**Tecnológico de Costa Rica**

Análisis de algoritmos

***Curso***

M. Ed. David Mora

***Profesor***

Proyecto #01 – Solitario

***Trabajo***

Semestre #1 – 2020

***Semestre - Año Lectivo***

Ingrid Fernández - 2017089872

Aaron Vargas - 2018163800

Daniel Calderón -

***Integrantes***

Contenido

[**Requisitos Generales:** 3](#_Toc25117033)

[**Requisitos Funcionales:** 3](#_Toc25117034)

[**Alcances y limitaciones del programa:** 3](#_Toc25117035)

[**Procedimientos de desarrollo:** 4](#_Toc25117036)

[Chasis básico 4x4 8](#_Toc25117037)

[Sensor de distancia ultrasónico HY-SRF05 8](#_Toc25117038)

[Raspberry Pi 3 Starter Kit 8](#_Toc25117039)

[Puente doble H L293D(H-Bridge) 8](#_Toc25117040)

[Cable de puente macho a macho 8](#_Toc25117041)

[Cable de puente macho a hembra 8](#_Toc25117042)

[**Diseño inicial de las funciones para el proyecto** 9](#_Toc25117043)

[**Dependencias externas:** 9](#_Toc25117044)

[**Análisis de resultados:** 9](#_Toc25117045)

[**Lecciones aprendidas:** 10](#_Toc25117046)

[**Bibliografía** 10](#_Toc25117047)

En el este documento se presenta la documentación necesaria para un mejor entendimiento de la solución que creamos para el proyecto número 1 del curso Análisis de algoritmos el cual consiste en crear un juego de solitario.

El cual nos ayuda a entender más a fondo el uso de las diferentes técnicas de programación vistas hasta el momento en el curso como lo es el backtracking, divide y conquista, reduce y conquista, programación dinámica y algoritmos voraces.

El uso del programa está dirigido a toda la población que cuente con un conocimiento básico en lo que al uso de computadoras se refiere y tenga un leve conocimiento sobre las reglas del juego.

# **Requisitos Generales:**

1. Crear un programa que utilice los pines GPIO de un Raspberry Pi para el control de un “Mini-Carro”.

# **Requisitos Funcionales:**

1. Utilizar los pines de entrada y salida GPIO del raspberry pi.
2. Utilizar un sensor de distancia ultrasónico, para la detección de objetos frente al vehículo y dependiendo del modo en el que se encuentre (rastreador o autónomo) se obtiene una respuesta.
3. Realizar la codificación de los tres modos que tendrá el “mini-carro”.
4. Realizar la circuitería necesaria para el funcionamiento del “mini-carro”.

# **Alcances y limitaciones del programa:**

El programa cumple con todos los requisitos tanto generales como funcionales de la forma que fueron pensados por el grupo de trabajo. El dispositivo sufre de ciertas limitaciones ya que la batería se agota rápidamente lo que provoca que gradualmente su fuerza y velocidad para avanzar, retroceder y principalmente girar disminuya.

# **Procedimientos de desarrollo:**

**Herramientas Utilizadas:**

Para la codificación de este proyecto fue necesaria la utilización de un lenguaje de programación llamado Python para realizar los modos con los cuales iba a contar el “mini-carro”, el IDE utilizado fue Thonny Python disponible dentro del entorno del Raspbian OS.

Para la construcción del carro fueron necesarios los materiales que se muestran en la Figura 1, Figura 2 , Figura 3 , Figura 4 y Figura 5. La lista completa puede verse en la tabla **MaterialesUsados**

