

# Gilbert your Robo-Friend

## TC2037: Implementación de Métodos Computacionales

Rodrigo López Guerra  
Tecnológico de Monterrey, Campus Estado de  
Puebla  
A01737437@tec.mx

Valter Alejandro Kuhne Hernández  
Tecnológico de Monterrey, Campus Estado de  
Santa Fé  
A01379392@tec.mx

Ximena Cantera Reséndiz  
Tecnológico de Monterrey, Campus Estado de  
Querétaro  
A01277310@tec.mx

### Resumen

*El informe describe el desarrollo de un sistema para robots autónomos, enfocado en comprender y ejecutar peticiones en el lenguaje natural. Se hace uso de un Analizador Léxico y un Intérprete de Sintaxis para procesar las instrucciones dadas por el usuario. Se detalla la definición del lenguaje, incluyendo tokens y gramática, junto con algunos ejemplos de entrada. Se habla sobre la importancia de la precisión al interpretar los comandos y se discuten las conclusiones grupales e individuales del equipo.*

**Palabras clave:** Robot, Lenguaje, Autónomo, Analizador, Sintaxis, Gramática, Léxico, Intérprete, Token, Determinantes, Linux, LEX, YACC, C, PYTHON.

### 1 Introducción

En el campo del desarrollo de robots autónomos, la capacidad de comprender y ejecutar órdenes se puede convertir en un gran desafío. La demanda actual de robots capaces de interactuar de manera intuitiva con los usuarios ha creado la necesidad de diseñar sistemas que puedan interpretar una amplia variedad de expresiones lingüísticas. Este requisito ha llevado a los ingenieros a enfocarse en una tarea: “Crear un sistema que pueda procesar el diverso lenguaje humano, desde peticiones simples, como las más elaboradas.”

Como solución, se desarrolló un sistema compuesto por un Analizador Léxico y un Intérprete de Sintaxis, que al trabajar juntos, son capaces de analizar y comprender las peticiones proporcionadas por los usuarios. De esta forma es posible mantener una comunicación más natural entre los usuarios y los robots, mejorando la experiencia al momento de interactuar.

Con este contexto, el trabajo actual tiene como objetivo presentar una vista más detallada de un diseño e implementación de un sistema de control de movimiento, destacando la importancia en el mundo actual sobre la relación hombre-máquina. También se colocan ejemplos de comandos en el lenguaje humano y se analiza el procesamiento que hace el sistema, demostrando la capacidad de comprensión en una variedad de expresiones lingüísticas.

El proyecto tiene como intención el explorar y evaluar la efectividad de este tipo de sistemas en diferentes formas de usos. Se espera que los resultados obtenidos den un entendimiento mayor al que se tenía al iniciarlo con las capacidades y limitaciones del sistema. Además, se espera que el proyecto de una visión diferente que permita mejorar el avance del desarrollo de robots autónomos, siendo más inteligentes y capaces de interactuar de manera más natural y eficiente con los humanos en una amplia variedad de situaciones.

## 2 Descripción del problema

La industria poco a poco se está buscando automatizar procesos que pueden ser tardados o que incrementan el costo de producción, es por ello que es importante el desarrollo de sistemas que nos puedan ayudar a la realización de tareas, para así poder optimizar los procesos de producción y atender la demanda de la población. Con ello, dentro del ámbito de la computación, surgen las alternativas de inteligencia artificial básicas, donde el ser humano puede controlar a robots autónomos para poder desarrollar dichas tareas, teniendo un mayor rendimiento a la hora de trabajar y un mejor desempeño de los empleadores cuando se trata de la toma de decisiones colectivas, pasando a ser productores de manufactura a desarrolladores de ideas.

Las propuestas muchas veces se veían imposibles gracias a la enorme cantidad de conocimiento que necesitaba un robot en orden de realizar las tareas de maneras precisas y correctas, y con ello surgieron las gramáticas libres de contexto y los robots autónomos. En el desarrollo de robots autónomos, existe la necesidad de diseñar un sistema de control de movimiento capaz de interpretar comandos en lenguaje natural para guiarlos en sus acciones. La complejidad cae en la variedad de expresiones y estructuras que pueden utilizar los usuarios al comunicarse con estos.

