**UNIVERSIDAD DE TARAPACÁ**



**ESCUELA UNIVERSITARIA DE INGENIERÍA INDUSTRIAL, INFORMÁTICA Y DE SISTEMAS**



Área de Ingeniería en Computación e Informática



**GARRA**

Autor(es): Willian Herbas N.

Camilo Yamapara M.

Paolo Mamani C.

Curso: Proyecto I

Profesor: Ricardo Elías Valdivia Pinto

# Historial de Cambios

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha** | **Versión** | **Descripción** | **Autor(es)** |
| 16/08/2018 | 1.0 | Versión preliminar del formato | Paolo Mamani C.  Willian Herbas N. |
| 17/10/2019 | 1.1 | Versión corregida | Paolo Mamani C. |
|  | 2.0 | Informe de avance | Paolo Mamani C.  Camilo Yampara M.  Willian Herbas N. |
| Sin determinar | 3.0 | Informe final | Paolo Mamani C. |

Tabla de Contenidos

1. Panorama General
   1. Introducción (contexto)
   2. Objetivo General
   3. Objetivos Específicos
   4. Restricciones
   5. Entregables
2. Organización del Personal

2.1. Descripción de Roles

2.2. Personal que cumplirá los Roles

2.3. Mecanismos de Comunicación

1. Planificación del Proyecto

3.1. Actividades

3.2. Asignación de tiempo

3.3. Gestión de Riesgos

4. Planificación de los Recursos

4.1. Recursos Hardware-Software requeridos

4.2. Estimación de Costos (Hardware, Software, Recursos Humanos)

1. Análisis de diseño
   1. Especificación de requerimientos

5.2 Arquitectura propuesta.

5.3 Diseño de la interfaz de usuario.

6. implementaciones

6.1 Descripción de los programas implementados

* 1. Diagrama de interacción entre programas

1. Resultados

7.1Estado actual del proyecto

7.2 Problemas encontrados

7.3 Conclusiones

8. Referencias (utilizando el estándar IE)

1. Panorama General
   1. Introducción (contexto)

En este proyecto crearemos un robot el cuan mediante algoritmos que serán controlados por un usuario mediante una interfaz grafica pueda jugar un juego llamado “flip-tac-toe”, que al igual que la versión original (tic-tac-toe) consiste en hacer 3 en línea solo que en esta versión el juego es más dinámico, su principal característica es que la piezas pueden voltearse para convertirlas en una propia.

* 1. Objetivo General

Desarrollar un robot a control remoto que permita jugar el juego “FLIP-TAC-TOE”

* 1. Objetivos Específicos
* Construir un robot a base de piezas lego.
* Aprender y programar en Python.
* Diseñar una interfaz para el control.
  1. Restricciones
* Tiempo asignado para el desarrollo del Proyecto.
* Solo se puede utilizar el lenguaje python.
  1. Entregables

|  |  |  |
| --- | --- | --- |
| **Identificación entregable** | **Descripción entregable** | **Fecha de entrega** |
| Formulación del proyecto  “proyecto1” |  | 03/09/2018 |
| Presentación Formulación Proyecto |  | 05/09/2018 |
| Bitácoras | Se da a conocer todo lo que se ha hecho en la semana y lo que se debe hacer en la semana próxima. | Todos los Jueves |
| Manual de usuario | Manual de usuario donde se detallara el funcionamiento del robot | Finalización del proyecto |
| robot | Robot totalmente construido y funcional | Final del proyecto |
| wiki | Para cada informe se actualizara la wiki | Al finalizar los informes |
| Carta Gantt | **herramienta que definirá las actividades que realizaremos para finalizar el proyecto** | Entregado el primer informe |
| Video de pruebas | **Video de pruebas de avance** |  |

1. Organización del Personal

2.1. Descripción de Roles

* Jefe de Equipo:
* Coordinador:
* Constructor:
* Programador:

**Jefe de equipo** será el responsable de distribuir las tareas a los demás integrantes del equipo.

**Coordinador** de equipo será el responsable de realizar correctamente los informes, bitácoras y la preparación de las presentaciones.

