



**UNIVERSIDAD TECNOLÓGICA CENTROAMERICANA**

**FACULTAD DE INGENIERÍA**

**PROYECTO DE GRADUACIÓN**

**GENERADOR DE CÓDIGO G CON PYTHON**

**PRESENTADO POR:**

**11641109 - GONZALO ROBERTO VALLE ISCANO**

**ASESOR: ING. MIGUEL ENRIQUE RAMÍREZ ROMERO**

**CAMPUS TEGUCIGALPA; ABRIL, 2022**

## RESUMEN EJECUTIVO

Los lenguajes de programación en la actualidad han sido necesario y también útiles para la gran mayoría. En este proyecto veremos como otro lenguaje de programación funciona como traductor para la generación de código de otro lenguaje.

En este proyecto se utilizó Python por el hecho que es un lenguaje de alto nivel que asimila y se aproxima al lenguaje humano. Por otro lado, el Código G siendo un lenguaje de bajo nivel se aproxima al lenguaje máquina, siendo poco comprensible para un humano entender a lo que se esté ejecutando en el momento. Python siendo el lenguaje de programación traductor para generar el código G en una forma simple y amigable.

Este proyecto se enfoca en el desarrollo de una librería de Python la cual genera código G. Esta librería de igual manera se ha hecho con el propósito de que sea amigable e intuitiva. Se realizó la documentación necesaria para el entendimiento del uso del generador en GitHub. El generador se tuvo que programar que sea capaz de exportar a un archivo en gcode, cnc y nc, ya que son formatos los cuales una máquina CNC acepta para ser programada. De igual manera, se desarrolló una forma que fuese capaz de exponer el código G generado en un texto plano.

El simulador utilizado en el proyecto fue CAMotics el cual este se empleó de tal manera que acepta abrir archivos con extensión nc. También tiene un visualizador para mostrar el procedimiento de una fresadora CNC en movimiento. Como también muestra el trayecto y el cambio de herramienta, si el programa desea que se cambie de herramienta.

## **ABSTRACT**

Programming languages nowadays have been necessary and useful for many people. In this project we will see how another programming language works as a translator for the generation of code from another language.

Python was developed in this project because it is a high-level language that assimilates and approximates human language. On the other hand, the G Code, a low-level language, is close to machine language, being difficult for a human to understand what is being finished at that moment. Python being the translator programming language to generate G code in a simple and friendly way.

This project focuses on the development of a Python library which generates G code. This library has also been made with the purpose of being friendly and intuitive. The necessary documentation was made to understand the use of the generator on GitHub. The generator had to be programmed to be able to export a file in gcode, cnc and nc, since they are formats that a CNC machine accepts to be programmed. Similarly, a way will be developed that could expose the generated G-code in plain text.

The simulator used in the project was CAMotics, which was used in such a way that it accepts opening files with the nc extension. It also has a display to show the procedure of a CNC mill in motion. As it also shows the path and the tool change, if the program wants the tool to be changed.

# ÍNDICE DE CONTENIDO

I	Introducción.....	1
II	Planteamiento del problema.....	2
2.1	Precedentes del problema .....	2
2.2	Definición del problema.....	2
2.3	Justificación.....	2
2.4	Preguntas de investigación.....	2
2.5	Objetivos.....	2
2.5.1	Objetivo general .....	3
2.5.2	Objetivos específicos.....	3
III	Marco Teórico .....	4
3.1	Código G.....	4
3.2	Código M.....	4
3.3	Máquinas-Herramientas.....	5
3.4	Control Numérico por computador.....	5
3.5	Fresadora .....	6
3.5.1	Fresa.....	7
3.5.1.1	Clasificación de Fresas.....	7
3.5.2	Husillo .....	8
3.5.3	Función del Centro Numérico Computarizado .....	8
3.6	Visual Studio Code.....	8
3.7	Sublime Text.....	<b>Error! Bookmark not defined.</b>
3.8	Python.....	9

3.8.1	Clases .....	10
3.8.2	Librerías.....	10
3.8.2.1	Librerías Estática.....	10
3.8.2.2	Librerías Dinámicas.....	10
3.8.3	Método.....	10
3.8.4	Diccionarios .....	11
3.8.5	Enum.....	11
3.9	GitHub.....	11
3.9.1	Repositorios.....	12
3.10	CAMOTICS.....	<b>Error! Bookmark not defined.</b>
3.11	SolidWorks .....	12
IV	Metodología .....	13
4.1	Enfoque .....	13
4.2	Variables de investigación.....	13
4.3	Técnicas e instrumentos aplicados .....	13
4.4	Metodología de estudio.....	13
4.4.1	Programación del generador de código G en Python .....	14
4.4.2	Programación Amigable .....	14
4.4.3	Pruebas en simulador de CNC.....	14
4.4.4	Documentación después del desarrollo del generador.....	14
4.4.5	Exportación de un archivo en formato código G .....	14
4.4.6	Exponer texto plano .....	15
4.5	Cronograma de actividades.....	16

V	Resultado y análisis .....	17
5.1	Programación del generador .....	17
5.1.1	Primera versión.....	17
5.1.2	Segunda Versión.....	18
5.1.3	Tercera Versión.....	18
5.1.4	Ultima Versión .....	19
5.2	Programación amigable del generador .....	19
5.2.1	Comparación de comandos en código G y con el uso de la librería de Python. ....	19
5.3	Documentación del generador.....	20
5.4	Documentación elaborada para el proyecto.....	21
5.5	Pruebas en simulador de CNC.....	26
VI	Conclusiones.....	27
VII	Recomendaciones.....	28
VIII	Bibliografía.....	29

