toc25265824)

[Presentación 3](#_Toc25265825)

[Marco Teórico 3](#_Toc25265826)

[Implementación de la Solución 3](#_Toc25265827)

[Experimentación 6](#_Toc25265828)

[Conclusiones 6](#_Toc25265829)

[Recomendaciones 6](#_Toc25265830)

[Bibliografía 6](#_Toc25265831)

# Objetivo del Estudiante

Las competencias presentes en el curso son las siguientes:

* General: Ciudadanía (WASC Nivel 1)
* Específica: Aprendizaje Continuo y Autónomo (ABET Nivel 2)

# Presentación

En el presente documento se presentará el proyecto de final de curso, para el cual se desarrolló una aplicación mediante la cual se pueden generar programas en el lenguaje Melfa Basic-V para el brazo robótico RV-2SD Melfa de Mitsubishi en el CIM-Lab del campus Villa de la universidad. Esta aplicación puede ser controlada tanto de manera manual como por comandos simples de voz. Este permitirá una interacción más sencilla y rápida entre el usuario y el brazo robótico, puesto que no hace falta conocimiento previo respecto a las limitaciones del brazo o el lenguaje de programación que utiliza.

# Marco Teórico

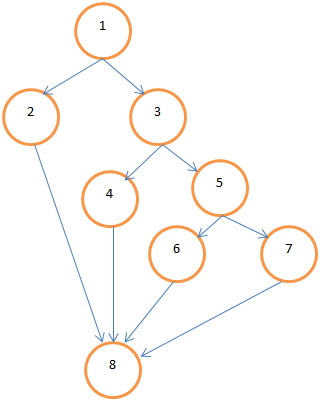
* Unity: Motor de videojuegos multiplataforma creado por Unity Technologies. Si bien el motor fue diseñado con la idea de apoyar el desarrollo de videojuegos, este puede ser utilizado para distintos tipos de aplicaciones como la presente.
* Autodesk 3DS Max: Programa de creación de gráficos y animaciones 3D programado en C++. La primera versión de la herramienta fue lanzada en 1990.
* Reconocimiento de Voz: El reconocimiento automático del habla o de voz, es una disciplina de la Inteligencia Artificial que tiene como objetivo permitir la comunicación hablada entre seres humanos y computadoras. Un sistema de reconocimiento de voz es una herramienta capaz de procesar y reconocer la señal de la voz y convertirla en texto o emitiendo órdenes que actúan sobre un proceso.
* Melfa Basic V: Lenguaje de programación usado para ciertos robots de marca Mitsubishi y paquetes de simulación.

# Implementación de la Solución

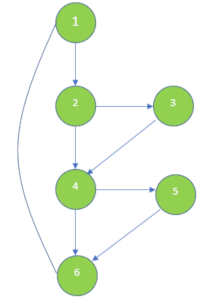
La solución fue implementada en el Motor de Videojuegos Unity haciendo uso del lenguaje de programación orientado a objetos C#. La aplicación cuenta con un modelo en 3D hecho en la herramienta 3DS Max, el cual está compuesto por distintas articulaciones tal como el modelo que se encuentra en el CIM-Lab. En la interfaz gráfica se encuentran los 6 ángulos que permiten las rotaciones de las distintas articulaciones del brazo, la representación gráfica del modelo y una caja de texto que contiene las distintas acciones que se han realizado al brazo. Durante el proceso de implementación del aplicativo se realizaron 4 pruebas, siendo 2 pruebas de caja blanca y 2 de caja negra.

Pruebas de Caja Blanca:

* La primera prueba de caja blanca se realizó al apartado de la inicialización de los datos de las articulaciones, dando como resultado 4 caminos únicos, los cuales representan cada uno de los ejes de rotación y el último siendo el caso predeterminado o por defecto.



* La segunda prueba de caja blanca se realizó para la exportación de código fuente, la cual dio como resultado 5 caminos únicos, los cuales son en base a la creación de los archivos y la sobre-escritura de los mismos o como último caso, que las cajas de texto no tienen contenido.