Imagen que contiene texto

Descripción generada automáticamente

Figura 1

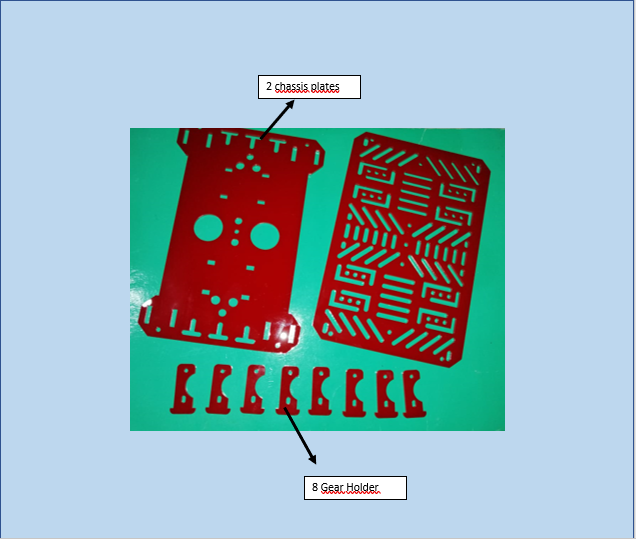


Figura 2

Imagen que contiene captura de pantalla

Descripción generada automáticamente

Figura 3

Imagen que contiene captura de pantalla

Descripción generada automáticamente

Figura 4

Imagen que contiene captura de pantalla

Descripción generada automáticamente

Figura 5

**MaterialesUsados**

|  |  |  |
| --- | --- | --- |
| **Nombre** | **Características** | **Precio** |
| Chasis básico 4x4 | * Ruedas de plástico de 4 x 65 mm (30 mm de ancho) con neumáticos de goma * 2 x placas de chasis (rojo) con una variedad de agujeros de montaje * 4 GearMotors * Todos los tornillos | $24.95 |
| Sensor de distancia ultrasónico HY-SRF05 |  | $ 4.49 |
| Raspberry Pi 3 Starter Kit | 1 x  [Raspberry Pi 3 - Modelo B +](https://www.crcibernetica.com/raspberry-pi-3-b-latest-version/)  1 x   [tarjeta MicroSDHC UHS-I con adaptador (clase 10) 16GB](https://www.crcibernetica.com/microsd-card-with-adapter-16gb-class-10/)  1 x [Fuente de alimentación del adaptador de pared - 5.25V DC 2.5A (USB Micro-B)](https://www.crcibernetica.com/wall-adapter-power-supply-5v-dc-2-5a-usb-micro-b/) (Corriente extra para Raspberry PI 3) 1 x  [Cable HDMI - 6 '](https://www.crcibernetica.com/hdmi-cable-6/)  1 x [GPIO en forma de T a](https://www.crcibernetica.com/raspberry-pi-wedge-gpio-breakout/) placa de interfaz de placa de prueba para Raspberry PI 3  1 x [teclado 4x4](https://www.crcibernetica.com/sealed-membrane-4x4-button-pad-with-sticker/)  1 x [Funda Raspberry Pi Negra](https://www.crcibernetica.com/raspberry-pi-case-black/) | $ 98.95 |
| Puente doble H L293D(H-Bridge) | * 2 x H-Bridge | $ 2.49 c/u |
| Cable de puente macho a macho | * 65 cables | $ 4.95 |
| Cable de puente macho a hembra | * 30 cables | $ 2.49 |