## ÍNDICE DE ILUSTRACIONES

Ilustración 1 Fresadora Horizontal.....	5
Ilustración 2 Fresadora CNC.....	6
Ilustración 3 Diferentes tipos de fresas para diferentes operaciones que se pueden realizar con una fresadora.....	7
Ilustración 4 Husillo con Fresa en Cabezal Fresador .....	8
Ilustración 5 Icono de Visual Studio Code.....	9
Ilustración 6 Imagotipo de Python.....	<b>Error! Bookmark not defined.</b>
Ilustración 7 Ejemplo de un Diccionario.....	11
Ilustración 8 Logo de GitHub .....	11
Ilustración 9 JSON utilizado para pruebas .....	17
Ilustración 10 Archivo prueba.py primera version.....	18
Ilustración 11 Comparación de lenguajes de programación. Preparación de la Máquina, Compensación de corte y corte en arco .....	19
Ilustración 12 Documentación elaborada en GitHub .....	20
Ilustración 13 Simulación del archivo "Robot Face.nc" generado del "Robot Face.py" .....	26

## ÍNDICE DE TABLAS

Tabla 1 Grupo de Códigos G con sus definiciones .....	4
Tabla 2 Cronograma de Actividades.....	16



## **LISTA DE SIGLAS Y GLOSARIOS**

CNC: Control Numérico por Computador

Código G: Lenguaje de programación de bajo nivel usado en máquinas herramientas CNC

Código M: Lenguaje de programación que ordenan maniobras a una máquina CNC.

Dextrógiro: Que gira en el sentido de las agujas del reloj.

Lenguaje de alto nivel: es el cual se aproxima más al lenguaje humano.

Lenguaje de bajo nivel: es el cual se aproxima más al lenguaje máquina.

Lenguaje máquina: Lenguaje binario expresado por ceros y unos, únicamente.

Fresadora por control numérico: Es una máquina-herramienta controlada por un programa, a diferencia a una fresadora convencional que es manual.

Impresora 3D: Es una maquina capaz de realizar replicas en 3D, para crear piezas a partir de un diseño digital.

Máquina-Herramienta: Es un tipo de máquina que se utiliza para dar forma a piezas sólidas, comúnmente en metales.

Programación amigable: Es una cualidad de una interfaz de programación que lograr interactuar con el usuario de una manera fácil e intuitiva

Torno por control numérico1: Es una máquina-herramienta controlada por un programa, a diferencia a un torno convencional que es manual.

## I INTRODUCCIÓN

El presente informe detalla los diferentes objetivos realizados en el proyecto de generación de código G con el lenguaje de Python. El código G también se enseña en clases universitarias para el uso de las maquinas-herramientas CNC de una forma básica.

El proyecto se centra en simplificar al desarrollador la realización de código G a través de un generador creado en Python. Se creará una librería capaz de generar código G. Se programará una librería de Python basado en funciones, para evitar tener código amontonado y desordenado.

Se realizará una documentación elaborada en GitHub de esta manera desarrolladores puedan generar código G solo teniendo el conocimiento de Python. Se exportará un archivo en un formato legible para una máquina CNC. Se expondrá en texto plano con la función print de una manera sencilla para visualizar el seguimiento del código realizado en el momento. Se realizarán pruebas en un simulador de fresadora CNC. Para comprobar que la librería no tiene fallas en la generación del código generado como también, se obligará al generador a fallar poniendo datos incoherentes.

## **II PLANTEAMIENTO DEL PROBLEMA**

### **2.1 PRECEDENTES DEL PROBLEMA**

El Código G fue inventado en la década de 1950. Observando la estructura del código G es un lenguaje de programación de bajo nivel, es poco amigable para para un programador y los que no son de ese rubro ya que tiene comandos mal descritos. En la actualidad uno de los aspectos más importantes de un lenguaje de programación es lo amigable y lo simplificado que es para el desarrollador. El auge de las impresoras 3D, fresadoras CNC y las máquinas de grabado laser hacen necesario el conocimiento de programación en código G.

### **2.2 DEFINICIÓN DEL PROBLEMA**

Actualmente existen lenguajes de programación de alto nivel los cuales se aproximan al lenguaje humano como los es Python. Lo ambiguo y lo poco descrito del código G es una de las consecuencias que los desarrolladores en la actualidad no les llama la atención aprenderlo. El código G es utilizado en la industria de manufactura automatizada por lo que es necesario aprender código G. Python siendo uno de los lenguajes de programación más comprensibles para los programadores, servirá como un traductor y generador de código G.

### **2.3 JUSTIFICACIÓN**

Simplificar el código G para que los desarrolladores de otros lenguajes puedan generar código G sin tener conocimiento de él. Así la industria de manufactura automatizada no tenga la necesidad de contratar personal con conocimiento de él.

### **2.4 PREGUNTAS DE INVESTIGACIÓN**

- ¿Cómo desarrollar una librería Python que sea amistosa con el usuario?
- ¿Cómo exportar datos de Python a un archivo Código G?
- ¿Cómo exponer datos de Python a texto plano?
- ¿Cuáles son los simuladores aptos para usar en este proyecto para evitar pérdidas económicas en la industria debido a fallas en la programación?