Un punto para destacar es el garantizar la precisión en interpretar los comandos que provienen de una oración dada por el usuario. Se deben de entender las instrucciones, incluso en situaciones que pueda haber ambigüedad en la forma de expresar los comandos. El sistema debe comprender la diversidad lingüística y tener la capacidad de procesar y ejecutar comandos de manera confiable en la variedad de contextos.

Los analizadores léxicos y sintácticos permiten una comunicación correcta entre máquina y humano, así pudiendo tener mejores resultados a la hora de realizar las tareas, buscando ser personalizables a lo largo del tiempo y sobre todo funcionales dentro de las variedades de tareas a las que se le puede asignar a un robot. La eficiencia que estos procesos de automatización traen a la industria, nos permiten desarrollar proyectos en los cuales nosotros podamos cuidar estos aspectos y complejidades para el uso cotidiano y correcto de los mismos, creando una coexistencia tanto digital como física para la mejora de la industria.

### 2.1 Descripción de la solución

El programa es desarrollado en dos partes principales: un Analizador Léxico (Lex) y un Intérprete de Sintaxis (YACC). El Analizador Léxico se encarga de dividir el texto de entrada en tokens, como verbos, números, y signos de puntuación, mientras que el Intérprete de Sintaxis define reglas gramaticales para interpretar las frases construidas a partir de esos tokens.

Una vez que el Analizador ha dividido el texto en tokens, el Intérprete entra para analizar la estructura gramatical de las frases. Esto es fundamental para comprender el significado de las instrucciones y traducirlas en acciones que el Robot Gilbert pueda realizar. El uso de reglas gramaticales se necesitan claras y precisas, para que el Intérprete garantice una interpretación coherente, lo que permite a Gilbert ejecutar las tareas provenientes de la oración dada por el usuario.

El alfabeto que puede llegar a entender nuestro robot es el suficiente para poder desarrollar las tareas cotidianas suficientes para satisfacer las necesidades de las mismas, buscando así poder tener una conexión entre robot y humano a través de la comunicación asertiva y correcta para la realización de acciones y actividades.

Con esta solución, podemos tener un robot el cual se pueda mover de manera correcta en un ambiente físico, en donde se le puede dar uso de mesero, transportador, e inclusive de herramienta de educación didáctica, teniendo muchos usos en la industria. Actualmente nuestro robot solamente se puede mover dentro de un espacio virtual simulado por el mismo CPU, sin embargo se busca que el robot pueda ser implementado en un ambiente físico, para que pueda ser servible dentro de la industria.

### 3 Proceso de interpretación del CPU

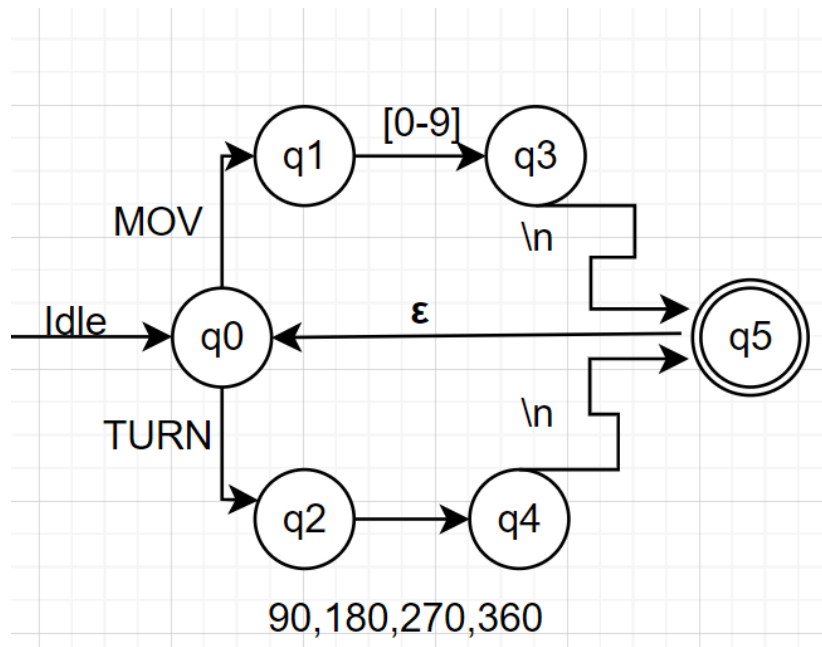
#### 3.1 Desarrollo del CPU

El Central Processing Unit (CPU) se utiliza en varias etapas dentro de la interpretación de comandos por parte del diseño para controlar al robot. Durante la fase de Analizador Léxico, el CPU interpreta el análisis léxico de la entrada, dividiéndolo en tokens como verbos, números y signos de puntuación. Para realizarlo, implica identificar y clasificar los componentes declarados en el código, siguiendo con procesar cada caracteres del texto de entrada para determinar su categoría correspondiente.

