**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. |
| 13/01/2020 | 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

Descripción de los programas implementados

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 siguiente informe detallaremos nuestro proyecto el cual es llamado “GARRA”, este proyecto consiste en la creación de un robot mediante piezas de lego mindstorns el cual tendrá algoritmos hechos en el lenguaje de programación de Python para que pueda realizar diversos movimientos los cuales podrán ser controlado por un usuario mediante una interfaz gráfica para que pueda jugar y competir en un juego llamado “FLIP-TAC-TOE”.

Este juego llamado “FLIP-TAC-TOE” 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 y su principal característica es que la piezas pueden voltearse para convertirlas en una propia.

* 1. Objetivo General

Construir y programar un robot a base de piezas de legos mindstorms el cual mediante sus funciones pueda jugar y competir un juego llamado “FLIP-TAC-TOE”

* 1. Objetivos Específicos
* Documentación de informes de avance, presentación y bitácora
* Investigación de los tipos de modelos del robot
* Construcción del robot con piezas de lego mindstorms ev3
* Programación de los movimientos del robot
* Diseño de la interfaz del robot
* Construcción del manual de usuario
  1. Restricciones
* Limitación de integrantes.
* Limitación de piezas entregadas para la construcción del robot.
* Solo se puede utilizar el lenguaje de programación Python.
  1. Entregables

|  |  |  |
| --- | --- | --- |
| **Identificación entregable** | **Descripción entregable** | **Fecha de entrega** |
| Formulación del proyecto  “proyecto1” |  | 03/09/2019 |
| Avance de proyecto | Se han completado todas las tareas para la finalización del proyecto como diseño de estructura, programación, documentación, etc. | 19/12/2019 |
| Presentación de Formulación Proyecto |  | 05/09/2019 |
| Finalización del proyecto | Entrega de todos los productos terminados | 13/01/2020 |
| 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 detallará el funcionamiento del robot | Finalización del proyecto |
| Robot | Robot totalmente construido y funcional | Final del proyecto |
| Wiki | Para cada informe se actualizará la wiki | Al finalizar los informes |
| Carta Gantt | **herramienta que definirá las actividades que realizaremos para finalizar el proyecto** | Entregado el primer informe |
| Video | **Video de usos del robot** | 13/01/2020 |

