



**Área de Ingeniería en Computadores**

**Lenguajes, Compiladores e Intérpretes**

**Tarea Programada II**

**CallCenterLog**

**Profesor**

Marco Rivera Menénes

**Estudiantes**

Gabriel Chacón Alfaro - 2021049454 [gchacon8@estudiantec.cr](mailto:gchacon8@estudiantec.cr)

Emanuel Marín Gutiérrez - 2019067500 [emarin702740530@estudiantec.cr](mailto:emarin702740530@estudiantec.cr)

Jose Andres Rodriguez Rojas - 2019279722 [joseandres216@estudiantec.cr](mailto:joseandres216@estudiantec.cr)

**Fecha**

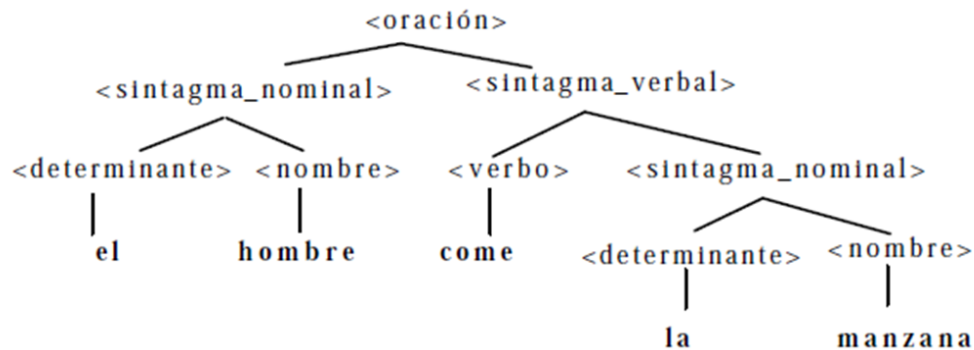
**27 de marzo, 2023**

## **Contenido**

- 1.1 Descripción de los algoritmos desarrollados
- 1.2 Descripción de las funciones implementadas
- 1.3 Descripción de las estructuras de datos utilizadas
- 1.4 Problemas sin solución
- 1.5 Problemas encontrados
- 1.6 Plan de actividades
- 1.7 Conclusiones
- 1.8 Recomendaciones
- 1.9 Bibliografía
- 1.10 Bitácora

## 1.1 Descripción de los algoritmos de solución desarrollados

Para este caso, como se intenta generar una comunicación eficiente con el usuario, se implementó una estructura gramática BNF, la cual nos ayuda a facilitar la detección de oraciones reales junto a nuestra base de datos. Comúnmente la gramática BNF se conoce por estar compuesta por un sintagma nominal y un sintagma verbal. Como se ve a continuación:



Pero para nuestro caso se agregaron diversas posibilidades más, ya que esta estructura anterior limitaría la detección de muchas otras oraciones. A continuación podemos ver distintas estructuras que abren paso a otras posibles oraciones con más componentes gramaticales como adverbios, adjetivos e interjecciones.

```

oracion --> sintagma_nominal(N), sintagma_verbal(N).
oracion --> interjeccion.
oracion --> interjeccion, interjeccion.
oracion --> interjeccion, interjeccion, interjeccion.
oracion --> interjeccion, sintagma_nominal(_).
oracion --> interjeccion, sintagma_nominal(_), sintagma_nominal(N),
sintagma_verbal(N).
oracion --> interjeccion, sintagma_nominal(N), sintagma_verbal(N).
oracion --> adverbio(_, _).
oracion --> adverbio(_, _), adverbio(_, _).
oracion --> adverbio(_, _), sintagma_nominal(N), sintagma_verbal(N).
  