### **2.5 OBJETIVOS**

### 2.5.1 OBJETIVO GENERAL

Generar los códigos más comunes del código G para fresadora CNC con ayuda del lenguaje de programación Python.

### 2.5.2 OBJETIVOS ESPECÍFICOS

- Desarrollar un generador de código G con Python para hacer la programación de forma intuitiva usando código limpio.
- Realizar una documentación posterior al desarrollo del generador para que cualquier usuario pueda desarrollar archivos.
- Exponer un texto plano para extender la funcionalidad de la librería de Python a futuro.
- Realizar pruebas en un simulador de fresadora CNC en CAMotics para comprobar que nuestro código G no tengas fallas en la generación y realización de la pieza.

### III MARCO TEÓRICO

#### 3.1 CÓDIGO G

Los códigos G son utilizados para dar órdenes específicas para la máquina: por ejemplo, movimientos, selección de plano, unidades, compensación de corte. También, ordenan características más complicadas que pueden implicar herramientas motorizadas opcionales y el eje C. Cada código G tiene un número de grupo. Cada grupo de códigos contiene comandos para un objetivo específico. Por ejemplo, el grupo 1 de la tabla 1 de códigos G ordena movimientos punto a punto de los ejes de la máquina, mientras que el grupo 7 es específico para la característica de Compensación de la herramienta de corte. Se puede observar más grupos en la Tabla 1 con su definición de acuerdo con el grupo. (Haas F1 Team, 2022a)

Las máquinas que suelen usar código G son los de control numérico por computador (CNC) como: Cortadora Láser, Fresadora CNC, Torno CNC, Impresoras 3D.

**Tabla 1 Grupo de Códigos G con sus definiciones**

<b>NÚMERO DE GRUPO</b>	<b>DEFINICIÓN</b>
00	No modal
01	Movimiento
02	Selección de plano
03	Modo Distancia
05	Modo velocidad de avance
06	Unidades
07	Compensación de la herramienta de corte
08	Corrector de longitud de la herramienta
09	Ciclos fijos
10	Modo de retorno en ciclos fijos
11	Escala
12	Selección del sistema de coordenadas
15	Parada exacta
16	Rotación
23	Corrector de piezas dinámico

Fuente: (Haas F1 Team, 2022a)

#### 3.2 CÓDIGO M

Los códigos M, se usan en conjunto con el código G, pero hacen distintas funciones, por ejemplo: detener el programa, encender o apagar el refrigerante, invertir el husillo entre otros. Solo se permite un código M por línea. Los códigos M son varios comandos de la máquina que

no ordenan el movimiento de los ejes. El formato de un código M es la letra M seguida de dos a tres dígitos, por ejemplo, M03.(Haas F1 Team, 2022b)

### 3.3 MÁQUINAS-HERRAMIENTAS

La definición de máquina-herramienta engloba una amplia variedad de máquinas con un denominador común: todas ellas se dedican a fabricar productos o partes de productos (generalmente metálicos, pero no obligatoriamente). Se dice que las máquinas-herramienta son máquinas madre porque posibilitan la fabricación de todas las demás máquinas incluyendo a ellas mismas.(Asociación Española de Fabricantes de Máquinas-Herramienta, 2022)



**Ilustración 1 Fresadora Vertical**

Fuente: (De máquinas y herramientas, 2015)

Hay varias máquinas-herramientas las cuales podemos utilizar, algunos ejemplos son, tornos, prensas, rectificadoras, máquinas de perforación, etc. Como se puede observar en la ilustración 1, es una fresadora vertical, las flechas en rojo, indican las manivelas de movimiento "X" y "Y" de la mesa.

### 3.4 CONTROL NUMÉRICO POR COMPUTADOR

El control numérico por computadora (comúnmente conocido como CNC) es un sistema que permite controlar en todo momento la posición de un elemento físico. Normalmente una herramienta, que está montada en una máquina.

Esto se consigue mediante un programa y un conjunto de órdenes añadidas. Con ambos, se pueden controlar las coordenadas de posición de un punto respecto a un origen (la posición de la máquina). En pocas palabras, estamos trabajando con una especie de Sistema de Posicionamiento, pero aplicado al mundo de los mecanizados, y muchísimo más preciso.(Mecanizados Sinc, 2015)



**Ilustración 2 Fresadora CNC**

Fuente: (Mecanizados Sinc, 2019)

En la ilustración 2 se puede observar una fresadora CNC en el recuadro rojo indica el panel de operaciones, lugar donde se puede observar el proceso del programa a realizar.

### **3.5 FRESADORA**

Una fresadora es una máquina-herramienta que sirve para desbastar, perforar, cortar y/o detallar, ya sea metal, madera o cualquier material sólido. Estas operaciones se logran haciendo girar y moviendo una herramienta de corte llamada "fresa" a través de un juego de manivelas, una para el eje vertical (eje z) y otras dos para el plano horizontal (eje x, y).(Ingeniería Mecafenix, 2018)

Su funcionamiento se realiza a través del arranque de viruta por medio de la rotación a alta velocidad de la fresa y del movimiento de los ejes, ya sea del elemento o del mecanismo.