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. |

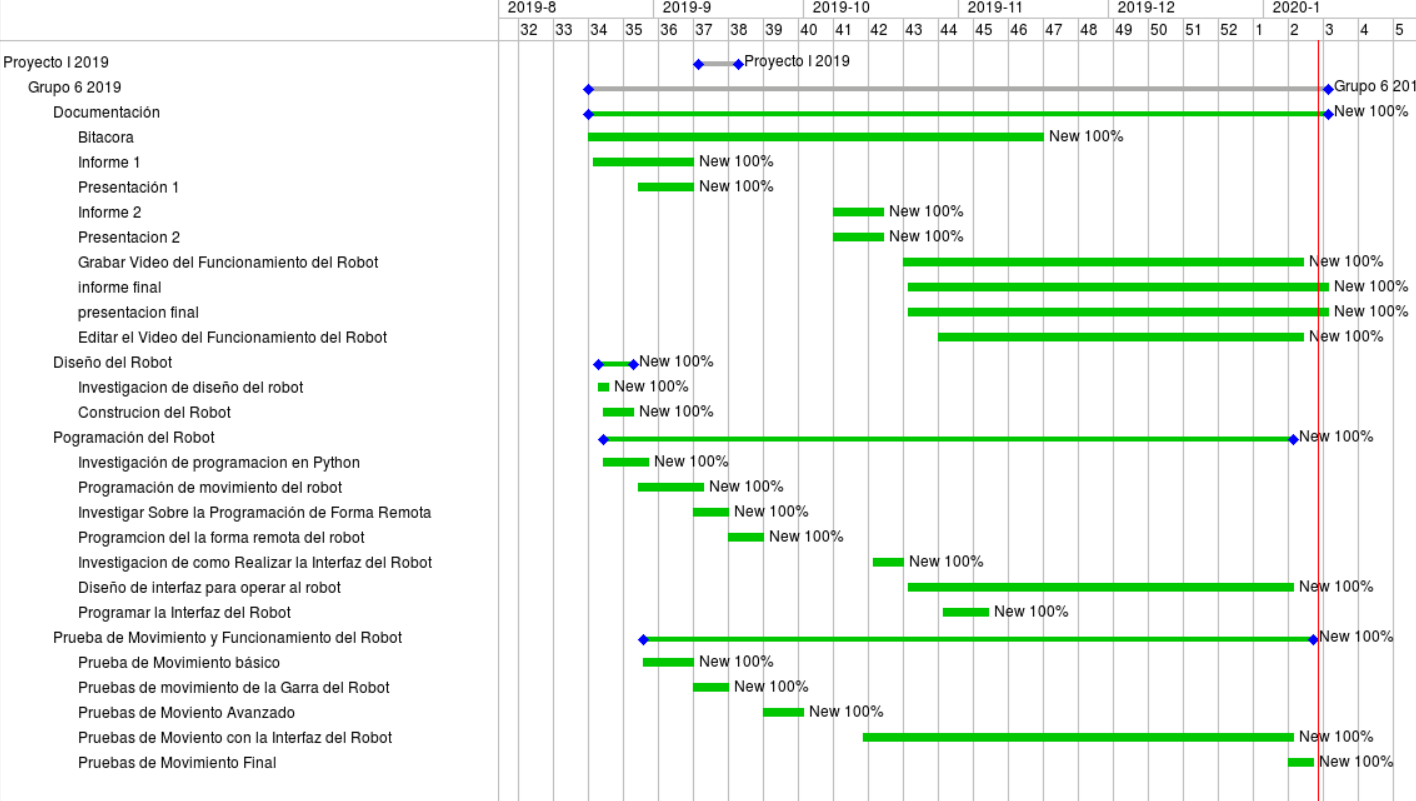
2.3. Mecanismos de Comunicación

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 | Informe 1 |
| Armado del Robot EV3 | Inicio del armado del robot EV3 con un set Lego. | William Herbas  Paolo Mamani | Diseño y  Estructura |
| Instalación del Sistema Operativo ev3dv | Proceso fundamental de la instalación del Software en el Robot. | Camilo Yampara | Robot programable |
| Programación de los códigos para el movimiento del robot. | Programación del robot con el lenguaje Python | Willian Herbas  Camilo Yampara  Paolo Mamani | Software de Control |
| Programación de la interfaz EV3 | Programación de la interfaz para manipular remotamente el robot | Willian Herbas  Camilo Yampara  Paolo Mamani | Interfaz EV3 |
| Pruebas | Pruebas del funcionamiento adecuado del Robot | Camilo Yampara | Resultados |
| Formulación del Manual de Usuario | Manual donde se enseñara a utilizar el robot. | Paolo Mamani | Manual de Usuario |
| Bitácora | Planificación de las tareas asignadas de la semana. | Paolo Mamani | Registro de tareas realizadas |



3.2. Asignación de tiempo.

En esta Carta-Gantt se plantearon todos los pasos que se debían seguir para concluir el proyecto, la cual se realizó con todos los integrantes para plantear los puntos y estimar el tiempo invertido en cada uno.