**Constructor** se encargará de realizar el armado del robot correctamente.

**Programador** será el responsable de programar los algoritmos del robot y modificar

El código de instrucciones para el robot.

**Secretario** documenta cada fase del proyecto.

2.2. Personal que cumplirá los Roles

|  |  |  |
| --- | --- | --- |
| **Actividad** | **Involucrados** | **Responsables** |
| **Jefe de proyecto** | William Herbas N. | William Herbas N. |
| **coordinador** | Paolo Mamani C. | Paolo Mamani C. |
| **programador** | Camilo Yampara M.  William Herbas N.  Paolo Mamani C. | Camilo Yampara M. |
| **Constructor**  **(diseño)** | William Herbas N.  Paolo Mamani C. | William Herbas N. |
| **secretario** | Paolo Mamani C.  William Herbas N.  Camilo Yampara M. | Paolo Mamani C. |

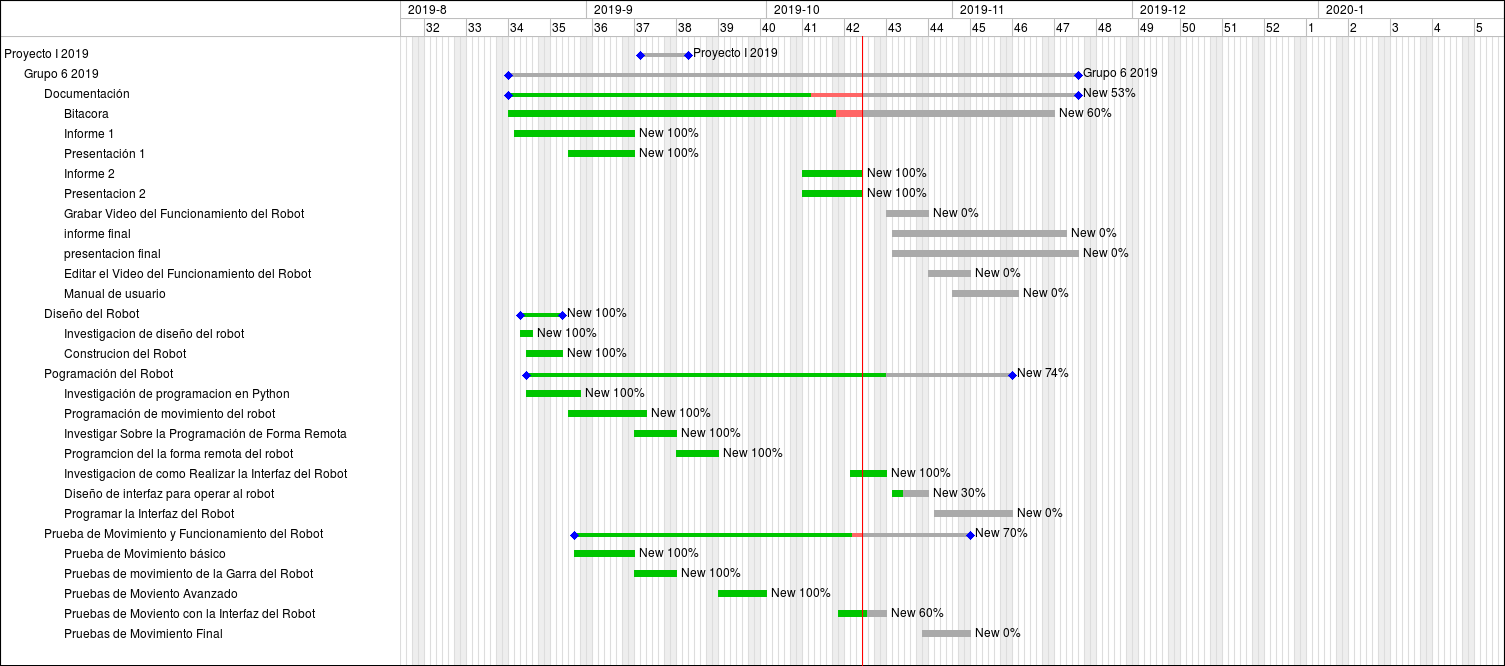
2.3. Mecanismos de Comunicación

Para tener una mejor comunicación hemos creado un grupo de WhatsApp y Messenger donde se compartirán todos los archivos para que todos los integrantes del equipo puedan tener acceso.