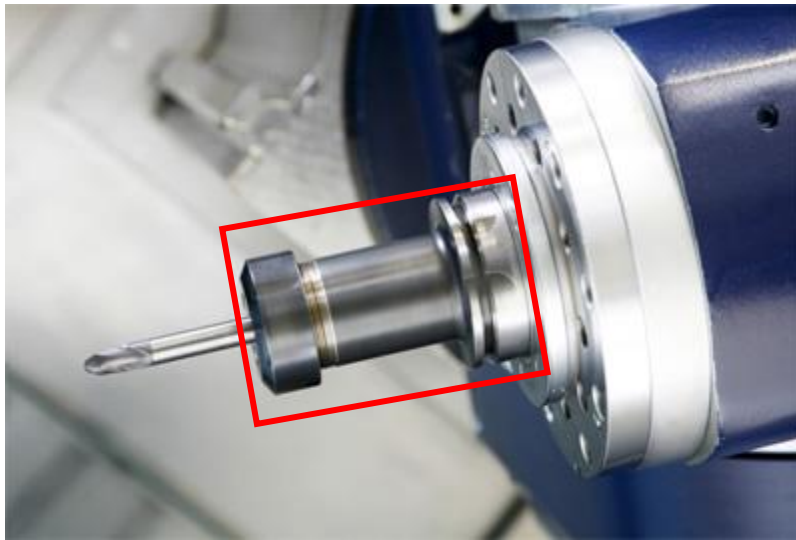
- Superficie Fresada
- Superficie Escalonada.

(De máquinas y herramientas, 2012)

### 3.5.2 HUSILLO

Proporciona una propulsión para árboles, cortadores y aditamentos utilizados en una maquina fresadora.(Steve F. Krar, 2001)

En la ilustración 4, en el recuadro rojo, podemos observar el husillo de una fresadora.



**Ilustración 4 Husillo con Fresa en Cabezal Fresador**

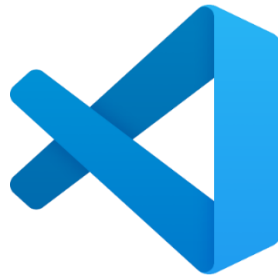
Fuente: (Asociación Española de Fabricantes de Máquinas-Herramienta, 2022)

### 3.5.3 FUNCIÓN DEL CENTRO NUMÉRICO COMPUTARIZADO

La función principal de una fresadora CNC es controlar los desplazamientos de la mesa movediza, los carros transversales o laterales y/o el movimiento del husillo en los ejes principales por medio de los datos numéricos ingresados en el programa Código G.(Stanser, 2018)

## 3.6 VISUAL STUDIO CODE

Visual Studio Code es un editor de código fuente, ligera y potente que se ejecuta en el escritorio y está disponible para Windows, macOS y Linux. Tiene soporte incorporado para JavaScript, TypeScript y Node.js también tiene una gran variante de extensiones para otros lenguajes (como Python, Java, Rust, Go) y tiempos de ejecución (como .NET y Unity).(Microsoft, 2022b)



**Ilustración 5 Icono de Visual Studio Code**

Fuente: (Microsoft, 2022a)

En la ilustración 5 podemos observar el icono que es utilizado en visual studio code, para identificar el programa.

### **3.7 PYTHON**

Python es un lenguaje de programación de alto nivel que se utiliza para desarrollar aplicaciones de todo tipo. En contraste de otros lenguajes como Java o .NET, se trata de un lenguaje interpretado, es decir, que no es necesario compilarlo para ejecutar las aplicaciones escritas en Python, sino que se ejecutan directamente por el ordenador utilizando un programa denominado interpretador, por lo que no es necesario "traducirlo" a lenguaje máquina.

Python es un lenguaje fácil de leer y escribir debido a su gran similitud con el lenguaje humano. Además, se trata de un lenguaje multiplataforma de código abierto. Con el paso del tiempo, Python ha ido ganando partidarios gracias a su simplicidad y a sus amplias posibilidades, sobre todo en los últimos años, ya que facilita trabajar con inteligencia artificial, big data, machine learning y data science, entre muchos otros campos en auge.(Santander Universidades, 2021)

### 3.7.1 CLASES

Una clase es la descripción de un conjunto de objetos similares; consta de métodos y de datos que resumen las características comunes de dicho conjunto. En un lenguaje de programación orientada a objetos se pueden definir muchos objetos de la misma clase de la misma forma que, en la vida real, haríamos galletas (objeto) con el mismo molde (clase) solo que, para entenderlo mejor, cada galleta tendría igual forma, pero es posible que tenga distinto sabor, textura, olor, color, etc.(José Manuel Ramírez, 2015)

### 3.7.2 LIBRERÍAS

Una librería de programación es el conjunto de archivos que se utilizan para poder desarrollar un software. Una librería es común que este compuesta de código su fin es utilizarlo en otros programas de forma totalmente autónoma.(devCamp, 2020)

#### *3.7.2.1 Librerías Estática*

Las librerías estáticas son grabadas en un programa, ejecutables (.exe en Windows) sirven exclusivamente para esto; ejecutando la librería estática, después se pueden borrar sin problema por el hecho de que el programa seguirá funcionando con la librería necesaria.(devCamp, 2020)

#### *3.7.2.2 Librerías Dinámicas*

Las librerías dinámicas son diferentes a las librerías estáticas en cuanto no se copian en el programa al compilarlas. Las subrutinas son cargadas en tiempo de ejecución, en vez de enlazarse en tiempo de compilación.(devCamp, 2020)

### 3.7.3 MÉTODO

Un método es un bloque de código que contiene una serie de instrucciones. Un programa hace que se ejecuten las instrucciones al llamar al método y especificando los argumentos de método necesarios.(Microsoft, 2022c)

### 3.7.4 DICCIONARIOS

Un diccionario es una estructura de datos y un tipo de dato en Python que tiene ciertas características que permiten almacenar cualquier tipo de valor como son enteros, cadenas, listas e incluso funciones.

```
diccionario = {'nombre' : 'Carlos', 'edad' : 22, 'cursos': ['Python', 'Django', 'JavaScript' ] }
```

#### **Ilustración 6 Ejemplo de un Diccionario**

Fuente: (Carlos Eduardo Plasencia Prado, 2021)

Como se observa en la ilustración 7 de esta manera se puede definir un diccionario, se utilizan llaves para encerrar el listado. Se utilizan las comas para separar una clave de otra, mientras que para separar la clave del valor asignado se utilizan los dos puntos.

