

Computer Engineering

(CEAB Substantially Equivalent Program and CFIA Accredited Seals)

Project II (Part 1): Formula E CE Tec Energy Saving and Telemetry Part I

Professors:

Pedro Gutiérrez García, Jeff Schmidt Peralta, Alejandro Vargas Chaves
& Milton Villegas Lemus

"Failure is an option here.

If things are not failing, you are not innovating enough."

"When something is important enough,
you do it even if the odds are not in your favor."

Elon Musk

Computer science inverts the normal.

In normal science, you're given a world, and your job is to find out the rules.

In computer science, you give the computer the rules,
and it creates the world.

Alan Kay





Objetivo General

- Construir un proyecto que integre el desarrollo de Software y Hardware. Aplicar conceptos básicos de proyectos propios de Ingeniería en Computadores.

Objetivos Específicos

- Construir un dispositivo independiente que reciba comandos de forma remota, a partir de una aplicación en el computador.
- Implementar un circuito, a partir de un diagrama electrónico, elaborar un plan de pruebas y construir un módulo final.
- Desarrollar un software capaz de comunicarse con otro, mediante un protocolo previamente definido.
- Aplicar las habilidades adquiridas en los talleres.

Antecedentes

La telemetría es una tecnología que permite la medición remota de magnitudes físicas y el posterior envío de la información hacia el operador del sistema.

En fórmula 1 la telemetría se introdujo a finales de los años 80, antes de eso, el piloto era el único mecanismo para analizar y entender el automóvil. En 1975 McLaren por primera vez implementó la telemetría donde se capturaba 14 variables que podían descargarse una vez el carro regresaba a los garajes

En los años 80, los equipos de Fórmula 1 incorporaron cada vez más tanto los microprocesadores como la miniaturización electrónica, en tareas de administración del desempeño, además de la telemetría. Los automóviles comerciales de hoy día, incorporan dichos elementos los que mejoran su eficiencia y confiabilidad. McLaren en la segunda mitad de los 80 introdujo el envío de lecturas por ráfagas cada vez que el auto pasaba frente al garaje, o que permitía a ingenieros y técnicos realizar ajustes antes de que el auto regresara al garaje [5].

Fue hasta los años 90 que se vio una explosión del uso computacional no solo en los autos sino en el garaje mismo. Un equipo de Fórmula 1 utiliza gran capacidad de poder computacional de última generación para medir, controlar, analizar además de simular las diferentes variables de un carro de competencia. Hoy día la F1 emplea internet para enviar “stream” de datos para los controles de carrera, autos y hasta para la televisión. Se calcula un ancho de banda diez veces más veloz que la mayor velocidad disponible para los hogares. De esta forma la telemetría puede viajar alrededor del globo a diferente personal de apoyo a cada uno de los equipos. La FIA (Federación Internacional del Automovilismo) impuso restricciones sobre el alcance de la manipulación remota sobre los autos, para permitir una competencia más equilibrada, con ello ha impulsado la ingeniería a procesos más refinados que optimizan recursos y que las escuderías trasladarían a los autos de calle.

De la mano con la velocidad de los autos de la Fórmula 1, van la velocidad en el desarrollo de computadores y el desarrollo del software.

En 2011 nació una idea con el fin de promover la movilidad de forma sostenible para contribuir en la limpieza del planeta. Esta idea fue del presidente de la FIA (Jean Todt) y un empresario español (Alejandro Agag) [6]. Hizo su debut en 2014 en la ciudad de Beijing China. En los campeonatos de fórmula E, denominada así por categoría para automóviles eléctricos monoplaza, se hereda el concepto de telemetría de la F1, para obtener datos que permitan generar estadísticas y así conocer el rendimiento del auto. Aunque su uso es mucho más restringido por el reglamento que los de la F1.

Descripción del proyecto

Es natural entonces, concebir un proyecto para Ingeniería en Computadores que no solo integre hardware y software sino también el uso de comunicaciones inalámbricas, pero la Fórmula E forma entonces el antecedente ideal para un proyecto de curso inicial.

En este proyecto se desea aplicar la telemetría a un carro de juguete, de esta manera conocer su estado general y controlar algunos de sus componentes desde un computador.

El trabajo se divide en dos partes: el hardware que ejecuta los comandos y el software que controla al hardware. El diagrama de Arquitectura del Proyecto, se pude apreciar en el diagrama 1 donde se muestran los componentes más relevantes del sistema incluyendo todas sus partes.