En la siguiente etapa, el Intérprete de Sintaxis continúa definiendo las reglas gramaticales para interpretar las oraciones construidas a partir de los tokens dados por el Analizador Léxico. En esta parte, el CPU procesa esas reglas y aplica los algoritmos de análisis para comprender la escritura y el significado de la petición dada como texto de entrada.

Finalmente, el CPU envía las instrucciones interpretadas. De ser necesario, las instrucciones se convierten en comandos que controlan los motores y otros componentes del robot, permitiendo realizar las tareas o peticiones indicadas por el usuario.

#### 3.2 Análisis del CPU



**Figura 1.1.** Autómata determinístico finito (ADF) para el CPU

Este Autómata determinístico finito (ADF) trabaja sobre la CPU utilizado para el movimiento del robot sobre un plano de dos dimensiones donde el comando recibido puede ser “MOV” o “TURN” que trabajan como argumentos guardados en un archivo “amd.” que recibirá el programa en python que generará una matriz de 10x10 unidades. Lo que demuestra el autómata es justamente como el programa recibe instrucciones de movimiento y dirección del input y las cantidades de unidades a moverse o cambiar de dirección que llevarán al final

## 4 Definición de lenguaje

### 4.1 Tokens de Lex

Los tokens son unidades léxicas básicas que representan elementos significativos del código fuente. Cada token representa una categoría gramatical específica, como palabras clave, símbolos de puntuación, números, identificadores, etc. Estos tokens son utilizados por el analizador léxico (o lexer) para identificar y clasificar los componentes del código fuente durante la fase de análisis.

**ROBOT:** Representa la palabra *"Robot"*, que indica el sujeto de una acción, así como Gilbert y Gentleman.

**PETITION:** Este token representa las frases *"will you"*, *"would you"*, *"can you"* y *"could you"* que indica una solicitud o pedido de acción.

**KINDLY:** Este token representa la palabra *"pretty"*, que puede ser parte de una frase verbal en la gramática, indicando cortesía o educación al hacer una solicitud.

**ASKING:** Representa la palabra *"please"*, *"mind to"* y *"kindly"*, que pueden ser parte de una frase nominal en la gramática, indicando la acción de pedir algo amablemente.

**MOVE, TURN, PROJECT:** Estos tokens representan palabras que son parte de una frase verbal en la gramática. El token **"MOVE"** representa la palabra *"move"*, que indica la acción de moverse. El token **"TURN"** representa la palabra *"turn"*, que indica la acción de girarse en una dirección específica. El token **"PROJECT"** representa la palabra *"project"*, que indica la acción de proyectar el recorrido hasta el momento.

**DEGREES:** Este token representa la palabra *"degrees"* o *"°"*, que puede ser utilizada para indicar la medida en grados de una dirección de giro numérico en la gramática.

**DIREC\_LEFT, DIREC\_RIGHT, DIREC\_AROUND:** Estos tokens representan direcciones específicas (*"left"*, *"right"*, *"around"*), que pueden ser utilizadas para indicar la dirección de giro con adverbios en la gramática.

**DIREC\_NUM:** Estos tokens representan direcciones específicas en grados (*90*, *180*, *270*, *360*), que pueden ser utilizadas para indicar la dirección de giro numérico en la gramática.

**SINGLE, NUM:** El token **"SINGLE"** representa una unidad, que puede ser utilizada para indicar una cantidad de un elemento en la gramática. El token **"NUM"** representa un número de unidades, que puede ser utilizado para indicar una cantidad específica de un elemento en la gramática.

**PLU\_BLOCKS, SING\_BLOCKS:** El token **"PLU\_BLOCKS"** representa la palabra *"blocks"* o *"spaces"*, que puede ser utilizada para indicar múltiples bloques en la gramática. El token **"SING\_BLOCKS"** representa la palabra *"block"* o *"space"*, que puede ser utilizada para indicar un solo bloque en la gramática.