1. Planificación del Proyecto

3.1. Actividades (nombre, descripción, responsable, producto)

|  |  |  |  |
| --- | --- | --- | --- |
| Nombre | Descripción | Responsables | Producto |
| Formulación Proyecto | Planificación y distribución del contenido. | William Herbas  Camilo Yampara  Paolo Mamani | (TERMINACIÓN DE CADA FASE) |
| Armado del Robot EV3 | Inicio del armado del robot EV3 con un set Lego. | William Herbas  Paolo Mamani | Terminado |
| Instalación del Sistema Operativo ev3dv | Proceso fundamental de la instalación del Software en el Robot. | Camilo Yampara | Terminado |
| Programación de los códigos para el movimiento del robot. | Programación del robot con el lenguaje Python | Willian Herbas  Camilo Yampara  Paolo Mamani |  |
| Programación de la interfaz EV3 | Programación de la interfaz para manipular remotamente el robot | Willian Herbas  Camilo Yampara  Paolo Mamani |  |
| Pruebas | Pruebas del funcionamiento adecuado del Robot | Camilo Yampara |  |
| Manual de Usuario | Manual donde se enseñara a utilizar el robot. | Paolo Mamani |  |
| Bitácora | Planificación de las tareas asignadas de la semana. | Paolo Mamani |  |

3.2. Asignación de tiempo.

3.3. Gestión de Riesgos

Niveles de riesgo:

* + - 1. Catastrófico
      2. Critico
      3. Circunstancial
      4. Irrelevante

|  |  |  |  |
| --- | --- | --- | --- |
| Riesgo | Probabilidad de ocurrencia | Nivel de impacto | Acción remedial |
| Tiempo de entrega | 60% | 3 | Mejor organización de tiempos |
| Retiro de un integrante del equipo | 10% | 3 | Se dividirán las tareas del miembro retirado entre los demás miembros del equipo. |
| Falta de piezas | 15% | 3 | Se pedirán piezas al ayudante |
| Complicación de salud de algún miembro del equipo | 10% | 3 | Dependiendo de la gravedad de la enfermedad, se aligerara las tareas repartiéndolas en lo demás miembros |
| Desprendimiento de piezas | 10% | 2 | Volver a rediseñar la parte afectada |
| Daño de hardware del proyecto (sd) | 10% | 1 | Se ara copias de seguridad de todo lo avanzado. |
| Daño de software (robot) | 10% | 1 | Se solicitara la parte dañada a los ayudantes |

1. Planificación de los Recursos

4.1. Recursos Hardware-Software requeridos

|  |  |
| --- | --- |
|  | Producto |
| hardware | Robot lego mindstorm  Tarjeta micro sd  Laptop |
| software | Sistema operativo ev3  Entorno de programación visual estudio |

4.2. Estimación de Costos (Hardware, Software, Recursos Humanos)

Estimación de costo, hardware y software:

|  |  |
| --- | --- |
| Producto | Valor clp |
| Robot lego midstorm | $490.200 |
| Tarjeta micro sd | $5.000 |
| Laptop | $450.000 x 2 |
| Sistema operativo ev3 | $0(software libre) |
| Ide visual estudio | $0(software libre) |
| MobaExtern | $0(software libre) |

Estimación de costos de, recursos humanos:

1 hora de trabajo: $10000

Tiempo total de trabajo: 240h

Costo por persona: $2.400.000

Costo total:

|  |  |
| --- | --- |
|  | costo |
| Hardware y software | $940.200 |
| Recursos humanos | $7.200.000 |
| total | $8.140.200 |