### 3.7.5 ENUM

Los Enum es un tipo de dato especial, que permite que una variable sea un conjunto de constantes predefinidas. Por ejemplo:

- Brújula (valores de Norte, Sur, Este y Oeste)
- Días de semana (lunes, martes, miércoles, jueves, viernes, sábado y domingo).

(Oracle, s. f.)

## **3.8 GITHUB**

GitHub es una plataforma de hospedaje de código para el control de versiones y la colaboración. Esto permite que nosotros y otras personas podamos trabajar en conjunto en proyectos desde cualquier lugar.(GitHub, Inc, 2022)

The image shows the GitHub logo, which consists of the word "GitHub" in a bold, black, sans-serif font. The "i" in "Git" has a small octocat icon as its dot.

**Ilustración 7 Logo de GitHub**

En la ilustración 8 podemos observar el logo que utiliza la página web de GitHub.

### 3.8.1 REPOSITARIOS

Un repositorio contiene todos los archivos de tu proyecto y el historial de revisiones de cada uno de ellos. Puedes debatir y administrar el trabajo de tu proyecto dentro del repositorio.

## 3.9 CAMOTICS

CAMotics es un simulador de programas en código G de 3 ejes para CNC y para lograr visualizar los resultados en 3D. CAMotics está para los sistemas operativos Linux, macOS o Windows. Es rápido, intuitivo, de código abierto y gratuito. El hecho de poder simular sus trayectorias de herramientas CNC, evita errores peligrosos y costosos.(CAMotics, 2019)

## 3.10 SOLIDWORKS

Es un software de diseño de CAD 3D para modelar ensamblajes en 3D y planos en 2D. El software ofrece la simulación de una fresadora CNC como un aditamento del paquete de "SolidWorks CAM". En la versión 2018 del programa fue incorporado la programación en CNC directamente en la herramienta de CAD 3D.(SOLIDDBI Inspira tu innovacion, 2019)

SolidWorks, no es compatible con el código G que generamos debido a que se genera a base de comandos y modificaciones específicas del programa.

## **IV METODOLOGÍA**

### **4.1 ENFOQUE**

El enfoque cualitativo también se guía por áreas o temas significativos de investigación. Sin embargo, en lugar de que la claridad sobre las preguntas de investigación e hipótesis preceda a la recolección y el análisis de los datos, los estudios cualitativos pueden desarrollar preguntas e hipótesis antes, durante o después de la recolección y el análisis de los datos.(Roberto Hernández Sampieri, 2014). Por esta razón se ha decidido usar un enfoque de metodología cualitativa, ya que el desarrollo de un proyecto de software implica analizar los datos y como se van a desarrollar. Este proyecto está orientado a asegurar el cumplimiento de requerimientos técnicos específicos para la creación de código G utilizando Python.

### **4.2 VARIABLES DE INVESTIGACIÓN**

- Distintos formatos de archivos de exportación para el código G.
- Distintos resultados de acuerdo con el simulador.

### **4.3 TÉCNICAS E INSTRUMENTOS APLICADOS**

- Python: lenguaje de programación de alto nivel, el lenguaje tiene varias librerías e incluso se puede añadir más librerías haciéndolas uno mismo. Para su uso es necesario el conocimiento de programación en Python y sus librerías.
- CAMotics: software de simulación de fresadora CNC (control numérico por computador) para su uso es necesario el conocimiento de programación en código G, y código M.
- Visual Studio Code: Es un software editor de código fuente, utilizado por programadores por el hecho de su soporte nativo para una gran variedad de lenguajes, entre ellos podemos destacar HTML, CSS, JavaScript, entre otros. También su posibilidad de personalización.

### **4.4 METODOLOGÍA DE ESTUDIO**

#### 4.4.1 PROGRAMACIÓN DEL GENERADOR DE CÓDIGO G EN PYTHON

Las metodologías tradicionales como el modelo de cascada son ideales para proyectos en los que se tiene bien definido el alcance y se sabe que no existirán modificaciones, las metodologías ágiles son flexibles, adaptables e iterativas. Para el desarrollo del proyecto se optó por un modelo ágil, ya que es sabido desde el inicio, que por la falta de conocimiento profesional de los lenguajes Python y G-Code, implicarían cambios ligeros durante el desarrollo del proyecto. Adicionalmente, este modelo permite hacer entregables durante tiempos cortos lo que permite pequeñas o grandes modificaciones del proyecto.

#### 4.4.2 PROGRAMACIÓN AMIGABLE

Se utilizó código limpio utilizando nombres de variables que indican cuales es su propósito y evitando la memorización de las variables. Al buscarlas en el código se le hace fácil recordar por el nombre utilizado y así conseguir una programación más amigable.