Pruebas de Caja Negra:

* La primera prueba de caja negra se realizó para las rotaciones de la base del brazo:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| N | Input (i) | Input (f) | Clase de Equivalencia valida | Clase equivalencia no valida | Propósito del Caso | Resultado Esperado |
| 1 | 30º | 250º | (-)240º <= ángulo <= 240º | Grados menores -240° y mayores a 240°. Strings. | Determinar si el ángulo de giro de la cintura que se recorrerá es físicamente posible para el brazo | Falla, el script no se ejecutara por violar las limitaciones del en el límite superior |
| 2 | (-)250º | 100º | (-)240º<= ángulo <= 240º | Grados menores -240° y mayores a 240°. Strings. | Determinar si el ángulo de giro de la cintura que se recorrerá es físicamente posible para el brazo | Error de input, el script no se ejecutara por violar la limitación en el input de primera posición |
| 3 | 14º | 199º | (-)240º<= ángulo <= 240º | Grados menores -240° y mayores a 240°. Strings. | Determinar si el ángulo de giro de la cintura que se recorrerá es físicamente posible para el brazo | El script se ejecutara correctamente |
| 4 | 240º | (-)240º | (-)240º<= ángulo <= 240º | Grados menores -240° y mayores a 240°. Strings. | Determinar si el ángulo de giro de la cintura que se recorrerá es físicamente posible para el brazo | El script se ejecutara correctamente |

* La segunda prueba de caja negra se realizó para las rotaciones del hombro del brazo:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| N | Input (i) | Input (f) | Clase de Equivalencia valida | Clase equivalencia no valida | Propósito del Caso | Resultado Esperado |
| 5 | 40º | 110º | (-)120º<= ángulo <= 120º | Grados menores -120° y mayores a 120°. Strings. | Determinar que se respeten los límites de movimiento de la articulación del hombro | El script se ejecutara correctamente |
| 6 | (-)250º | 100º | (-)120º<= ángulo <= 120º | Grados menores -120° y mayores a 120°. Strings. | Determinar previo a ejecución si el movimiento será físicamente posible de realizar por el brazo mecánico. | Error, no se ejecutara el movimiento por violación de límites de movimiento |
| 7 | 14º | 199º | (-)120º<= ángulo <= 120º | Grados menores -120° y mayores a 120°. Strings. | Determinar previo a ejecución si el movimiento será físicamente posible de realizar por el brazo mecánico. | Error, no se ejecutara el movimiento por violación de límites de movimiento |
| 8 | 240º | (-)240º | (-)120º<= ángulo <= 120º | Grados menores -120° y mayores a 120°. Strings. | Determinar previo a ejecución si el movimiento será físicamente posible de realizar por el brazo mecánico. | Error, no se ejecutara el movimiento por violación de límites de movimiento |

# Experimentación

Esta aplicación tiene como objetivo principal el facilitar el uso del brazo robótico mediante la posibilidad de realizar movimientos sin la necesidad de conocer las limitaciones del brazo puesto que el programa no permite ir por encima de estas, o del lenguaje, ya que los movimientos se llevan a cabo mediante los botones de la interfaz gráfica o por comandos de voz. Al realizar cualquier rotación de alguna de las articulaciones se actualiza la caja de texto de la articulación en la cual se especifica el ángulo actual de rotación y se observa también este cambio en la caja de texto de los movimientos realizados. Luego de mover el modelo hasta la posición deseada, se puede hacer click al botón de guardado para que esta posición sea guardada para luego ser exportada, después de hacer la sucesión de movimientos que se prefiera.

# Conclusiones

Luego de la culminación del trabajo se puede concluir que la creación de un modelo o simulación de la realidad, si bien debe asemejarse a la misma, lo ideal es que principalmente cumpla la funcionalidad principal de la herramienta u objeto. Al desarrollar una herramienta que facilite el uso de una más especializada, permite que esta sea ejecutada por personas de interés que no necesariamente estén capacitadas de manera técnica para hacer uso de ellas. A su vez, concluimos que las nuevas tecnologías como los comandos de voz Speech to Text aplicadas a proyectos de software ya existentes como es el caso del brazo mecánico modelo RV-2SD puede mejorar la eficiencia del flujo de trabajo al simplificar las tareas necesarias. Y para finalizar, el correcto desarrollo de un software usando las herramientas adecuadas, como las de control de versiones (en este caso GitHub) para verificar que las versiones de código cumplan con las historias de usuario y los requerimientos respectivos junto con una buena documentación que incluya elementos necesarios como las pruebas independientes, pruebas de caja blanca y caja negra, son esenciales para agilizar el proceso de desarrollo y tener una aplicación funcional que cumpla con lo que se necesita.

# Recomendaciones

En caso se fuera a realizar este experimento nuevamente, o dar uso al prototipo planteado, se recomienda conocer las limitaciones físicas reales del brazo, puesto a que se trabajó en base a las limitaciones que tiene, sin embargo, no se tomó en consideración la ubicación del mismo o el ambiente en el que se encuentra, lo que puede llegar a causar problemas o un mal funcionamiento del mismo. A su vez, sea el caso que se lleva a cabo nuevamente el desarrollo de una aplicación similar, se recomienda tomar en consideración las dimensiones reales del modelo, para que la simulación y los movimientos que se realicen sean lo más precisos y apegados a la realidad posible.