5 Análisis de diseño

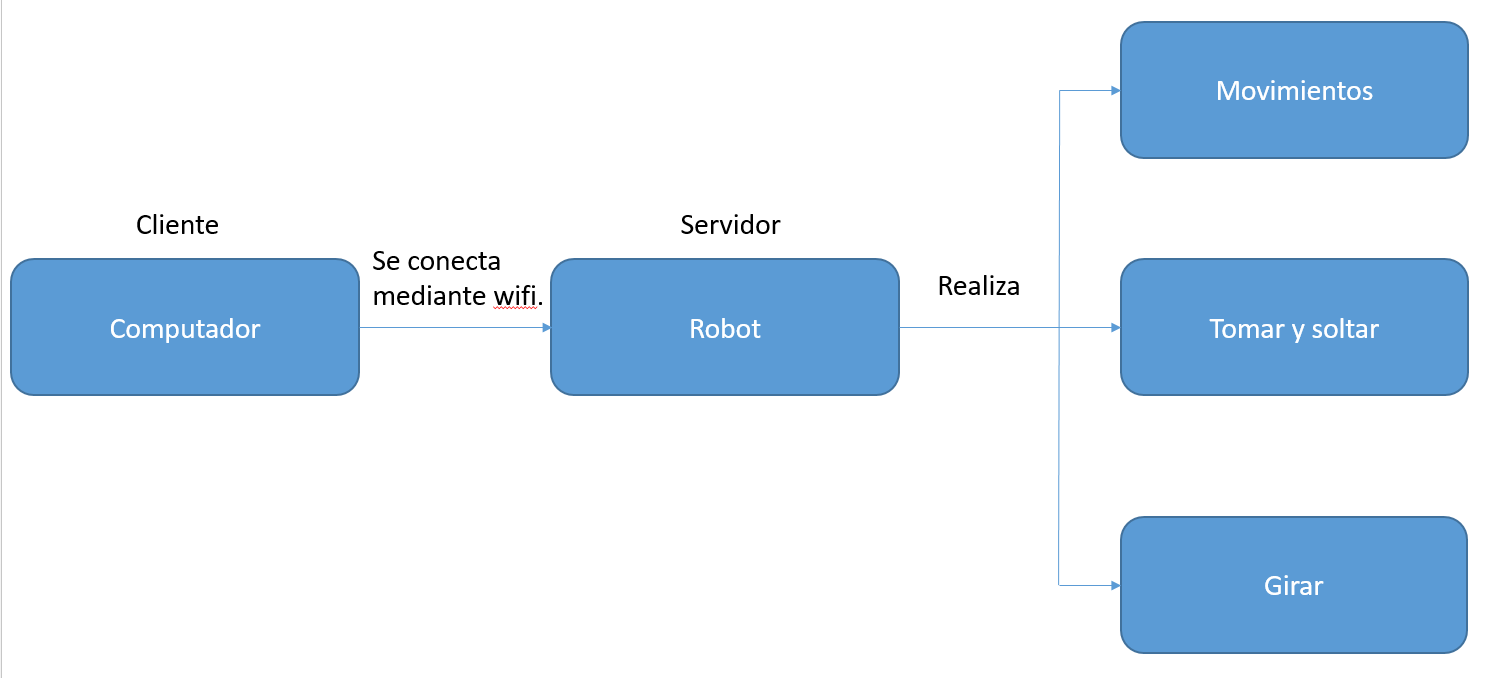
5.1 Especificación de requerimientos

Requerimientos funcionales:

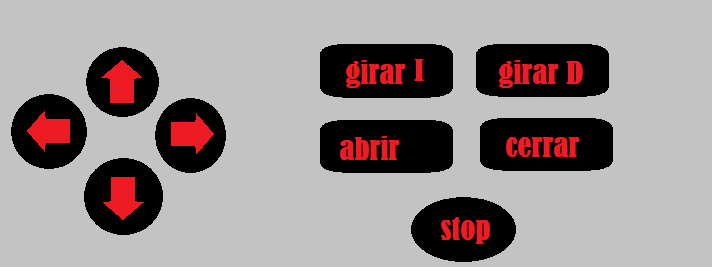
* Se implementara un robot el cual se conectara con un computador vía wifi, lo que permitirá movimientos en todas direcciones que el usuario podrá elegir mediante una interfaz gráfica.

|  |  |
| --- | --- |
| Funcional | No funcionales |
| * Se implementara un robot el cual se conectara con un computador vía wifi, lo que permitirá movimientos en todas direcciones que el usuario podrá elegir mediante una interfaz gráfica. | * El sistema debe contar con un manual de usuario * El sistema debe poseer interfaces gráficas. |

5.2 Arquitectura propuesta.



5.3 Diseño de la interfaz de usuario.



6 implementaciones.