#### 4.4.3 PRUEBAS EN SIMULADOR DE CNC

Se evaluó un simulador, donde se pudiera utilizar en cualquier sistema operativo cómo lo es Linux, Windows y MAC OS, de tal manera que no haya limitantes al probar el código G generado.

#### 4.4.4 DOCUMENTACIÓN DESPUÉS DEL DESARROLLO DEL GENERADOR

Al finalizar el desarrollo del generador se empezó con la documentación de dicho generador para nuevos desarrolladores que quieran utilizar la librería creada. Se realizaron las instrucciones de uso de cada una de las funciones creadas en la librería y como utilizarlas individualmente.

#### 4.4.5 EXPORTACIÓN DE UN ARCHIVO EN FORMATO CÓDIGO G

Se investigó como programar en Python para exportar un archivo con el formato deseado, en este caso es con las extensiones comunes: nc, cnc y gcode.

```
for idx, line in enumerate(lines):
    code += f'N{idx}    {line}\n'
    self.code = code

exported_filename = file_name or f'{self.nombre_proyecto}.gcode'

with open(exported_filename, 'w') as f:
    f.write(code)
```

### **Ilustración 8 Investigación de exportación de archivo en formato aceptable de código G.**

#### 4.4.6 EXPONER TEXTO PLANO

Se buscó en libros de texto de Python la manera de exponer un texto plano por si desea extender la funcionalidad de la librería de Python en un futuro y así continuar con la simplicidad de la programación del proyecto.



#### 4.5 CRONOGRAMA DE ACTIVIDADES

A continuación, se observa un diagrama de Gantt referente a los objetivos específicos del proyecto a lo largo de un plazo de 10 semanas

**Tabla 2 Cronograma de Actividades**

<b>Actividades</b>	<b>Semana 1</b>	<b>Semana 2</b>	<b>Semana 3</b>	<b>Semana 4</b>	<b>Semana 5</b>	<b>Semana 6</b>	<b>Semana 7</b>	<b>Semana 8</b>	<b>Semana 9</b>	<b>Semana 10</b>
Programación del generador de código G en Python	■	■	■	■					■	
Programación del generador convertirla más amigable	■	■	■	■					■	
Pruebas en simulador de CNC				■	■	■				
Documentación después del desarrollo del generador						■	■	■		
Exportación de un archivo en formato código G									■	
Exponer en texto plano									■	

Fuente: (Elaboración propia)

## V RESULTADO Y ANÁLISIS

En este proyecto se desarrolló una librería de Python en el cual se utiliza como generador de código G. Esta librería se hizo con el propósito de generar los códigos más comunes utilizados en una maquina fresadora CNC. En las próximas secciones podremos ver la comparación entre la librería desarrollada y la programación en código G.

### 5.1 PROGRAMACIÓN DEL GENERADOR

A medida se estuvo programando la librería del generador de código G se hicieron cuatro versiones las cuales cada una contenían distintas formas de desarrollar el código.

#### 5.1.1 PRIMERA VERSIÓN

La primera versión del generador contenía los comandos de inicialización y la generación del código G. La inicialización del programa contenía solamente lo que es el tipo de posicionamiento, las unidades a utilizar, el numero identificador del proyecto, el nombre del programa, revoluciones del husillo y el encendido. Adicionalmente se utilizó un archivo JSON para una configuración de una maquina con atributos ya predefinidos por la máquina, para uso de prueba de igual modo se creó un nuevo programa llamado prueba.py con el mismo fin.

```
() config.json > ...
1  {
2  |   "machine_name": "GR-510",
3  |   "position": "absoluto",
4  |   "units": "pulgadas",
5  |   "tools": [
6  |     {
7  |       "id": 3,
8  |       "rpm": 7000,
9  |       "fv": 1.4,
10 |       "fh": 4
11 |     },
12 |     {
13 |       "id": 1,
14 |       "rpm": 6500,
15 |       "fv": 1.6,
16 |       "fh": 5
17 |     }
18 |   ]
19 | }
```

**Ilustración 9 JSON utilizado para pruebas**

Fuente: (Elaboración propia)

```

T1 = {
    'label': 'Excavadora corta',
    'name': 'Excavator23',
    'rpm': '7000',
    'h': 23,
    'id': 23,
}
T2 = {
    'label': 'Excavadora larga',
    'name': 'Excavator44',
    'h': 44,
    'id': 44,
}

machine = CNC(
    id_proyecto=1234,
    project_name="Aleta de Enfriamiento",
    x=1,
    y=2,
    position='incremental',
    units='milimetro',
    tools=[T1,T2]
)

machine.print_code()

```

### Ilustración 10 Archivo prueba.py primera version

Fuente: (Elaboración propia)

#### 5.1.2 SEGUNDA VERSIÓN

La segunda versión del programa se le añadieron los comandos de corte, cambio de herramienta, vaciado de forma circular, movimiento encima del material, compensaciones y el cierre de programa del código G generado.

#### 5.1.3 TERCERA VERSIÓN

La tercera versión de la librería se le agregaron los comandos de refrigerante, vaciado en forma de espiral y el de subrutina externa, se le agregó Enum a las clases de Plano y CicloTaladro de forma que se hereden a cada una también se exportó a un archivo con extensión gcode. Se modificaron los comandos necesarios por el hecho de agregar el comando de refrigerante debido a que algunos comandos cambian por el uso de refrigerante. También se añadió el archivo de "Robot Face.py" como prueba de la generación correcta de código G de la librería.