```

De esta forma abrimos más la gama de oraciones que detectamos. Pero al mismo tiempo dentro de los sintagmas expandimos la cantidad de combinaciones y

componentes gramaticales que se pueden encontrar. Por ejemplo, en esta imagen se pueden apreciar todas las combinaciones agregadas de un sintagma nominal.

```
sintagma_nominal(N) --> vacio(N).
sintagma_nominal(N) --> pronombre(_, N).
sintagma_nominal(N) --> sustantivo(_, N).
sintagma_nominal(N) --> pronombre(G, N), sustantivo(G, N).
sintagma_nominal(N) --> preposiciones, pronombre(_, N).
sintagma_nominal(N) --> preposiciones, sustantivo(_, N).
sintagma_nominal(N) --> preposiciones, sustantivo(_, N), preposiciones, sustantivo(_, N).
sintagma_nominal(N) --> determinante(G, N), sustantivo(G, N).
```

Así, en las oraciones un sintagma nominal puede cubrir más posibilidades. Esto mismo ocurre para los sintagmas verbales.

Como podemos observar en la estructura del sintagma nominal anterior, existen varios componentes dentro de este sintagma. Estos distintos componentes como pronombres, sustantivos y preposiciones se validan con los distintos pronombres, sustantivos, etc, que se agregaron en el mismo archivo de gramática. Como podemos observar adelante:

```
sustantivo(G, N) --> [P], {es_sustantivo(P, G, N), sustantivo(pc, _, singular).
es_sustantivo(pcs, _, plural).
es_sustantivo(laptop, femenino, singular).
es_sustantivo(laptops, femenino, plural).
es_sustantivo(computador, masculino, singular).
es_sustantivo(computadora, femenino, singular).
es_sustantivo(computadoras, femenino, plural).
es_sustantivo(impresora, femenino, singular).
es_sustantivo(impresoras, femenino, plural).
es_sustantivo(mouse, masculino, singular).
es_sustantivo(teclado, masculino, singular).
es_sustantivo(vpn, femenino, singular).
es_sustantivo(vpns, femenino, plural).
es_sustantivo(callcenterlog, _, singular).
es_sustantivo(amigo, masculino, singular).
es_sustantivo(amiga, femenino, singular).
```

```
preposiciones --> [].
preposiciones --> [a].
preposiciones --> [ante].
preposiciones --> [bajo].
preposiciones --> [con].
preposiciones --> [contra].
preposiciones --> [de].
preposiciones --> [desde].
preposiciones --> [durante].
preposiciones --> [en].
preposiciones --> [entre].
preposiciones --> [hacia].
preposiciones --> [hasta].
preposiciones --> [mediante].
preposiciones --> [para].
preposiciones --> [por].
preposiciones --> [segun].
preposiciones --> [sin].
preposiciones --> [sobre].
preposiciones --> [tras].
```

De este modo, para el caso de algunos componentes gramaticales es importante especificar el género y número, para que cumpla la concordancia y coherencia de la oración. Cabe destacar que si se usa una palabra que no está en la gramática no se va a poder reconocer como una oración válida. Esto junto a las limitadas estructuras de la oración que implementamos evita que se reconozcan muchas frases alejándose del lenguaje natural.

## 1.2 Descripción de las funciones implementadas

Dentro de los diversos componentes que le dan vida y un correcto funcionamiento a este programa existen funciones importantes con participación vital. Todas estas funciones se ubican en el archivo de “interfaz”. La primera que es importante analizar y una de las más importantes es la función diagnóstico, que se ve a continuación.

```
diagnostico([], [], []):-
    write("CallCenterLog:      Se recomienda ver el problema mas en detalle con un tecnico o
profesional\n"), !.

diagnostico([P|R1], [_R2], [_R3]):-
    retractall(entrada_usuario(_)),
    write("CallCenterLog:      "), write(P), write("\n"),
    write("Usuario:            "),
    leer_entrada,
    revisar_entrada,
    entrada_usuario(Frase),
    afirmacion(Frase), !,
    diagnostico(R1, R2, R3).

diagnostico(_, [S|_], [""|_]):-
    revisar_entrada,
    entrada_usuario(Frase),
    negacion(Frase), !,
    write("CallCenterLog:      "), write(S), write("\n").

diagnostico(_, [S|_], [R|_]):-
    revisar_entrada,
    entrada_usuario(Frase),
    negacion(Frase), !,
    write("CallCenterLog:      "), write(S), write("\n"),
    write("                    "), write(R), write("\n").

diagnostico(L1, L2, L3):-
    nlp_error,
    diagnostico(L1, L2, L3).
```