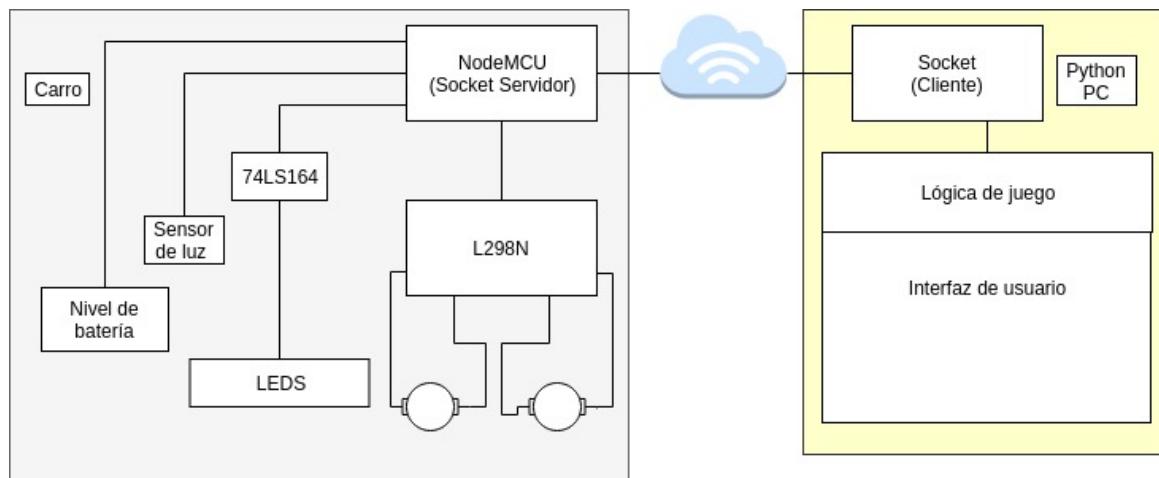


Diagrama 1. Arquitectura del Sistema

Parte I: Desarrollo del Hardware

Para el desarrollo del proyecto se requiere la consecución de un carro de juguete a baterías de dos motores. Se insta al grupo obtener el carro por métodos de convicción con hermanos, primos, amigos etc. No es necesario la compra de un carro nuevo. Los motores deben servir y la carcasa puede ser adaptada.

A la carrocería se le van a habilitar dos luces delanteras, luces direccionales, dos luces de parada y un sensor de luz exterior. Estas carrocerías se pueden cortar con

cuchillas de precisión o navajas x-acto, para hacer compuertas o accesos y hacer la adaptación más estética. En las fotos 1 y 2 se pueden apreciar las adaptaciones realizadas en proyectos similares.



Foto 1 Habilitación de una portezuela cortando la carrocería y substitución de bombillos por Leds.



Foto 2. Habilitación de conectores y luces exploradoras y de parada. Este carro fue pintado como detalle estético.

En esta sección se explicará el rediseño del carro.

Consiste en utilizar módulos de bajo costo y construir un circuito eléctrico que realice las funciones deseadas en el carro, para ejecutar las órdenes que reciba del computador central.

Los Módulos a utilizar son:

- **NodeMCU:** Es un módulo que integra un microcontrolador y un adaptador de WiFi, es el encargado de recibir los datos del computador y generar las señales eléctricas para los demás componentes[1][3].

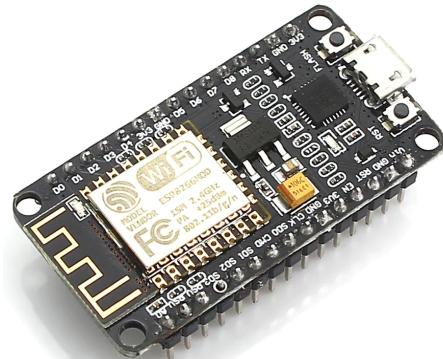


Figura 1. Módulo NodeMCU

- **L298N:** Es un “driver” de motor (hardware no software), contiene un puente H [2]. Este módulo es necesario para controlar, por medio de señales de baja potencia, motores que el NodeMCU no puede alimentar. Además de cambiar la polaridad en los motores para cambiar la dirección de giro.



Figura 2. Circuito integrado L298N

- **74LS164:** Es un registro de corrimiento que nos permite controlar las luces del carro con 2 pines de entrada.