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

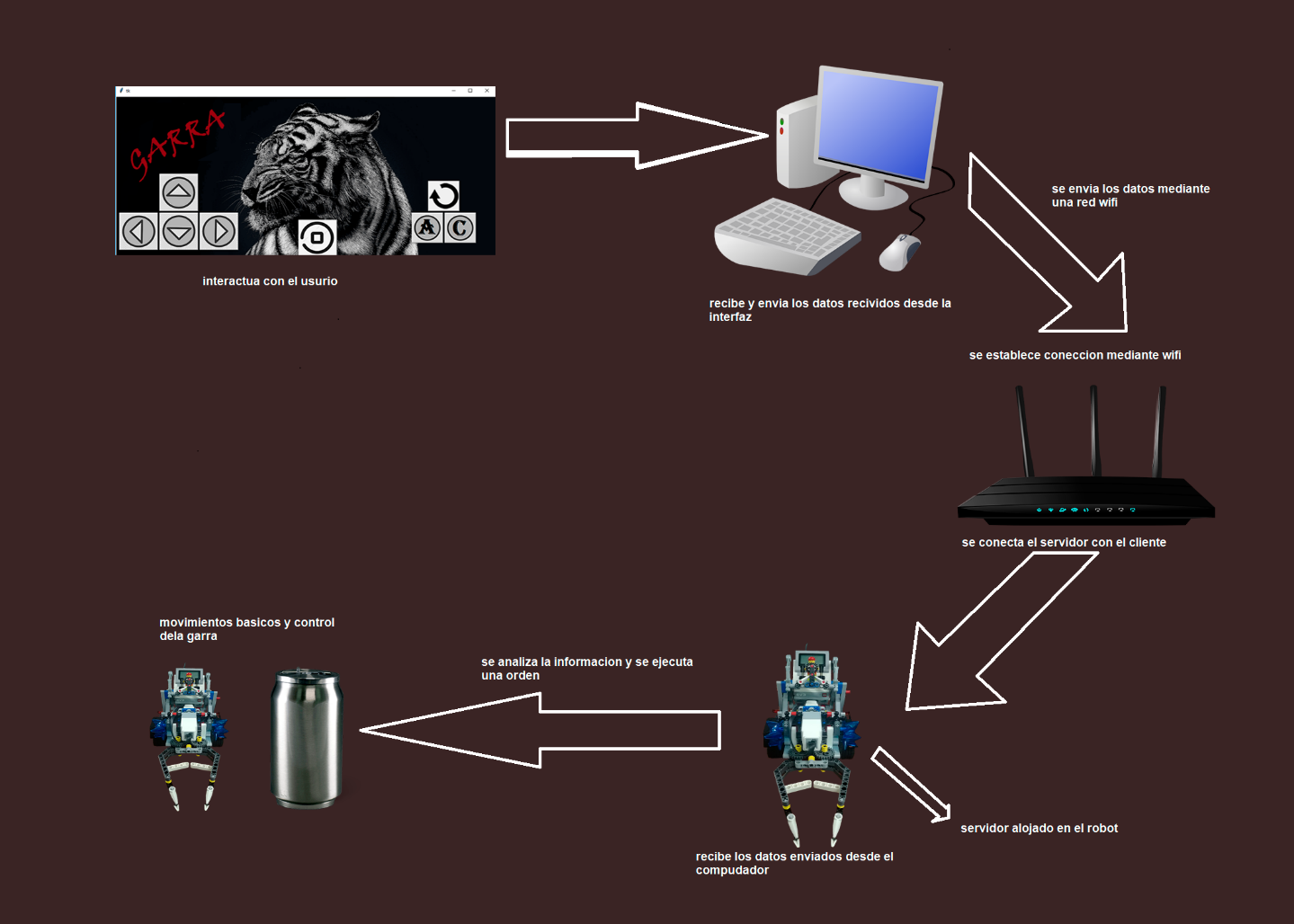
5.1 Especificación de requerimientos

Requerimientos funcionales:

* Se implementará un robot el cual se conectará 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 y además permitirá girar, agarrar y soltar la garra. El usuario podrá elegir estas acciones mediante una interfaz gráfica. * tiene que ser capaz de jugar flic-tac-toe | * El sistema debe poseer interfaz gráfica. * El programa debe ser echo Python * El robot debe ser echo con piezas de Lego mindstorms ev3 |

5.2 Arquitectura propuesta.

1.- interfaz gráfica: software de fácil entendimiento, recibe datos de entrada para controlar el robot.

2.- computador: es donde se ejecuta el software, envía los datos.

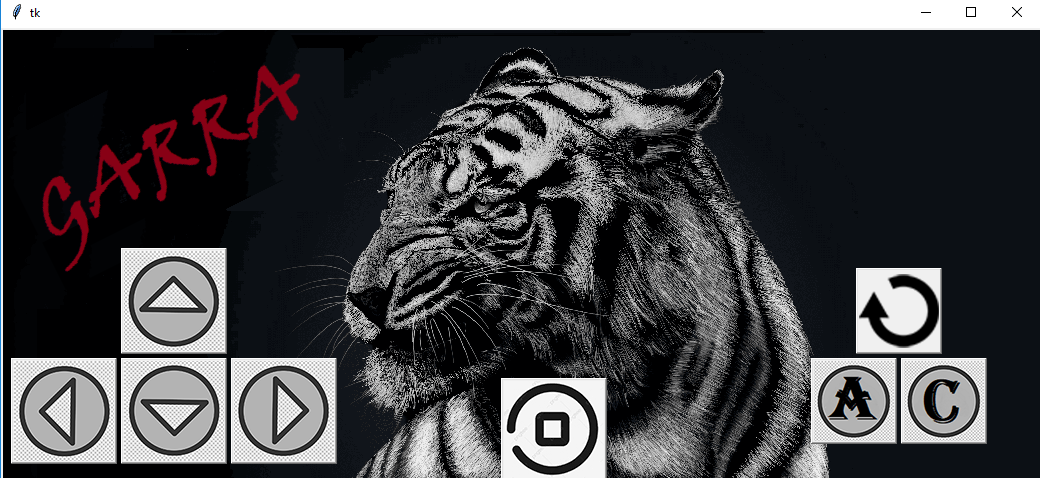
3.- Router: sirve de medio de comunicación, permite establecer una conexión entre servidor y cliente atravez de una red wifi.

4.- Robot: es el dispositivo que será controlado, este contiene el servidor.

5.- Acciones: son los movimientos básicos del robot, avanzar, retroceder, girar hacia la derecha y ala izquierda.