**CONJ\_THEN, CONJ\_ALSO, CONJ\_AND:** El token **"CONJ\_THEN"** representa la conjunción *"then"*, que puede ser utilizada para indicar una secuencia de acciones en la gramática. El token **"CONJ\_ALSO"** representa la conjunción *"Also"*, que puede ser utilizada para indicar una acción adicional en la gramática, también indica la secuencia de más oraciones. El token **"CONJ\_AND"** representa la conjunción *"and"*, que puede ser utilizada para indicar una conjunción entre acciones en la gramática.

**FRONT, BACK:** El token **"FRONT"** representa la palabra *"front"*, que puede ser utilizada para indicar la dirección hacia adelante en la gramática. El token **"BACK"** representa la palabra *"back"*, que puede ser utilizada para indicar la dirección hacia atrás en la gramática.

**COMMA, PERIOD, QUEST, EXCLA:** El token **"COMMA"** representa el símbolo de coma *","*, que se utiliza para separar elementos en una lista o frase en la gramática. El token **"PERIOD"** representa el símbolo de punto *"."*, que se utiliza para indicar el final de una oración en la gramática. El token **"QUEST"** representa el símbolo de interrogación *"?"*, que se utiliza para indicar una pregunta en la gramática. El token **"EXCLA"** representa el símbolo de exclamación *"!"*, que se utiliza para indicar una exclamación o enfatizar una oración en la gramática.

**EOL:** Representa el final de línea, que indica el final de una línea en el código fuente.

**OTHER:** Representa cualquier otro elemento léxico que no se ajusta a ninguno de los tokens anteriores y/o espacio

## 4.2 Gramática Libre de Contexto (CFG)

La gramática describe las instrucciones para controlar al robot, estas se encuentran divididas en categorías como el sujeto (NOUN\_PHRASE) y acción (VERB), esta gramática nos permite construir comandos que sean comprensibles para guiar las acciones de Gilbert, desde movimientos simples hasta tareas más complejas como realizar un giro de forma precisa o una proyección del camino recorrido.

**INPUT → SENTENCE:** Esta regla indica que un INPUT está compuesto por una SENTENCE. Esto es similar a decir que una expresión está compuesta por una serie de términos que forman una oración que el robot puede entender.

**INPUT → INPUT error EOL:** Esta regla indica cuando el token OTHERS no se encuentra dentro de las normas establecidas.

**INPUT → € | EOL:** Esta regla indica que se va a ignorar cualquier nueva línea o INPUT que se encuentre vacío.

**SENTENCE → NOUN\_PHRASE VERB\_PHRASE:** Una SENTENCE está compuesta por un NOUN\_PHRASE seguido de un VERB\_PHRASE. Esto quiere decir que una expresión aritmética está compuesta por un sujeto (NOUN\_PHRASE) seguido de un verbo (VERB\_PHRASE).

**VERB\_PHRASE → VERB SIGN | VERB CONJ\_PHRASE:** Un VERB\_PHRASE puede ser un VERB seguido de un SIGN (signo de puntuación) o un VERB seguido de una CONJ\_PHRASE. Esto es similar a decir que un verbo puede estar seguido de un signo de puntuación o frases, lo cual podría ser una serie de acciones o instrucciones adicionales.

**NOUN\_PHRASE → ROBOT ASK | ROBOT COMMA ASK:** Un NOUN\_PHRASE puede ser simplemente "ROBOT ASK" o "ROBOT COMMA ASK". Esto indica que un sujeto complejo puede ser una petición directa al robot, posiblemente con una coma para separar.

**ASK → KINDLY ASKING | ASKING | PETITION KINDLY ASKING | PETITION ASKING:** La regla ASK define varias formas de petición al robot. Podría ser simplemente "ASKING" o "KINDLY ASKING", o podría incluir la palabra "PETITION" para indicar que es una petición más formal.

**VERB → MOVING | TURNING | PROJECT:** Un VERB puede ser "MOVING", "TURNING" o el token "PROJECT". Esto muestra acciones que podría ser un movimiento, una rotación o una proyección.

**MOVING → MOVE BLOCKS | MOVE | MOVE BACK | MOVE FRONT | MOVE BLOCKS BACK | MOVE BLOCKS FRONT | MOVE FRONT BLOCKS | MOVE BACK BLOCKS:** La regla MOVING define varias formas de movimiento. Puede ser simplemente "MOVE", "MOVE BACK", "MOVE FRONT", o puede incluir un número de bloques (BLOCKS) a mover. "BACK" y "FRONT" indican las direcciones posibles para dirigir el robot.