Figura 3. Registro de corrimiento

- **Motor:** Es el elemento que consume la mayor cantidad de energía en el carro. Se utilizan 2 motores, uno para dar dirección y otro para la tracción. El motor dc trabaja con 2 terminales, si hay una diferencia de potencial entre los terminales el rotor girará en una dirección y si se invierte la polaridad el rotor girará en sentido contrario.



Figura 4. Motor de corriente directa (DC)

- **Sensor de luz:** Circuito que genera o no, una diferencia de tensión dependiendo de la cantidad de luz en el ambiente. Utilizando una fotodiode.



Figura 5. Fotoresistencia

- **Transistor 3906:** Semiconductor que funciona como interruptor. Transistor 2N3906 o equivalentes.

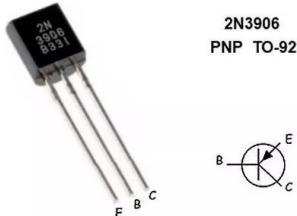


Figura 6. Transistor npn 3906

- **LM7805:** Regulador de tensión. Mantiene la tensión en un valor adecuado para no dañar ningún componente.



Figura 6. LM7805

Conexión de los componentes

Los elementos descritos anteriormente se van a conectar siguiendo el plano electrónico especificado el diagrama 2.

Una vez que compruebe cada componente, ensamble el circuito por módulos en una “protoboard”, se sugiere montarlo en el carro para ejecutar las pruebas del plan de pruebas incluyendo todos los comandos de la lista, tal montaje se muestra en la Foto 3.

Posteriormente se puede trasladar el circuito a una placa preperforada.



Foto 3. Prueba del circuito en protoboard montada en el carro para pruebas.

Antes del traslado se deben planificar las conexiones para realizarlas de forma ordenada, se puede utilizar cable de bus para organizar no solamente el cableado sino también los conectores para los cuales se puede apalancar de DB9 o DB25. Se procede a trasladar a una placa preperforada. Es muy recomendado realizar planos de conexión y pegarlos cerca del circuito como se muestra en la Foto 4.

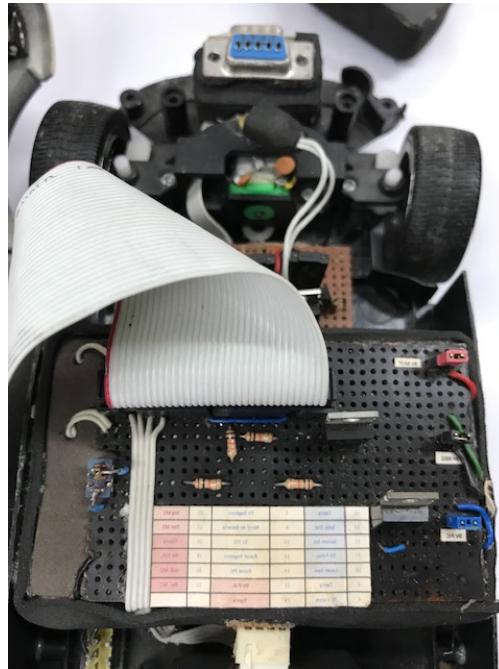


Foto 4. Montaje en placa preperforada (pintada de negro) y uso de cable tipo bus (IDE) para conectar.

Debe ubicar un interruptor para encender o pagar todo el sistema estratégicamente.