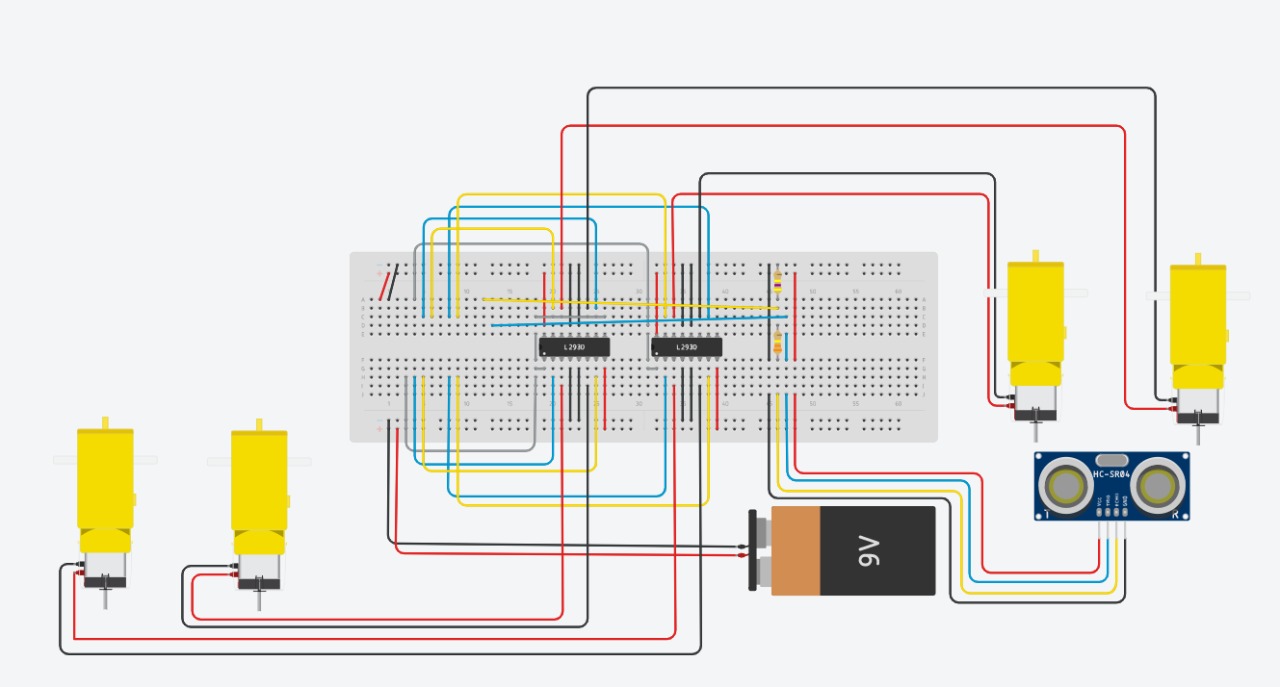
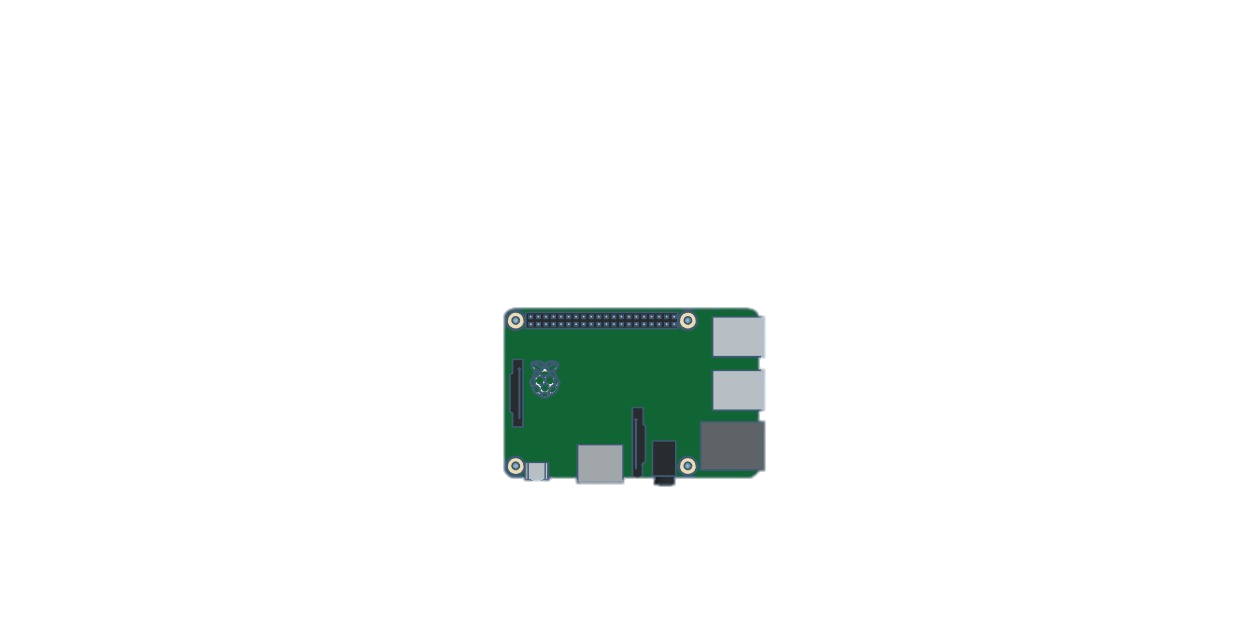
**Planificación:**