**TURNING → TURN DIRECTION | TURN:** TURNING puede ser simplemente "TURN" o "TURN" seguido de una dirección (DIRECTION), como "LEFT" o "RIGHT".

**BLOCKS → SINGLE SING\_BLOCKS | NUM PLU\_BLOCKS | SINGLE | NUM:** BLOCKS puede ser un solo bloque (SINGLE) o un número (NUM) seguido de "BLOCKS" en singular (SING\_BLOCKS) o plural (PLU\_BLOCKS).

**DIRECTION → DIREC\_RIGHT | DIREC\_AROUND | DIREC\_LEFT | DIREC\_NUM DEGREES:** La regla DIRECTION especifica las posibles direcciones en las que puede girar el robot, incluyendo giros específicos de 90, 180, 270 o 360 grados.

**CONJ\_PHRASE → COMMA CONJ\_THEN VERB\_PHRASE | COMMA CONJ\_AND VERB\_PHRASE | COMMA CONJ\_AND CONJ\_THEN VERB\_PHRASE | SIGN CONJ\_ALSO COMMA ASK**

**VERB\_PHRASE | SIGN CONJ\_ALSO COMMA VERB\_PHRASE | COMMA VERB\_PHRASE:** La regla CONJ\_PHRASE define varias formas de conjunciones de frases, lo que permite la unión de varias instrucciones.

**SIGN → PERIOD | EXCLA | QUEST:** SIGN puede ser un punto, un signo de exclamación o un signo de interrogación, que marca el final de una oración o frase.

## 5 Entradas de muestra

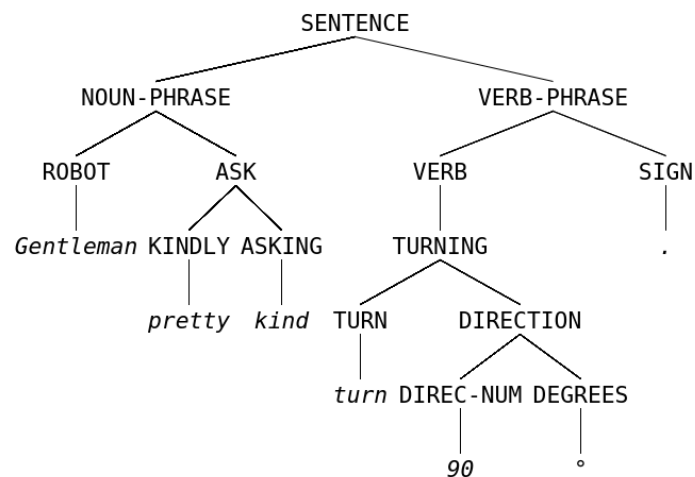
### 5.1 Tabla de Ejemplos

	Entrada	Descripción
CORRECTO	Robot, please move 3.	Es el ejemplo más simple de una entrada correcta, ya que inicia con el sustantivo, sigue con la petición, después la acción con el complemento correcto.
	Gentleman, pretty kindly move 3 blocks back, and then turn 360 degrees?	Esta frase es educada y detallada. Comienza con un sustantivo (“Gentleman”), seguido de una petición amable (“pretty kindly move 3 blocks back”) y finalmente una instrucción adicional (“and then turn 360 degrees”). En general, es una solicitud clara y educada dada, siguiendo una de las estructuras declaradas.
	Robot, will you pretty please move 3 blocks forward?	Esta es una solicitud directa y educada. Comienza con el sustantivo (“Robot”), seguido de una petición amable (“will you pretty please move 3 blocks forward”), complementando en un orden correcto.
	Gilbert, pretty please move back, and turn 270 degrees.	Aquí, se menciona el nombre especificado (“Gilbert”). La solicitud es educada (“pretty please move back”) y se agrega una instrucción adicional (“and turn 270 degrees”). En general, es una solicitud clara y sigue con una de las reglas establecidas.
	Robot, would you pretty mind to turn around, move 3 blocks ahead, and then project!	Esta solicitud comienza con el sustantivo (“Robot”) y utiliza la expresión “would you pretty mind” para ser educada. Luego se dan instrucciones adicionales (“turn around, move 3 blocks ahead, and then project”). Aunque es un poco más detallada, sigue siendo comprensible.
INCORRECTO	Gilbert please move 2 blocks quick!	La oración es incorrecta ya que la palabra “quick” no se encuentra dentro del alfabeto. La forma correcta sería: “Gilbert please move 2 blocks!”
	Moves Robot please 2 blocks, and turn 89 degrees.	El orden de las palabras es incorrecto, y debe de colocarse: “Robot please move 2 blocks, and turn 89 degrees.” Cabe recalcar que “89” no es parte de los grados establecidos.