Esta función es la encargada de hacer las preguntas del problema al usuario para darle un diagnóstico y eventualmente una solución. Esta función recorre la lista de

preguntas hasta que el usuario indique cual síntoma presenta para presentarle una posible solución. Al mismo tiempo esta solución es la correspondiente con la pregunta realizada al usuario, y trae consigo una referencia relacionada a la solución en varios casos. Como podemos ver al principio se encuentra una función diagnóstico con las listas vacías, este caso hace referencia a cuando el usuario no indico ninguno de los síntomas dentro de nuestra base de datos, por lo que se le indica acudir a un profesional y que no le podemos ayudar en ese caso.

Por otra parte, otra función de importancia es “soy\_experto\_en”, esta función se encarga de analizar la oración para detectar el dispositivo o software del cual se habla en la oración y del cual vamos a dar un diagnóstico. Este soy\_experto\_en recorre toda la oración buscando una coincidencia con las palabras clave colocadas en la base de datos.

```
soy_experto_en(Palabra, [Palabra|_]):-
    es_sustantivo(P,_,_),
    conozco(P,_).

soy_experto_en(Palabra, [Palabra|_]):-
    es_verbo(P,_),
    conozco(P,_).

soy_experto_en(Palabra, [_|Resto]):-
    soy_experto_en(Palabra, Resto).
```

```
conozco(pc, 1).
conozco(pcs, 1z).
conozco(laptop, 1).
conozco(laptops, 1).
conozco(portatil, 1).
conozco(computador, 1).
conozco(computadora, 1).
conozco(computadoras, 1).
conozco(imprimir, 2).
conozco(impresora, 2).
conozco(impresoras, 2).
conozco(impresion, 2).
conozco(mouse, 3).
conozco(raton, 3).
conozco(clic, 3).
conozco(click, 3).
conozco(clicker, 3).
conozco(puntero, 3).
```

En el código podemos notar cómo en esta función se utiliza los hechos “conozco”, donde fueron declaradas nuestras palabras clave las cuales se ven relacionadas con un número específico que es el que nos especificara cual lista usar de la base de datos.

## 1.3 Descripción de las estructuras de datos utilizadas

Listas: En este caso, como se intenta modelar un sistema experto con una base de datos importante, la única estructura de datos que se vio necesaria implementar fueron listas. Estas nos facilitan la implementación de algunas funcionalidades de respuesta para el usuario. En este caso, para las tablas de preguntas, causas, soluciones y referencias utilizamos una lista donde se reflejan los distintos posibles outputs dependiendo del caso. A continuación, un ejemplo:

```
causas(1, ["No esta conectada a alimentacion electrica.",
           "Si es portatil, no esta cargada.",
           "Tiene el sistema de refrigeracion danado.",
           "Los ventiladores estan sucios o danados.",
           "Tiene muchas pestanas o un programa muy grande abierto.",
           "No tiene suficiente espacio en el disco.")).

preguntas(1, ["Ya reviso que tenga alimentacion electrica?",
              "Si es portatil, esta cargada?",
              "Tiene el sistema de refrigeracion en buenas condiciones?",
              "Tiene los ventiladores en buen estado?",
              "Tiene pocas pestanas abiertas y programas ligeros?",
              "Tiene espacio disponible en el disco duro?"]).

soluciones(1, ["Conecte su computadora al enchufe electrico.",
               "Cargue su laptop durante un tiempo suficiente.",
               "Reemplace el sistema de refrigeracion por uno nuevo.",
               "Limpie o cambie los ventiladores involucrados.",
               "Cierre las pestanas menos importantes y cierre programas pesados. En otro caso, anadale memoria a su computadora.",
               "Borre datos en disco que no este en uso o aumente el espacio en el disco. En otro caso puede conectar un dispositivo de almacenamiento externo.")).

referencias(1, ["",
                "",
                "Referencia para enfriamiento: https://extremetechcr.com/tienda/23-enfriamiento",
                "Referencia para ventiladores: https://extremetechcr.com/tienda/26-ventiladores.",
                "Referencia para memoria: https://www.intelec.co.cr/COMPONENTES/MEMORIAS-PARA-LAPTOP.",
                "Referencia para almacenamiento: https://extremetechcr.com/tienda/19-almacenamiento."]).
```

Como se puede ver en la imagen, dentro de causas, preguntas, soluciones y referencias notamos como hay un identificador, el cual es un 1 en el ejemplo dado, y posteriormente viene la lista implementada. De esta forma, habrá una función que recorra tal lista brindando el output requerido en cada caso.