## 5.1.4 ULTIMA VERSIÓN

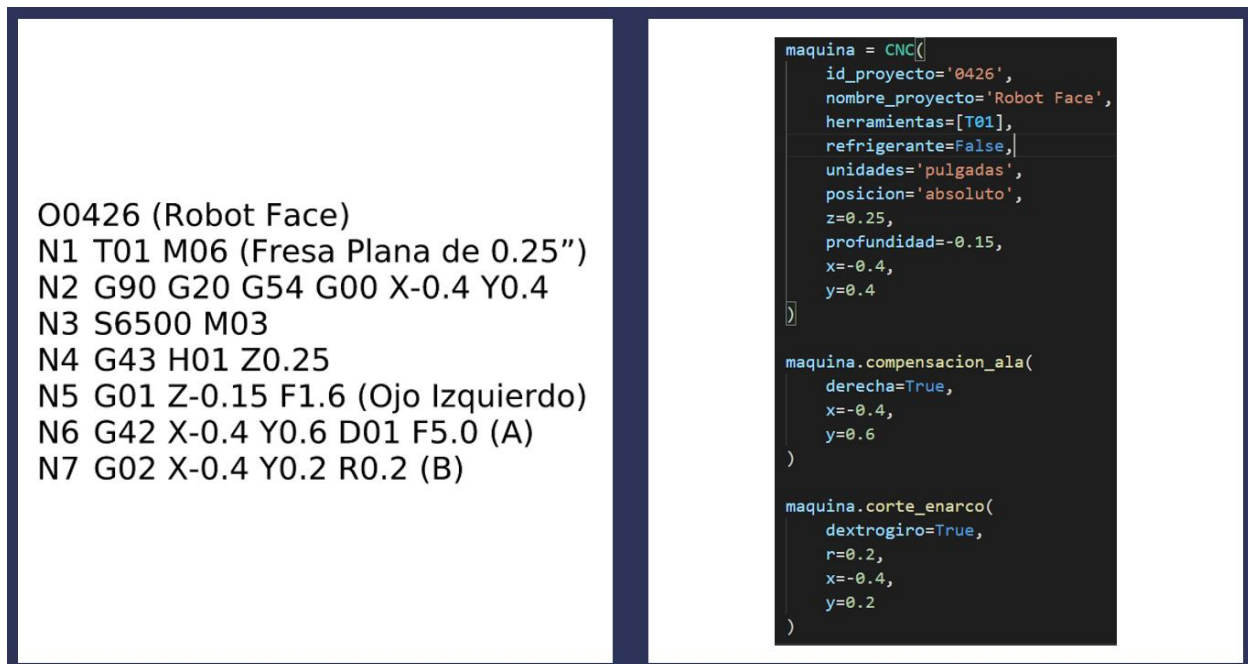
La última versión cuenta con todas las variables en un solo idioma, se agregó comandos como: el final de subrutina, subrutina interna, final de taladrado, se modificó el comando de movimiento por el uso del comando de ciclo de taladrado.

## 5.2 PROGRAMACIÓN AMIGABLE DEL GENERADOR

### 5.2.1 COMPARACIÓN DE COMANDOS EN CÓDIGO G Y CON EL USO DE LA LIBRERÍA DE PYTHON.

Al observar la ilustración 11 a la izquierda podemos apreciar que el código G se basan en letras y números al igual que el sistema numérico Hexadecimal. A simple vista no se aprecia bien el cuales son los comandos que a ejecutar.

Por otro lado, en el lado derecho de la ilustración 11, podemos observar que el entorno de Python hace que se nos haga más fácil leer el código que se ejecutara en ese momento.



**Ilustración 11 Comparación de lenguajes de programación. Preparación de la Máquina, Compensación de corte y corte en arco**

Fuente: (Elaboración propia)