	Robot, could you pretty please left turn?	Gramaticalmente la pregunta es incorrecta y no tiene sentido. La oración acomodada correctamente es: “Robot, could you pretty please turn left?”
	Gentleman would you pretty please turn ahead?	La solicitud es incorrecta, ya que a pesar de ser formal, se ocupa “ahead” que es incorrecto al usar el verbo “turn”. La forma correcta es: “Gentleman would you pretty please turn 270 degrees?”
	Gentleman, would you pretty please turn tree.	La petición es incoherente, ya que al colocar un “turn” se espera que le siga una dirección, ya sea en número o como punto cardinal. La forma correcta es “Gentleman, would you pretty please turn tree.”

**Tabla 1.1.** Inputs aceptados y negados.

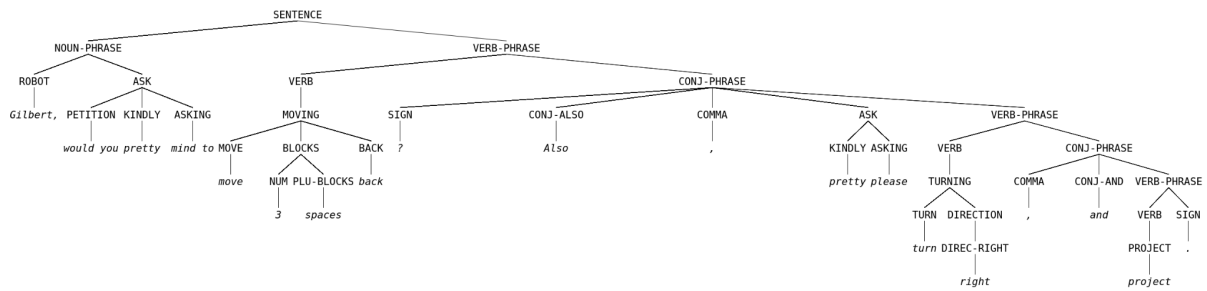
## 5.2 Árboles de Análisis Sintáctico



**Figura 2.1.** Árbol de análisis sintáctico para el input “Gentleman pretty kind turn 90°.”.

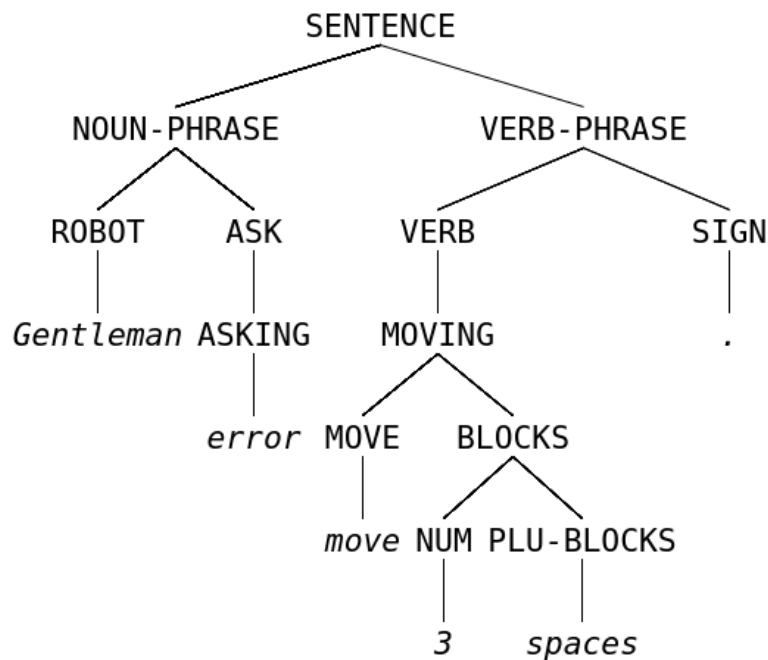
El árbol de análisis de la **Figura 2.1** es correspondiente a una entrada aceptada con una oración simple. Durante el proceso de análisis se identifica al sustantivo “*Gentleman*”, siguiendo con la petición formal “*pretty kind*”. Siguiendo con el VERB\_PHRASE en el orden correcto, dándole continuidad con el VERB “*turn*” y finalmente colocando un número de grados ya definidos como token.