También los movimientos de la garra que son tomar, soltar y girar un objeto.

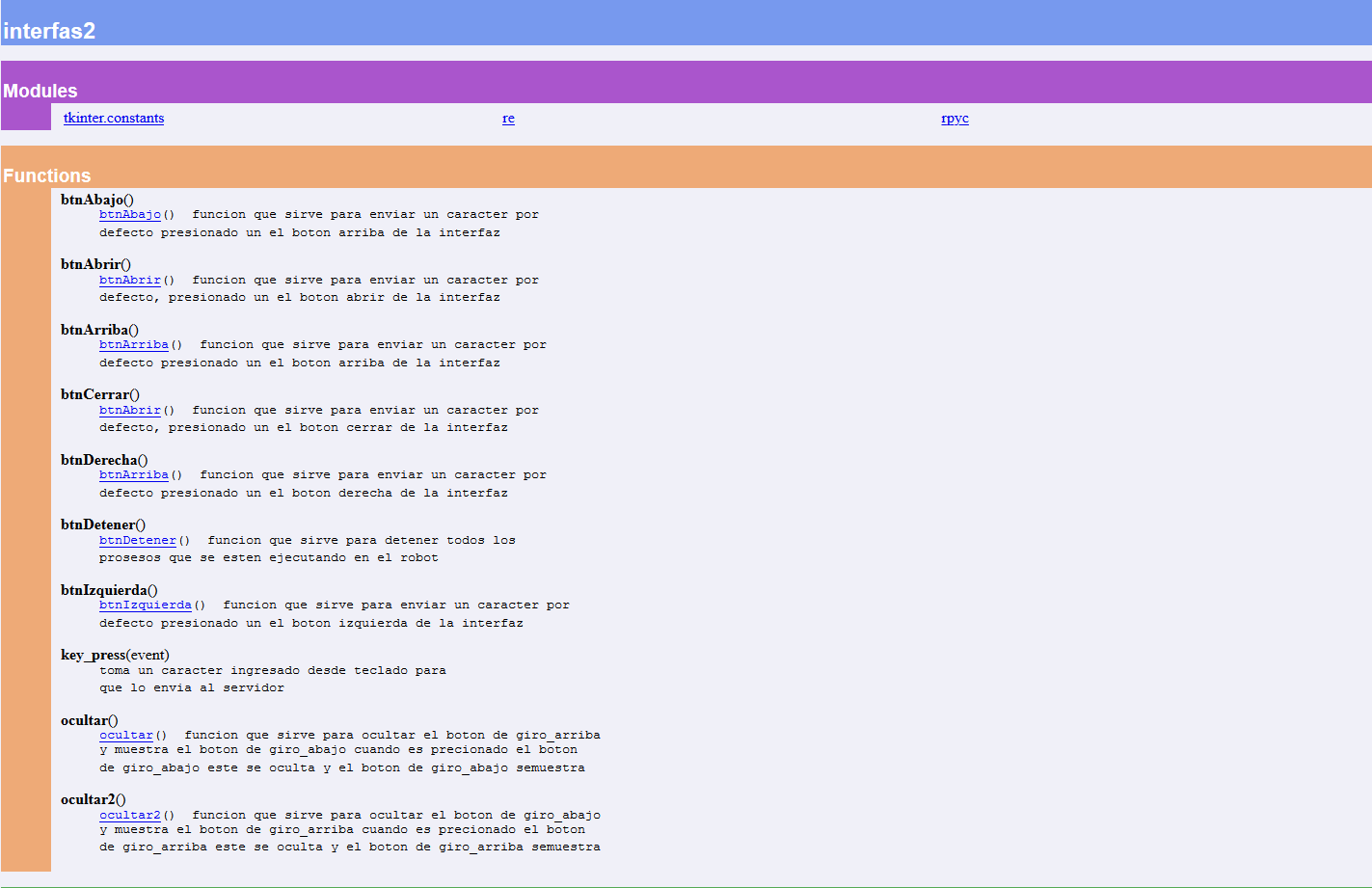
* 1. Diseño de la interfaz de usuario.

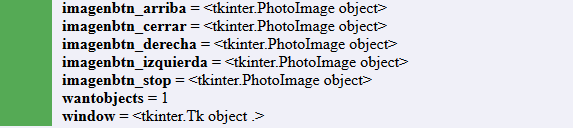


Como se puede apreciar esta es la interfaz de los movimientos del robot, en la esquina izquierda de esta se encuentran los botones para controlar sus movimientos (arriba, abajo, izquierda, derecha), en la parte central de esta se encuentra el botón de stop, el cual detiene todos los movimientos del robot y en la parte izquierda se encuentran tres botones los cuales son del movimiento de la garra, el “A” para abrir la garra, el “C” para cerrarla y la flecha para girar la garra.