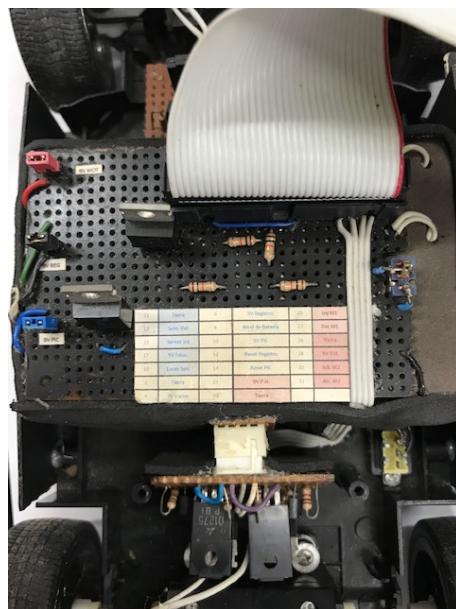
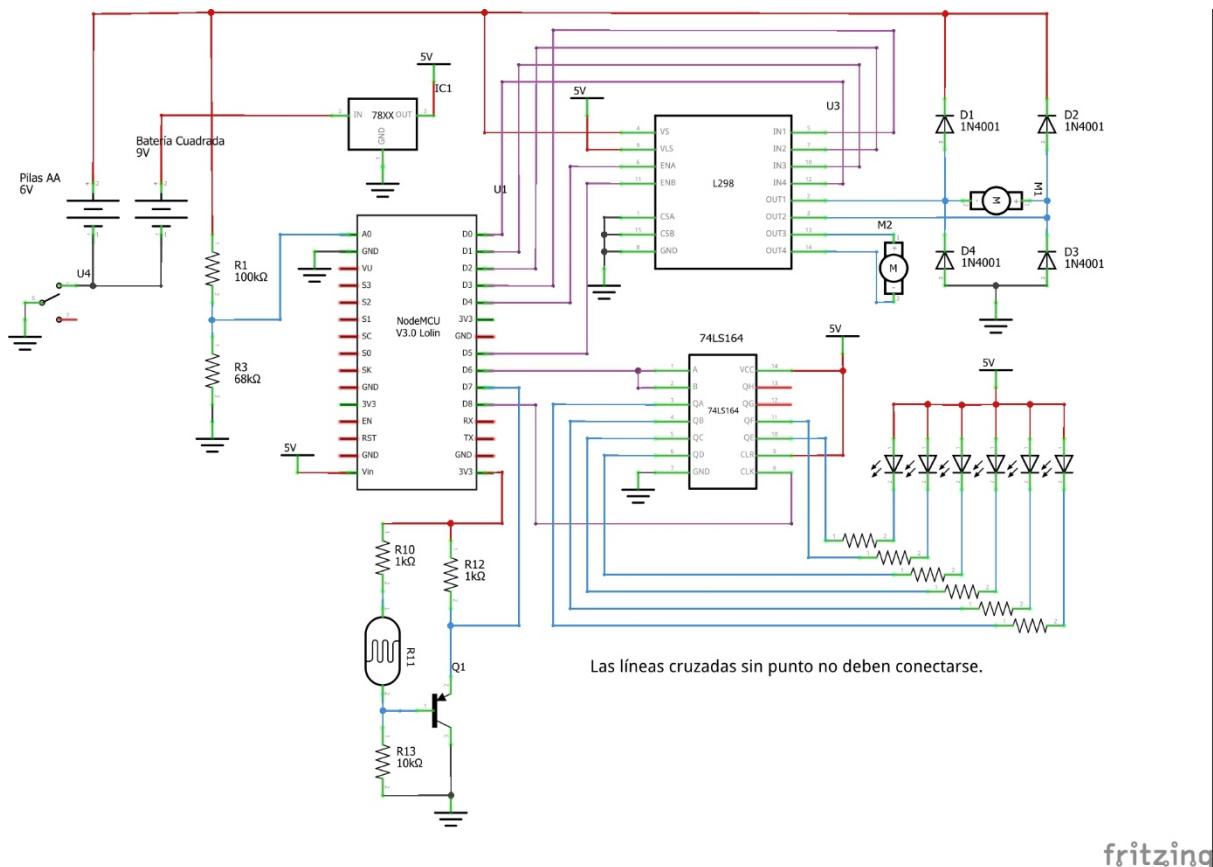


Foto 6. Montaje de los componentes electrónicos en placa preperforada y conexión entre ellos.

El diagrama 2 representa el circuito que se debe implementar.

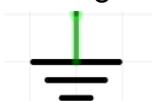


fritzing

Diagrama 2. Circuito final del proyecto.

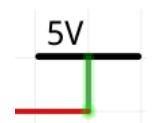
Algunas consideraciones importantes:

1. El circuito debe ser implementado en protoboard y probar que todo funcione, antes de soldar en una tarjeta pre perforada o impresa.
2. Conectar los negativos de la batería, el regulador, los módulos y los circuitos integrados. Para tener una sola referencia, el símbolo 1 representa tierra (GND) que para nuestro circuito será negativo.



Símbolo 1. Representa tierra en un circuito eléctrico.

3. Vcc se refiere a la conexión después de pasar por el regulador de 5v, está representada por este símbolo 2. Físicamente es una conexión igual a las demás, se utiliza con un símbolo para no sobrecargar el diagrama, de igual forma que el símbolo de tierra.