**Figura 2.2.** Árbol de análisis sintáctico para el input “Gilbert, would you pretty mind to move 3 spaces back? Also, pretty please turn right, and project.”.

En la **Figura 2.2** hace referencia a el proceso de análisis a una entrada correcta, colocando una petición más extensa y por ello, más compleja. Comienza con el sujeto (NOUN\_PHRASE) “Gilbert”, seguido de la petición formal de “would you pretty mind to”. Seguido de esto, se indica la cantidad de espacios a mover con el número “3” seguido de la palabra “spaces” que especifica la unidad de medida. En este caso se encuentra la conjunción “Also” para indicar que se encuentra una acción extra. La acción solicita una petición formal de un “pretty please” un “turn right”. Finalmente, la frase termina con la acción “project”, con el objetivo de proyectar el camino recorrido hasta el momento.



**Figura 2.3.** Árbol de análisis sintáctico para el input “Gentleman move 3 spaces.”.

En la **Figura 2.3** hace referencia a el proceso de análisis a una entrada incorrecta, colocando una petición sin el token “ASKING”. El árbol representa la entrada “Gentleman move 3 spaces.” con base a las reglas definidas en la gramática. Se identifica la palabra “Gentleman” que viene siendo parte del NOUN\_PHRASE, pero con la falta de una petición ASK por lo que ya se genera un error automático con la entrada. Y la respuesta del robot es mostrar un mensaje indicando que no le es posible entender la petición de entrada.

## 6 Industria actual

El uso de los sistemas de control de movimiento ha revolucionado la interacción entre humanos y tecnología en muchas áreas de la industria. Estos sistemas permiten a los usuarios comunicarse con dispositivos de manera intuitiva, utilizando un lenguaje cotidiano para dar las instrucciones y poder realizar tareas específicas. Esta capacidad no solo mejora la eficiencia operativa, también la accesibilidad y la usabilidad de la tecnología, dando paso a nuevas oportunidades en la industria y el hogar. Algunas de las áreas donde estos sistemas son útiles son:

- **Robótica:** En las áreas de fabricación y ensamblaje, los robots autónomos pueden recibir comandos en lenguaje natural para realizar tareas, como mover materiales, ensamblar productos o inspeccionar la calidad de los productos. Esto facilita la reconfiguración de los robots para adaptarse a diferentes procesos de producción. Empresas como ABB Robotics, Fanuc y KUKA Robotics utilizan este sistema.
- **Logística y almacenes automatizados:** Los robots que son capaces de moverse pueden recibir instrucciones verbales para realizar tareas de recolección, ensamblaje y distribuir productos. Esto permite una gestión eficiente en el inventario y una respuesta ágil a las demandas del mercado. Amazon es un ejemplo de una empresa que utiliza robots móviles en sus centros de distribución en el inventario, recolectar y empaquetar los pedidos de manera automatizada.
- **Asistentes en el hogar:** Alexa, desarrollado por Amazon, es capaz de interpretar comandos de voz en lenguaje natural para realizar una variedad de tareas, como reproducir música, controlar dispositivos domésticos inteligentes, y muchas más. Estos asistentes están diseñados para ayudar en la automatización de tareas diarias y mejorar la experiencia del usuario en el hogar mediante una interacción verbal.
- **Servicios de atención al cliente:** Los Chatbots son capaces de procesar el lenguaje para comprender las consultas y solicitudes de los clientes, decidiendo si pueden dar respuesta automática o si es necesario dirigir las consultas a agentes humanos. Esto mejora la satisfacción del cliente en atención médica, servicios financieros o correos electrónicos.