6 implementaciones.

Documentación del cliente





Documentación del servidor



1. Resultados
   1. Estado actual del proyecto

Servidor: para el servidor utilizamos la librería de python “RPyC”, el servidor se encuentra programado en el robot, está terminado y ya es completamente funcional.

Cliente: para la interfaz se utilizó la librería “Tkinter”, su estado es terminado y ya es completamente funcional.

Diseño de robot: para el diseño del robot investigamos varios modelos y tomamos ideas para lo que es necesario para nuestro robot, el modelo ya está terminado.

7.2 Problemas encontrados

|  |  |
| --- | --- |
| Problemas encontrados | Soluciones propuestas |
| * Desprendimiento de piezas de la garra. | * Rediseño y reforzamiento de las zonas afectadas. |
| * Conexión de wifi deficiente. | * Utilizamos nuestra red de celular, soluciono el problema parcialmente. |
| * Poca información en español sobre las librerías RPyC. | * Utilizar el traductor de google. |
| * Problemas con la documentación del servidor. | * Pedir ayuda al profesor. |
| * Poco personal para llevar a cabo el proyecto. | * Aumentar la cantidad de reuniones de trabajo. |

* 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 una interfaz gráfica, sin embargo, gracias a todos estos inconvenientes hemos aprendido a trabajar en equipo y mejorar nuestra capacidad de organización para así poder estar preparado para el futuro.

1. Referencias (utilizando el estándar IEEE)

# [1] D. Becerra, <<Lego Digital Designer [ LDD ] | Robot (유)>>, 3 dic. 2017. [en linea]. Available: <https://www.youtube.com/watch?v=0fCW0Q1X5DU>

# [2] CEP de Lanzarote, <<1.6 Construcción de modelos>>, 2014-2015. [en línea]. Available:<http://canaltic.com/rb/legoev3/16_construccin_de_modelos.html?fbclid=IwAR3Q8fjn_eS9d8oPX_3wXmVA7JEWa2Rk50diASf9kJzXLsnfko55q9tvQzs>

# [3] W.Flores, <<Most Simple EV3 Robot Claw>>, 16 abr. 2015. [en línea]. Available: <https://www.youtube.com/watch?v=VEsLKZAAoSc&fbclid=IwAR1GUgdhtIZVTmRlCr5mVal6KCmdbynZhLUXs_AHbaXacVbFUdQ7e50pJRo>

# [4] J. Mertz, <<Documenting Python Code: A Complete Guide>>, 2018-2019. [en lines]. Available: <https://realpython.com/documenting-python-code/#reader-comments>

# [5] Pherkad, << Docstrings >>, 1 feb. 2014. [en linea]. Available: <https://python-para-impacientes.blogspot.com/2014/02/docstrings.html>

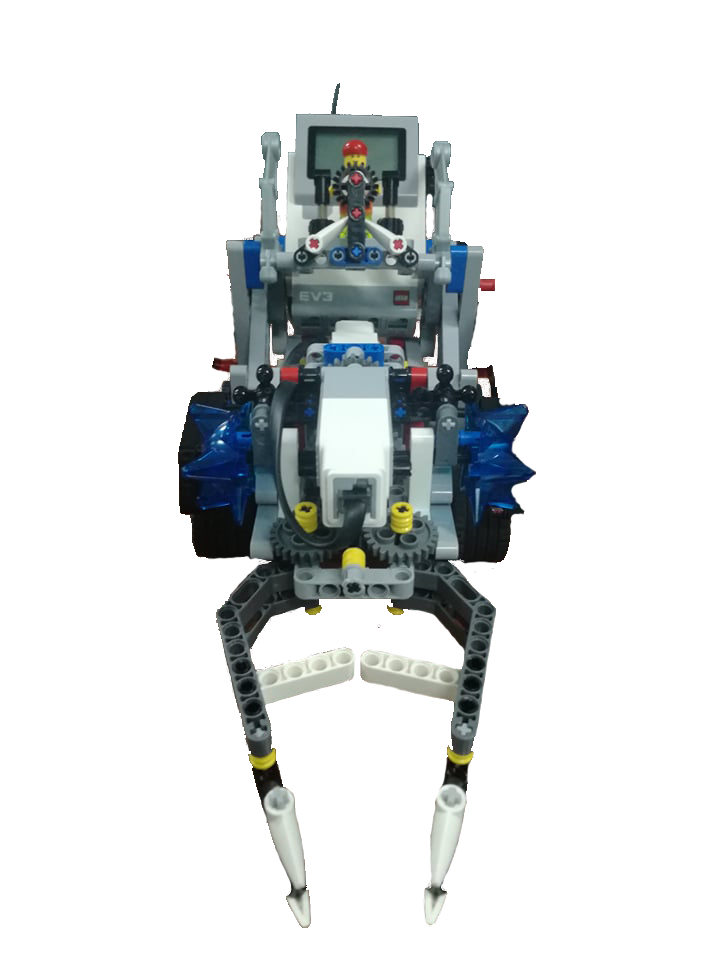
# [6]. Nigel Ward, <<ev3devpython >>, 15 jun. 2019. [En línea]. Available: <https://sites.google.com/site/ev3devpython/>