Símbolo 2. Línea de 5 voltios

4. Las conexiones representadas en el diagrama pueden ser puntos de soldadura, cables o puentes. Sin importar el color. Por convención se utiliza rojo para positivo y negro para negativo.
5. Utilizar una placa preperforada como base del circuito para soldar los componentes. Para elementos como LEDs o fotorresistencias, que deben estar expuestos se recomienda usar un cable con varios hilos (tipo BUS) que no se quiebre fácilmente, ver Foto 6.
6. El circuito se alimenta con 5 batería AA de 1.2v ó 1.5v, conectadas en serie para sumar una tensión mayor o igual a los 6v, con suficiente corriente para alimentar a los motores.
7. Utilizar bases para los circuitos integrados y el módulo para que pueden ser reutilizados o reemplazados sin necesidad de desoldar el componente. Así mismo pueden utilizar “headers” hembra para conectar el NodeMCU.

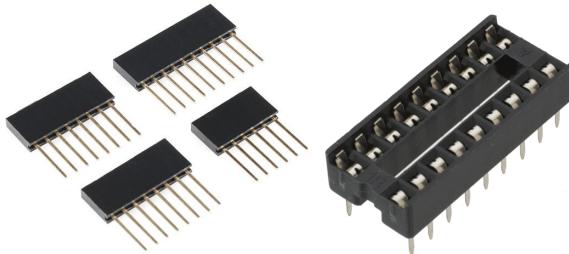


Figura 7 . Headers de diferentes tamaños y base para integrado de 18 pines.

Potencia del carro

Para simular diferentes velocidades en el carro se utilizará modulación por ancho de pulso (pwm por sus siglas en inglés). Lo que se hace es tomar una señal y obtener valores menores por medio de cambios rápidos en el estado de salida.

Si desea conocer más sobre el pwm y su funcionamiento puede consultar la bibliografía [2].

En el caso del lenguaje Arduino, el pwm se expresa con la función “analogWrite”, que recibe un valor entre 0 y 1023 y un pin de salida. El pin de salida del módulo debe soportar este tipo de modulación.

Tabla 1. Comandos a implementar, rango de valores de parámetros y su función.

Formato	Valor	Función
Instrucciones de mando		
pwm	[-1023: min] [min :1023]	Potencia: Envía la señal al auto para avanzar a una potencia determinada, si se envía en 0 el auto deja de enviar energía a los motores. El valor mínimo va a depender de los motores que se utilizan, ronda el valor de 500 en pwm. Si el valor es negativo entonces el auto retrocede.
dir	[-1, 0, 1]	Dirección: decide en qué dirección girar, se envía -1 para ir hacia la izquierda, 0 para seguir directo y 1 para girar a la derecha.
If	[1, 0]	Luces frontales: Control de ambas luces, 1 encender y 0 apagar luces.
lb	[1, 0]	Luces traseras: Se controlan ambas luces, 1 para encender y 0 para apagar
ll	[1, 0]	Direccional Izquierda: Controlar la direccional izquierda, 1 encender y 0 apagar
lr	[1, 0]	Direccional Derecha: Controlar la direccional derecha, 1 encender y 0 apagar
Instrucciones de telemetría		
sense	-----	Retornar el valor de los sensores de la siguiente manera
blvl	[10 - 100]	Battery level Porcentaje de batería restante
ldr	*	(light sensor) Valor del sensor de luz. El valor es interpretado por el desarrollador
Instrucciones de patrón		
Circle	[1,-1]	El carro realiza una vuelta circular y se detiene en el punto de partida
Infinite	-----	El carro debe seguir un recorrido de ∞ u 8 invertido. Termina al llegar al punto de partida
ZigZag	-----	El carro debe seguir un recorrido en zig zag por un tiempo determinado por el desarrollador. Debe apreciarse el movimiento.
especial		Definida por el programador**

Notas:

- El Proyecto es en grupos de máximo 2 personas, es preferible que el grupo se mantenga para el Proyecto III pero no es mandatorio. Se debe entregar el hardware el día Martes 28 de Mayo de 2019 a las 9:30 horas en clase de Taller de debidamente empacado, desconectado con baterías suficientes para el día de la defensa del proyecto, debe incluir documentación escrita (incluir bitácora) y la parte digital (si hay) enviarla ese mismo día a más tardar a las 23:30. Debe entregar todos los componentes de hardware y software necesarios para probar todos los comandos de la Tabla1, de forma que demuestre pleno funcionamiento.
- Debe indicarse un archivo readme.txt con la versión de Python (>=3.3) a utilizar para la revisión, ruta de git y alguna otra indicación que se considere importante. Cada archivo base debe enviarse el nombre construido de la siguiente forma: Iniciales de los nombres (primer integrante) en mayúsculas, guion bajo primer apellido, guion bajo segundo apellido guion bajo Iniciales de los nombres (segundo integrante) en mayúscula, guion bajo primer apellido guion bajo segundo apellido y extensiones txt y doc.
- Cualquier duda, omisión o contradicción en la especificación se debe aclarar con el profesor y se difundirá por el grupo de Facebook.
- Se debe mandar la documentación (archivo .doc) ésta debe contar con lo siguiente: Introducción; Conclusiones; Recomendaciones; Análisis de Resultados, dentro de este punto incluir: Diagramas de Módulos, Plan de Pruebas, Hojas con las Reglas de Grupo, Actividades, Roles y Fechas de Entregables; y Literatura o Fuentes Consultadas.

Además, adjuntar la auto-documentación generada con el método print de Python de los principales módulos de su programa. El resto de la auto-documentación debe estar disponible en la defensa.

- El código debe estar suficientemente documentado de tal forma que los integrantes, se puedan orientar en él fácilmente durante la defensa. Debe contar con auto documentación para ser generada mediante el método print de Python.

- El proyecto se debe defender previa cita con el profesor y el asistente. El profesor o asistente publicará unas fechas de defensa. Ud debe inscribirse para defender el proyecto, de no asistir a la defensa, únicamente obtendrá la nota correspondiente a la documentación del proyecto.
- Las funciones de repetición dentro de la lógica del programa cuya naturaleza puede ser resuelta recursivamente, no se aceptará con iteración. Desviarse de ésta directriz implicará la anulación de esa parte del código mediante comentarios en él y su correspondiente pérdida de puntos.
- Cualquier clase de copia de código será sancionada de acuerdo con el reglamento vigente y se llevará hasta la consecuencia de amonestación con carta al expediente. Código adoptado para el manejo de interfaz debe especificarse claramente la fuente y reconocer los créditos. Está prohibida la copia de código que involucre la solución lógica general del algoritmo.
- Si en la defensa no demuestra su autoría con el dominio propio de esa calidad solo se le otorgarán los puntos correspondientes a los obtenidos en la documentación.
- Las hojas con los rubros de evaluación serán suministradas por cada profesor

Código

Enlace: https://github.com/santigr17/FormulaE_CE_TEC-.git

En el repositorio se encuentra un proyecto con 3 archivos:

- TelemetryLog.py
Código de ejemplo para utilizar el WiFiClient. Consiste en una consola para enviar mensajes al servidor. Muestra el resultado de cada mensaje.
- WiFiClient.py
Código del cliente para hacer la conexión con el servidor, envía y recibe mensajes.
- NodeMCU/NodeMCU.ino (**Incompleto**)
Código para controlar el módulo NodeMCU. Este código consiste en el servidor que recibe y envía mensajes de un cliente. Esta funcionalidad está completa.
Se debe completar las acciones para mover los motores, controlar las luces y los sensores.

Literatura y Fuentes Consultadas

- [1] del Valle Hernández, L. (2019). NodeMCU y el IoT tutorial paso a paso desde cero. Recuperado de <https://programarfacil.com/podcast/nodemcu-tutorial-paso-a-paso/>
- [2] (2019). Recuperado de
https://www.sparkfun.com/datasheets/Robotics/L298_H_Bridge.pdf
- [3] Crespo, E. (2016). NodeMCU y el IoT tutorial paso a paso desde cero. Recuperado de <https://programarfacil.com/podcast/nodemcu-tutorial-paso-a-paso/>
- [4] (2019). Recuperado <https://goes.gsfc.nasa.gov/text/databook/section09.pdf>
- [5] <https://www.mclaren.com/formula1/team/a-brief-history-of-computing-in-F1-1052199/>. Recuperado 31 de Marzo 2019.
- [6] . Recuperado 31 de Marzo 2019 <https://www.fiaformulae.com/en/discover/history>