## **1.4 Problemas sin solución**

Uno de los principales problemas que no se pudieron solucionar debido a su alta dificultad, es que el programa pueda reconocer todo tipo de oraciones, incluso lenguaje natural. Esto tiene 3 grandes inconvenientes, el primero es que la base de datos que tenemos es bastante limitada comparada con todas las palabras del idioma español, por lo que sería sumamente difícil agregar todas las posibilidades. Seguidamente, las limitadas estructuras gramaticales que incluimos ya que hay muchísimas formas de acomodar los distintos componentes gramaticales, por lo que de la misma forma habría que incluir todas las posibilidades de estructura de una oración. Por último, para entender el lenguaje natural del humano, no simplemente hay que entender el vocabulario y las estructuras gramaticales, sino que muchas veces el lenguaje coloquial ni siquiera llega a respetar esas reglas gramaticales.

## **1.5 Problemas encontrados**

Uno de los problemas más importantes encontrados fue que el programa pudiera reconocer 2 palabras clave por oración. Con esto me refiero a que, anteriormente se explicó que existen algunas palabras clave de periféricos y dispositivos que se distinguen dentro de la oración en los que somos “expertos”. Por ejemplo, la impresora, cuando el programa detecta la palabra impresora dentro de la oración sabemos que se refiere a un problema sobre esta, pero para el caso de cuando el usuario solamente pide referencias al respecto fue más difícil debido a que se debía reconocer la palabra referente a la “referencia”, ya sea “enlace”, “link” o “referencia” y al mismo tiempo reconocer el dispositivo del que desea la referencia. Por esta razón, durante el proceso de validación en el documento de “interfaz” hicimos un protocolo de “centerlog” que primero valide que la oración pida una referencia y posteriormente el programa sigue con normalidad validando el dispositivo o periférico solicitado, solamente que a diferencia del protocolo común esta devolverá las referencias relacionadas al dispositivo pedido. A continuación vemos la comparación de ambos protocolos.



```
centerlog_aux:-
    entrada_usuario(EntradaUsuario),
    usuario_pide_referencia(EntradaUsuario),
    split_string(EntradaUsuario, " ", "", Oracion),
    quitar_comillas_lista(Oracion, PalabrasSinComillas),
    soy_experto_en(Palabra, PalabrasSinComillas),
    conozco(Palabra, Numero),
    referencias(Numero, ListaReferencias),
    dame_referencias(ListaReferencias),
    finallog.
```

```
centerlog_aux:-
    entrada_usuario(EntradaUsuario),
    split_string(EntradaUsuario, " ", "", Oracion),
    quitar_comillas_lista(Oracion, PalabrasSinComillas),
    soy_experto_en(Palabra, PalabrasSinComillas),
    conozco(Palabra, Numero),
    preguntas(Numero, ListaPreguntas),
    soluciones(Numero, ListaSoluciones),
    referencias(Numero, ListaReferencias),
    diagnostico(ListaPreguntas, ListaSoluciones, ListaReferencias),
    !, finallog.
```

De esta forma notamos como en la primera imagen hay una línea de “usuario\_pide\_referencia” la cual valida que haya una palabra clave de referencia. Y seguidamente utiliza “soy\_experto\_en” la cual valida que haya una palabra clave del dispositivo y adelante retorna “referencias” con la lista de referencias. Por otra parte, el segundo protocolo es para el caso de que no se detecte el primero, por lo que simplemente empieza a preguntar para dar el diagnóstico.