# [7] Starr Cards, << Flip-Tac-Toe puts heart-pumping spin on tic-tac-toe >>, 16 abr. 2015. [en línea]. Availlable : <https://www.youtube.com/watch?v=nvRmpKmmcPc>

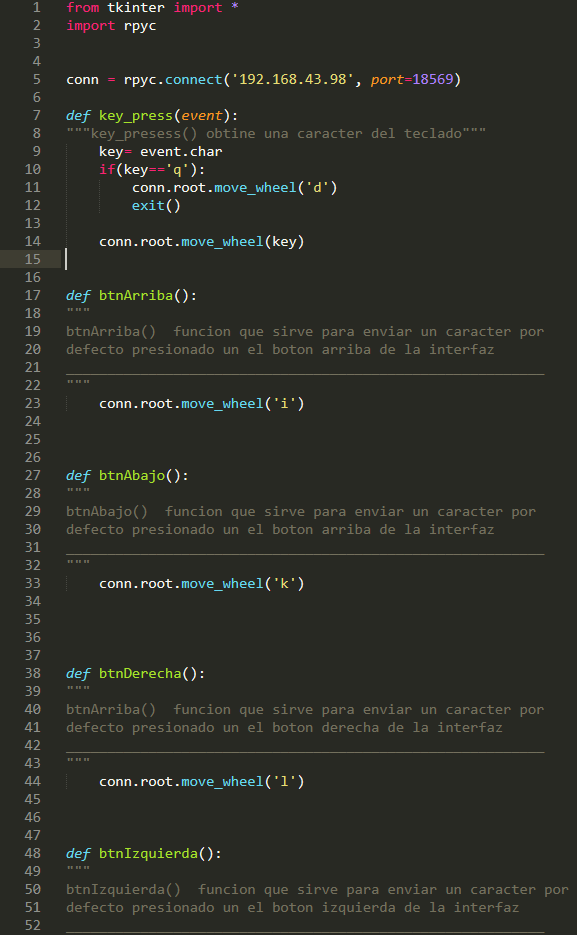
Anexos:

Anexo A: Hardware.

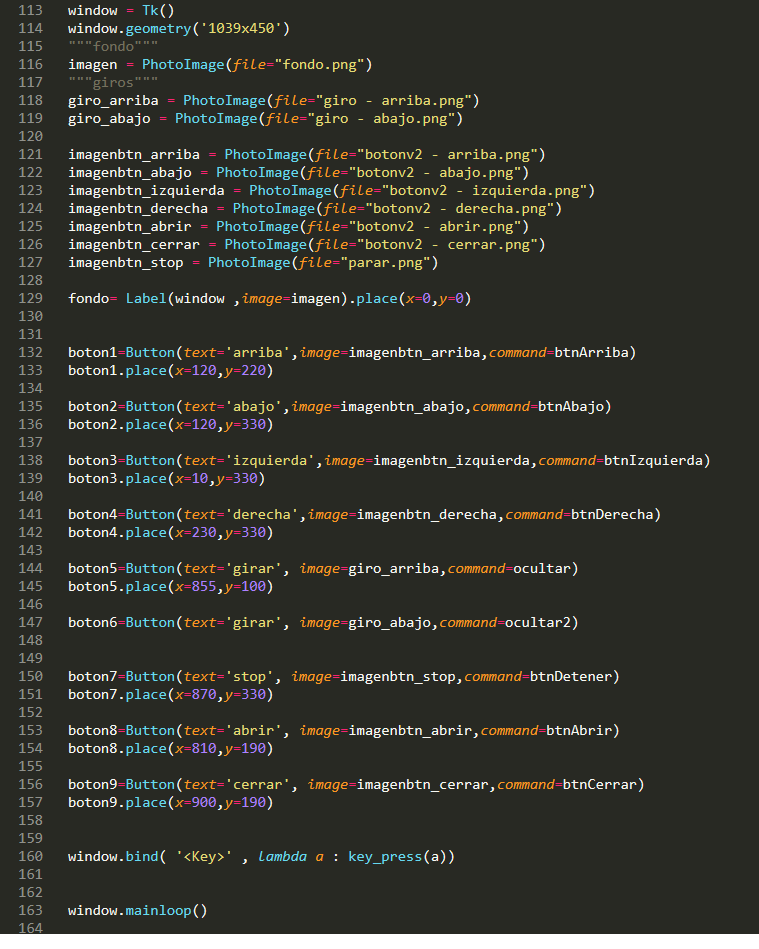
A continuación daremos un breve resumen de las partes robot.