## 6.1 Descripción de los programas implementados

* m=LargeMotor('outD'): Esta clase realiza el giro de la rueda izquierda del robot.
* s=LargeMotor('outA'): Esta clase realiza el giro de la rueda derecha del robot.
* g=MediumMotor('outB'): Esta clase se encarga del giro de la garra
* a=MediumMotor('outC'): Esta clase se encarga de abrir y cerrar la garra

6.2 Diagrama de interacción entre programas

Los movimientos básicos del robot son:

1. Hacia adelante.
2. Hacia atrás.
3. Curvar hacia la izquierda.
4. Curvar hacia la derecha.
5. Abrir y cerrar la garra.
6. Girar en ambos sentidos.

El programa interactúa de la siguiente manera:

El cliente y el servidor se comunican mediante una conexión wifi utilizando la librería “RPyC” de python.

El cliente recibe un carácter por teclado y lo envía al servidor para iniciar una acción.

Servidor

accion

Cliente

1. Resultados
   1. Estado actual del proyecto

El proyecto actualmente se encuentra en el desarrollo de la interfaz y el manual de usuario.

El código de los movimientos básicos se encuentra finalizados, por lo que el robot ya es completamente funcional.

* 1. Problemas encontrados

|  |  |
| --- | --- |
| Problemas encontrados | Soluciones propuestas |
| * Desprendimiento de piezas de la garra. | * Rediseño y reforzamiento de las zonas afectadas. |
| * Conexión de wifi deficiente. | * Adquirir una red de conexión más estable. |

* 1. conclusiones

A lo largo de este proyecto hemos encontrado muchas dificultades ya sea en armar el diseño del robot, para que sea eficiente y tenga buen aspecto, programarlo para que este tenga la capacidad de jugar flip tac toe, mediante un usuario que controle al robot desde una interfaz gráfica ejecutada desde un notebook.

1. Referencias (utilizando el estándar IEEE)

Anexos:

Anexo A: Hardware.

A continuación daremos un breve resumen de las partes robot.

* Base: es una base armada con piezas lego.
* Garra: cuenta con dos motores medianos, uno que se encargarse de abrir y cerrar la garra y otro que se encarga de girar la garra en 180°.
* Movimientos: para el movimiento del robot se utilizan dos motores grandes que hacen que el robot pueda moverse en distintas direcciones.

Anexo B: Software.

* Move\_wheel(self, orden) : es la función que está dentro del servidor que actúa dependiendo de la orden que recibe el servidor, esto ara actuar de una manera el robot, las cuales son moverse hacia adelante, atrás, Girar hacia la derecha o izquierda, además de abrir, cerrar y girar la garra.

Anexo C: Comunicación

Servidor: la comunicación cliente servidor se realiza mediante conexión wifi, utilizando la librería de python “RPyC”.

* class Control(rpyc.Service): crea una clase control y ejecuta el servidor, esta tiene una única función que recibe e inicia una acción dependiendo de valor recibido.

Move\_wheel(self, orden)

Cliente: es la parte que envía órdenes al robot, cuenta con una única función que es conseguir una clave desde el teclado y enviarla hacia el robot para su ejecución.

* conn=rpyc.connect('192.168.43.98', port=18569): inicia la conexión con el robot, y permite accede a las funciones contenidas en el servidor.
* Key\_press(): es una función que retorna una carácter cuando es presionada una tecla.
* conn.root.move\_wheel(key): envia la variable key, que contiene un carácter, a la función “move\_wheel()”.