## 1.6 Plan de actividades

Descripción de la tarea	Tiempo estimado	Responsable	Fecha de entrega
Lectura de los capítulos 1, 2, 3 y 6 del libro de Prolog y sobre gramáticas libre de contexto	3 horas	-Emanuel Marín	Martes 14 de marzo
Trabajar en la base de la gramática y que el programa pueda entender ciertas oraciones con distinción de género y número	2 horas	-Emanuel Marín	Jueves 16 de marzo
Lograr que el programa pueda dar un diagnóstico basado en el primer ejemplo de la especificación de la tarea, también que pueda distinguir al inicio de la conversación si el usuario solo saludo o indica de una vez el problema que presenta	3 horas	-Emanuel Marín	Viernes 17 de marzo
Hacer que el programa entienda cuando el usuario realiza una pregunta o pide por causas de un problema, basado en el segundo ejemplo de la especificación de la tarea	2 horas	-Emanuel Marín	Domingo 19 de marzo
Agregar documentación interna al archivo de interfaz	1 hora	-Emanuel Marín	Lunes 20 de marzo
Colaborar para que el programa pueda entender cuando el usuario pide por referencias de un problema	2 horas	-Emanuel Marín	Martes 21 de marzo
Agregar la funcionalidad de referencias, de modo que el usuario obtenga referencias sobre un problema específico.	4 horas	-Gabriel Chacon -Andres Rodriguez	Martes 21 de Marzo
Hacer mejoras y optimizaciones al código, ampliar la gramática para que el programa pueda entender más tipos diferentes de oraciones.	4 horas	-Emanuel Marín	Jueves 23 de marzo

Crear dos versiones del programa, la estándar y pro, en donde la última no se limita a reconocer la estructura de la oración, sino palabras claves de la misma			
Redactar el manual de usuario para un óptimo uso del programa.	4 horas	-Andres Rodriguez	Jueves 23 de marzo
Realizar una primera expansión de la base de datos del código.	2 horas	-Gabriel Chacon	Sabado 18 de marzo
Terminar la expansión completa de toda la base de datos con los distintos periféricos.	2 horas	-Gabriel Chacon	Domingo 19 de marzo
Terminar los distintos aspectos de la documentación técnica.	2 horas	-Gabriel Chacon	Domingo 26 de marzo

## 1.7 Conclusiones

- Se reafirmó y amplió el conocimiento del paradigma de programación lógico creando un Sistema Experto (SE) en el diagnóstico de problemas de TI con el lenguaje de programación Prolog.
- Las gramáticas libres de contexto son fáciles de implementar en Prolog por su naturaleza declarativa.
- Las listas como estructura de datos facilitaron el manejo y flujo de la información necesaria para la ejecución del programa.

## 1.8 Recomendaciones

- Se recomienda tomar en consideración el lenguaje Prolog cuando se desee desarrollar un SE de cualquier índole.
- Por la naturaleza y funcionamiento de los sistemas expertos, se recomienda hacer que estos sean muy flexibles entorno al aumento de su base de conocimientos y la manipulación de la misma, lo mismo para el caso de la gramática.
- Utilizar gramáticas de cláusulas definidas ya que facilitan la lectura y estructura de una oración y quienes las componen.

## 1.9 Bibliografía

Alonso Jiménez, J. A., Naranjo, M. A. G., & Martín Mateos, F. J. (s/f). *Tema 5: Procesamiento de lenguaje natural*. Cs.us.es. Recuperado el 27 de marzo de 2023, de <https://www.cs.us.es/~jalonso/cursos/ia2-00/temas/tema-05.pdf>

El Lenguaje de Programación PROLOG. (s/f). Mural.uv.es. Recuperado el 14 de marzo de 2023, de <http://mural.uv.es/mijuanlo/PracticasPROLOG.pdf>

swi-prolog.org. (s.f.). *Prolog Syntax*. Recuperado el 27 de marzo de 2023, de <https://www.swi-prolog.org/pldoc/man?section=syntax>

Extreme Tech CR. (s.f.). *Enfriamiento*. Recuperado el 27 de marzo de 2023, de <https://extremetechcr.com/tienda/23-enfriamiento>