* 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.

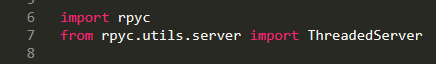
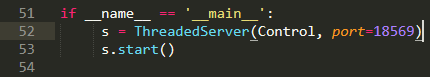
Anexo B: Software.

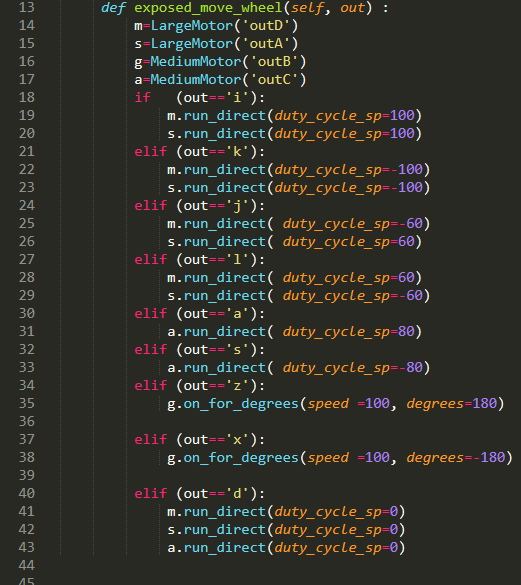
A continuación se muestra el código del cliente el cual contiene la interfaz grafica





Anexo C: Comunicación

Servidor: para la comunicación utilizamos la librería “RPyC” de python

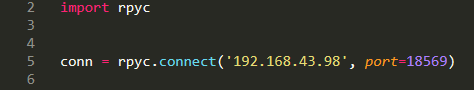
* C:\Users\Rodrigo\Downloads\ev3dev\Nueva_carpeta\documentacion\servidor2.pngclass Control(rpyc.Service): clase que controla el servidor y todas sus funciones.
* def exposed\_move\_wheel(self,out): function que ejecuta las acciones del robot.

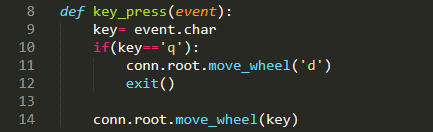
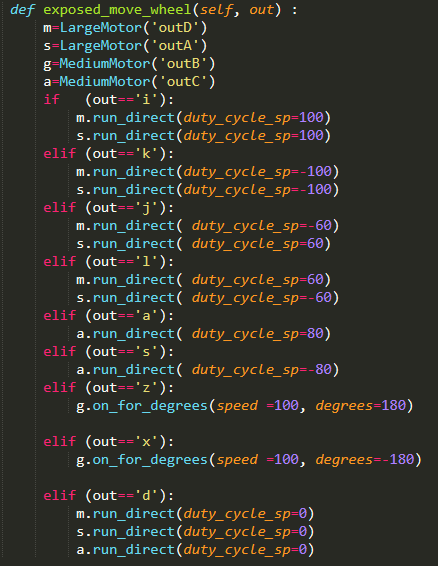
Cliente: para la interfaz del cliente utilizamos la librería “Tkinter”, es la parte que envía órdenes al robot, cuenta con 2 opciones para controlar el robot.

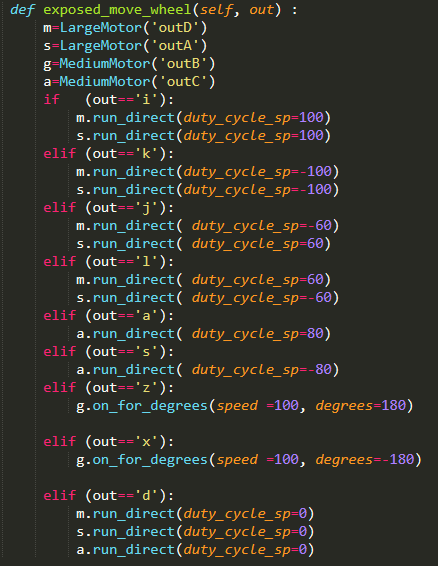
La primera es mediante botones, y la segunda es el control utilizando el teclado.