Inicialmente realizamos la propuesta del dispositivo que deseábamos llevar a cabo, realizamos la búsqueda de los componentes que necesitaríamos para el desarrollo del dispositivo y revisamos que cumpliera con las indicaciones brindadas por el profesor.

En los siguientes días decidimos realizar la compra de los componentes, una vez teníamos los componentes iniciamos una parte del equipo con la investigación del funcionamiento del sensor de distancia ultrasónico, otro investigó el funcionamiento de los motores y otro investigó el funcionamiento de los h-bridge.

En el día que tuvimos la información de las investigaciones una parte del equipo inició la construcción del dispositivo y la otra parte inicio la codificación de los modos. Al finalizar la construcción y codificación realizamos pruebas y corregimos los inconvenientes que se nos presentaron, finalmente realizamos la documentación y creación de manuales necesarios.

**Diseño circuito para el proyecto:**



# **Dependencias externas:**

El programa depende de un cliente y servidor VNC para el acceso remoto al Raspberry Pi, también depende de un IDE de Python disponible dentro del entorno de Raspbian OS para la creación del programa y para su ejecución se puede utilizar el mismo IDE o bien la consola de Raspbian OS.

# **Análisis de resultados:**

1. Utilizar los pines de entrada y salida GPIO del raspberry pi: **Alcanzado exitosamente**
2. Utilizar un sensor de distancia ultrasónico, para la detección de objetos frente al vehículo y dependiendo del modo en el que se encuentre (rastreador o autónomo) se obtiene una respuesta: **Alcanzado exitosamente**
3. Realizar la codificación de los tres modos que tendrá el “mini-carro”: **Alcanzado exitosamente**

Aunque se presentaron problemas para conseguir que el “mini-carro” pudiera girar ya que las llantas delanteras creaban mucha fricción por lo que como solución usamos cinta aislante para minimizar la fricción.

1. Realizar la circuitería necesaria para el funcionamiento del “mini-carro”: **Alcanzado exitosamente**

# **Lecciones aprendidas:**

* Este proyecto nos ayudó a aprender cómo funciona los pines GPIO de un Raspberry Pi
* Nos brindó la posibilidad la oportunidad de crear un dispositivo de forma física, ver el cuidado y dificultades que conlleva.
* Nos mostró también lo bueno de trabajar en equipo y poder delegar trabajos, así como la retroalimentación entre el equipo para mejorar y agregar al proyecto.

# 

# **Bibliografía:**

Materiales:

<https://www.crcibernetica.com/male-to-female-jumper-wire-40-pcs-in-one-bunch/>

<https://www.crcibernetica.com/male-to-male-jumper-wire-65pcs-in-one-bunch/>

<https://www.crcibernetica.com/basic-4x4-chassis-with-4-gear-motors-and-4-wheels/>

<https://www.crcibernetica.com/l293d-dual-h-bridge-16-pin-dip/>

<https://www.crcibernetica.com/crcibernetica-raspberry-pi-3-starter-kit/>

<https://www.crcibernetica.com/hc-sr05-ultrasonic-distance-sensor/>

Como conectar Sensor de distancia ultrasónico:

<https://tutorials-raspberrypi.com/raspberry-pi-ultrasonic-sensor-hc-sr04/>

Como escuchar cuando se acciona una tecla:

<https://zsiegel92.github.io/evilpython/lesson_6.html>

Como conectar los h-bridge:

<https://business.tutsplus.com/es/tutorials/controlling-dc-motors-using-python-with-a-raspberry-pi--cms-20051>

DataSheet gearmotor:

<http://dlnmh9ip6v2uc.cloudfront.net/datasheets/Robotics/DG01D.pdf>