Extreme Tech CR. (s.f.). *Ventiladores*. Recuperado el 27 de marzo de 2023, de <https://extremetechcr.com/tienda/26-ventiladores>

Extreme Tech CR. (s.f.). *Almacenamiento*. Recuperado el 27 de marzo de 2023, de <https://extremetechcr.com/tienda/19-almacenamiento>

HP. (s.f.). *Cómo conectar tu impresora a wifi*. Recuperado el 27 de marzo de 2023, de <https://www.hp.com/mx-es/shop/tech-takes/como-conectar-tu-impresora-a-wifi>

HP. (s.f.). *Cómo conectar una impresora a tu computadora*. Recuperado el 27 de marzo de 2023, de <https://www.hp.com/mx-es/shop/tech-takes/como-conectar-una-impresora-a-tu-computadora>

Intel. (2022). *Descargar controladores de gráficos Intel*. Recuperado el 27 de marzo de 2023, de <https://www.intel.la/content/www/xl/es/support/articles/000023767/graphics.html>

Intelec. (s.f.). *Memorias para Laptop*. Recuperado el 27 de marzo de 2023, de <https://www.intelec.co.cr/COMPONENTES/MEMORIAS-PARA-LAPTOP>

Microsoft. (s.f.). *Cómo actualizar Office*. Recuperado el 27 de marzo de 2023, de <https://support.microsoft.com/es-es/office/-c%C3%B3mo-actualizo-office-ee68f6cf-422f-464a-82ec-385f65391350>

Microsoft. (s.f.). *Cómo descargar e instalar los controladores de impresora más recientes*. Recuperado el 27 de marzo de 2023, de <https://support.microsoft.com/es-es/windows/c%C3%B3mo-descargar-e-instalar-los-controladores-de-impresora-m%C3%A1s-recientes-4ff66446-a2ab-b77f-46f4-a6d3fe4bf661>

Microsoft. (s.f.). *Cómo limpiar el mouse o el teclado de Microsoft*. Recuperado el 27 de marzo de 2023, de <https://support.microsoft.com/es-es/windows/-c%C3%B3mo-limpiar-el-mouse-o-el-teclado-de-microsoft-889f667b-3909-35db-af54-5091373c6219#:~:text=Limpiar%20el%20teclado&text=Humedece%20ligeramente%20un%20pa%C3%B1o%20sin,gotee%20agua%20en%20el%20teclado>

Pérez, C. (2019, febrero 14). *Cómo conectar tu portátil al monitor PC: diferencias HDMI, USB-C, DisplayPort*. Xataka. Recuperado el 27 de marzo de 2023, de <https://www.xataka.com/basics/conectar-tu-portatil-al-monitor-pc-como-hacerlo-diferencias-hdmi-usb-c-displayport>

Sony Latin America. (s.f.). *SVL series*. Recuperado el 27 de marzo de 2023, de <https://www.sony-latin.com/es/electronics/support/desktop-pc-svl-series/articles/00019317>

Sony Latin America. (s.f.). *Computadoras personales*. Recuperado el 27 de marzo de 2023, de <https://www.sony-latin.com/es/electronics/support/personal-computers/articles/00019315>

Solvetic. (2019, marzo 7). *Cómo Conectar un Portátil a un Monitor de PC*. Recuperado el 27 de marzo de 2023, de [https://www.youtube.com/watch?v=vLS6K6ohEgc&ab\\_channel=solvetic.com](https://www.youtube.com/watch?v=vLS6K6ohEgc&ab_channel=solvetic.com)