En el cliente utilizamos nuevamente la librería “RPyC” para poder conectar el cliente con el servidor.

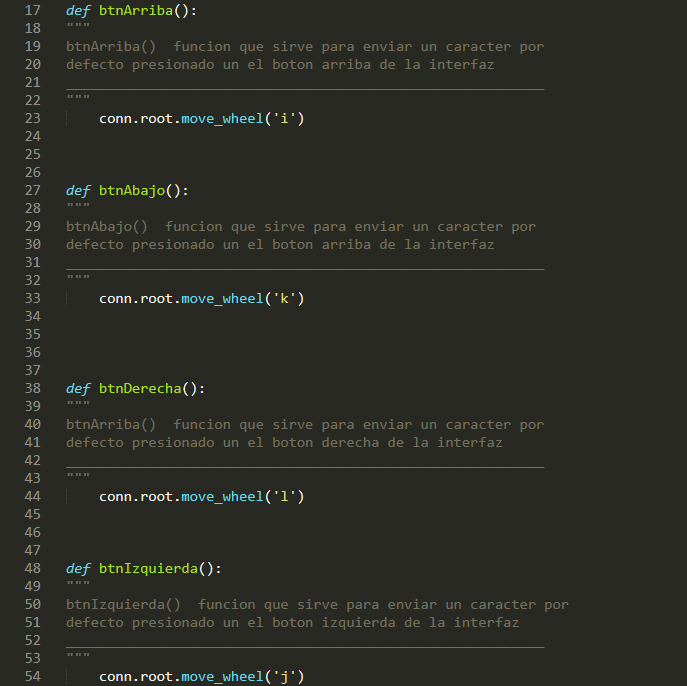
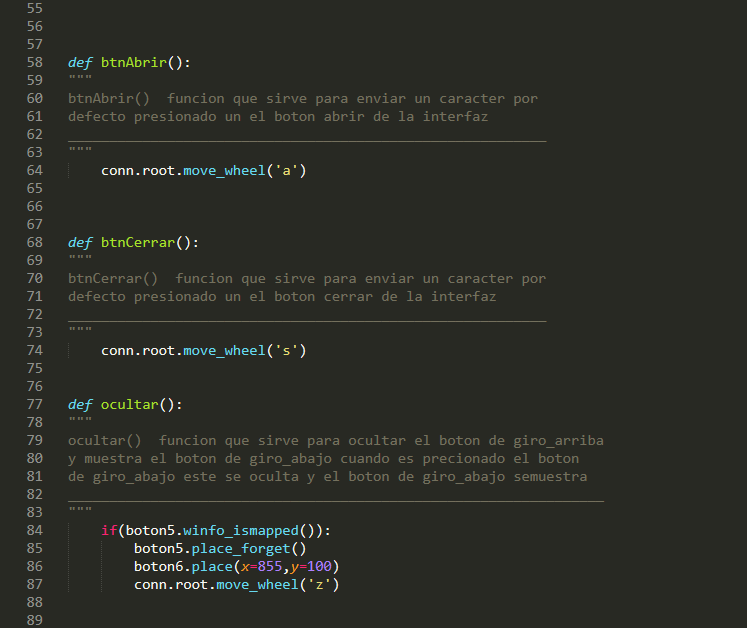
C:\Users\Rodrigo\Downloads\ev3dev\Nueva_carpeta\documentacion\cliente1.png

* 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.
* C:\Users\Rodrigo\Downloads\ev3dev\Nueva_carpeta\documentacion\cliente3.pngwindow.bind( '<Key>' , lambda a : key\_press(a)): esta línea recibe un evento del teclado y la envía como una variable “a” a la función key\_press(event).
* Key\_press(): es una función que recibe una variable y la transforma a carácter para luego enviarla al servidor.

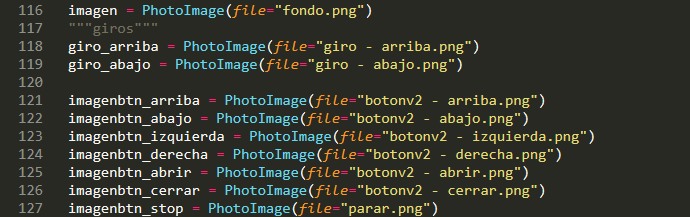
conn.root.move\_wheel(key): envía la variable key, que contiene un carácter, a la función “move\_wheel(self,out)” que está en el servidor.



* Botones, configuraciones y sus funciones.
  + - * + Botones: acciones y configuración, utilizando “command” se llama al funcional presionarlo.

Sus funciones



Configuración de las imagenes.