## **7 Links**

### **7.1 GitHub**

<https://github.com/iiRoy/Gilbert-Your-RoboFriend.git>

### **7.2 Ejecución y pruebas**

<https://drive.google.com/file/d/1P2xFf7P5aoOS-UMNCDPCVV66XgtIhqFP/view?usp=sharing>

### **7.3 Presentación**

<https://1drv.ms/p/s!AofjNkBI15dxitoi6D3l2dN8S-jv5Q?e=Et9viw>

## 8 Conclusiones

### 8.1 Individual

**Rodrigo:** En este proyecto pude desarrollar de mejor manera lo que fue el conocimiento de las gramáticas libres de contexto, además de conocer el trasfondo de todo el proceso sintáctico y léxico que los compiladores muchas veces procesan. Del mismo modo, pude reforzar conocimientos dentro de lo que es mi conocimiento de programación, haciendo uso de listas dinámicas para la realización de procesos dentro de nuestro proyecto.

Pude entender de manera más detallada los diversos procesos que existen alrededor de un analizador gramatical, además de poder adaptarlo a mis futuros códigos para poder desarrollar habilidades para la programación continua. Las herramientas como lo son GitHub, LEX, YACC y Python, ayudan a poder desarrollar proyectos de largo impacto en corto tiempo, así teniendo un periodo de mejora continua, buscando ser siempre mejores a la hora de realizar un vistazo detallado, como lo son los árboles de análisis sintáctico o los diagramas de autómatas determinísticos finitos y no determinísticos finitos.

**Valter:** Este proyecto da resultados, tanto en el CV como en la vida, un robot que sigue órdenes y que seamos capaces de usar CFG para describir lo que hace es una gran herramienta para el futuro. El autómata funciona como una nueva herramienta para la visualización de programas complejos y sencillos como el CPU usado en este caso. Además, este proyecto me enseñó a desarrollar habilidades aprendidas tras observar cómo funciona el robot y como todo se puede combinar para crearlo. Además de apoyar con lo que pude en el proceso de crear los tokens y detalles en cada reunión que tuvimos. Finalmente agregamos una implementación de Python de la que me encargue para proyectar el código.

**Ximena:** Para mi, el desarrollo de este tipo de sistemas es esencial si se quiere llegar a una “era” automatizada. Estos avances no solo mejoran el realizar las tareas cotidianas, incluso pueden transformar en el área industrial. Pero es importante desarrollarlo con un enfoque multidisciplinario y ético para garantizar que la sociedad lo use de forma responsable y con el objetivo de mejorar. Finalmente, considero impresionante el potencial que tiene esta tecnología para mejorar la calidad de vida y la orientación de este proyecto para fomentar la colaboración armoniosa entre humanos y máquinas en un futuro no muy lejano.

### 8.2 Grupal

Dentro del desarrollo de este proyecto, pudimos resaltar la importancia de los autómatas dentro del análisis de unidades de procesamiento para la comprensión de gramáticas libres de contexto. De la misma forma, pudimos tener un mayor entendimiento de la visualización efectiva de dichas gramáticas a través de recursos gráficos como árboles de analizadores gramaticales.

Durante la realización del proyecto, tuvimos la oportunidad de conocer el trasfondo a más detalle de los diversos asistentes virtuales mediante sus alfabetos, variables y reglas sintácticas para el desarrollo de nuestro propio analizador léxico y compilador, así dándole una oportunidad a las actualizaciones para realizar las actividades del día a día.

Como equipo, aprendimos que una buena organización para la obtención de metas en conjunto logra un resultado efectivo en el avance de las actividades en diversos entornos digitales.

## Referencias

- [1] Bikaina. Descubre qué son y cómo funcionan los robots autónomos - guía paso a paso - RoboticaHoy.com. RoboticaHoy. <https://roboticahoy.com/que-son-y-como-funcionan-los-robots-autonomos/>.
- [2] Chakraborty, E. Robotics and Autonomous Systems: 13 Important facts - LAMBDAGEEKS. Lambda Geeks. <https://es.lambdageeks.com/robotics-and-autonomous-systems/>.
- [3] Admin. Robots autonomos. Ripipsa Cobots. <https://ripipsacobots.com/robots-autonomos/>.
- [4] Cordis, Cordis. Europa. E. *¿Los robots promueven o dificultan el desarrollo sostenible?*. CORDIS | European Commission. <https://cordis.europa.eu/article/id/436613-are-robots-helping-or-hindering-sustainable-development/es>.
- [5] Kamila. Lex (generador de analizadores léxicos). TechEdu. <https://techlib.net/techedu/lex-generador-de-analizadores-lexicos/>.
- [6] Esneca. Automatización industrial: sus avances y ventajas para empresas. Esneca. <https://www.esneca.com/blog/automatizacion-industrial-avances/>.