## 1.10 Bitácora

Fecha	Presentes	Duración	Avance
14/03/2023	-Gabriel Chacon -Andres Rodriguez -Emanuel Marin	1 hora	Se realizó una reunión en la que se discutió la especificación de la tarea y se definió un tema central para la misma.
17/08/2023	-Gabriel Chacon	50 mins	Se realizó una expansión pequeña de la base de datos tomando el caso del mouse y teclado. De esta forma se agregaron causas, preguntas y soluciones.
17/03/2023	-Gabriel Chacon -Emanuel Marin	1 hora	Se realizó una reunión para entender la gramática y la base de datos con la que se empezó. Para posteriormente agrandar estas.
18/03/2023	-Gabriel Chacon -Emanuel Marin	1hora	Se comentaron algunas posibles soluciones para poder hacerle preguntas al usuario. Además se evaluaron algunos ejemplos distintos para poder agregarlos a la base de datos.
19/03/2023	-Gabriel Chacon	1hora	Se pensaron distintos periféricos con los que se podrían trabajar para la solución de problemas del cliente. De esta forma se empezó a generar

			la base de datos de preguntas, causas y soluciones.
20/03/2023	-Gabriel Chacon	1 hora	Se terminó la mayoría de la base de datos, teniendo hasta el momento 13 perifericos.
21/03/2023	-Gabriel Chacon -Andres Rodriguez	4 horas	Se hizo una reunión para añadir la funcionalidad de las referencias, de forma que se buscaron distintos sitios para comprar o solucionar problemas del usuario. Seguidamente, se agregaron a las referencias a cada una de las soluciones y se agregó una tabla de referencias en la base de datos. Por último, se agregaron funciones para el reconocimiento directo de referencias.
24/03/2023	-Gabriel Chacon	1 hora	Se empezó la documentación con las estructuras de datos implementadas y los algoritmos utilizados.
26/03/2023	-Gabriel Chacon	2 horas	Se terminó la documentación añadiendo imágenes y otras secciones.
14/03/2023	-Emanuel Marin	3 horas	Hoy empecé a estudiar sobre gramáticas libres de contexto y BNFs y repasé algunos capítulos del libro de Prolog que se relacionan con el proyecto de CallCenterLog.
16/03/2023	-Emanuel Marin	2 horas	El día de hoy trabajé en la base de la gramática y que el programa pueda entender ciertas oraciones, en algunos casos sabiendo distinguir el número (singular o plural) y género (masculino o femenino).
17/03/2023	-Emanuel Marin	3 horas	<p>Hoy logré que el programa pueda seguir una conversación muy parecida al primer ejemplo de la especificación de la tarea, donde el usuario saluda e indica el problema y a partir de eso el programa hace preguntas para hacer un diagnóstico e identificar cuál podría ser la solución.</p> <p>El programa además ya sabe identificar si al inicio de la conversación el usuario solo saluda o indica de una vez el problema.</p> <p>También definí una estructura para la base de datos del programa.</p>

19/03/2023	-Emanuel Marin	2 horas	En la tarde trabajé para que el programa pueda reconocer cuando el usuario le hace una pregunta o pide por causas de un problema que presenta.
20/03/2023	-Emanuel Marin	1 hora	Hoy le hice mejoras a la gramática y agregué documentación interna al código de la interfaz. También estuve haciendo pruebas para optimizar el flujo del programa y la llamada de cláusulas auxiliares.
21/03/2023	-Emanuel Marin	2 horas	Estuve en llamada colaborando con Gabriel y Andres para que el programa pueda entender cuando el usuario pida por referencias de un problema. Agregué la documentación interna de eso.
22/03/2023	-Emanuel Marin	1 hora	Hice algunas mejoras y optimizaciones al código de la gramática e interfaz para que pueda entender más estructuras de oraciones. También trabajé para que cuando el programa espera que el usuario ingrese algún texto, si este lo cierra antes de, el programa no caiga en bucle.
23/03/2023	-Emanuel Marin	1 hora	Hoy creé dos versiones del programa, la estándar que se limita a que el programa solo entienda aquello que está definido o considerado como una oración en el archivo de gramática y la versión pro que quita esa limitante y se concentra más en encontrar patrones o palabras claves para el flujo de la conversación.
23/03/2023	-Andres Rodríguez	4 horas	Se redactó el manual de usuario para garantizar un correcto uso del programa por parte del usuario final.
25/03/2023	-Andres Rodríguez	1 hora	Se complementó el manual de usuario con más ejemplos y casos particulares importantes para el uso del programa.