## 5.3 DOCUMENTACIÓN DEL GENERADOR

Enlace de la documentación: <https://github.com/Nekrox00111/Gcode#readme>

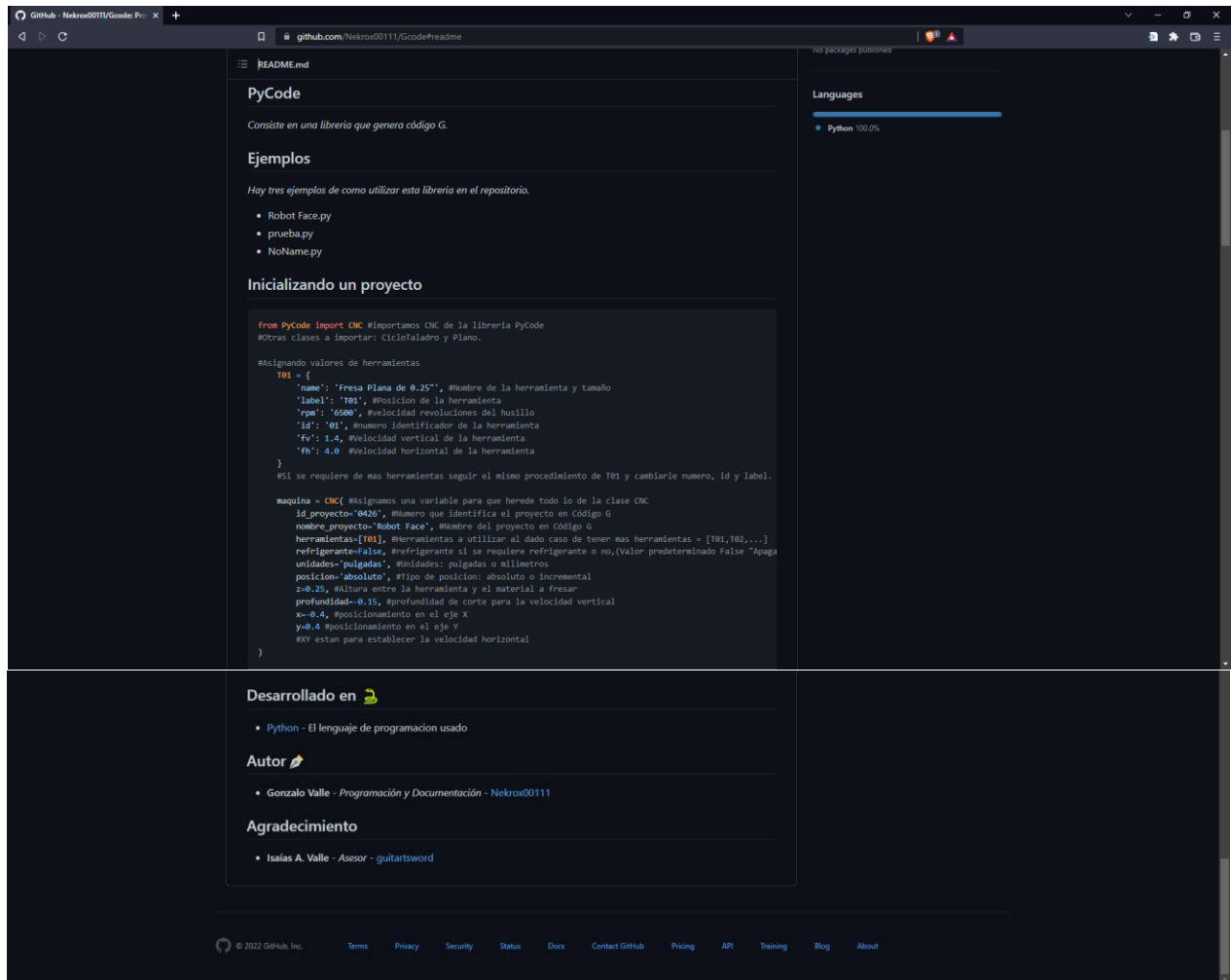


Ilustración 12 Documentación elaborada en GitHub

Fuente: (Elaboración Propia)

## 5.4 DOCUMENTACIÓN ELABORADA PARA EL PROYECTO

```
## PyCode

_Consiste en una Libreria que genera Los códigos G básicos para una Fresadora
CNC._

## Ejemplos

_Hay tres ejemplos de como utilizar esta libreria en el repositorio._

* Robot Face.py
* prueba.py

## Inicializando un proyecto

``` python
from PyCode import CNC #importamos CNC de la libreria PyCode
#Otras clases a importar: CicloTaladro y Plano.

#Asignando valores de herramientas
T01 = {
    'name': 'Fresa Plana de 0.25"', #Nombre de la herramienta y tamaño
    'label': 'T01', #Posicion de la herramienta
    'rpm': '6500', #velocidad revoluciones del husillo
    'id': '01', #numero identificador de la herramienta
    'fv': 1.4, #Velocidad vertical de la herramienta
    'fh': 4.0 #Velocidad horizontal de la herramienta
}
#Si se requiere de mas herramientas seguir el mismo procedimiento de T01 y
cambiarle numero, id y label.

maquina = CNC( #Asignamos una variable para que herede todo lo de la clase
CNC
    id_proyecto='0426', #Numero que identifica el proyecto en Código G
    nombre_proyecto='Robot Face', #Nombre del proyecto en Código G
    herramientas=[T01], #Herramientas a utilizar al dado caso de tener mas
herramientas = [T01,T02,...]
    refrigerante=False, #refrigerante si se requiere refrigerante o no,(Valor
predeterminado False "Apagado")
    unidades='pulgadas', #Unidades: pulgadas o milímetros
    posicion='absoluto', #Tipo de posicion: absoluto o incremental
    z=0.25, #Altura entre la herramienta y el material a fresar
    profundidad=-0.15, #profundidad de corte para la velocidad vertical
    x=-0.4, #posicionamiento en el eje X
```

```

        y=0.4 #posicionamiento en el eje Y
        #XY estan para establecer la velocidad horizontal
    )
    ...

## Tipos de Cortes
**Siguiendo el mismo programa de la seccion de "Inicializando un proyecto"**

    ... python

maquina.corte_lineal(
    x=1.5, #coordenada final del corte en X
    y=3.2 #coordenada final del corte en Y
)

maquina.corte_enarco(
    dextrogiro=False,#Corte en contra de las agujas del reloj, valor
predeterminado True (Con las agujas del reloj)
    x=3.5, #coordenada final del corte en X
    y=5.2, #coordenada final del corte en Y
    r=2.0 #radio de 2 unidades
)

maquina.corte_vertical(
    profundidad = -0.15 #profundidad de corte, normalmente se utiliza con el
metodo de mover.
)
...

## Mover
    ... python

maquina.mover(
    x=3, #coordenada final del corte en X
    y=0, #coordenada final del corte en Y
    cambiar_plano=False#Argumento opcional, predeterminado: cambiar_plano=False.

    fuera_material=False#Argumento opcional, predeterminado:
fuera_material=False. Si se requiere salir del material, valor de Z definida en
la seccion "Inicializando un proyecto"
)
...

*NO SON OPCIONALES cuando se utiliza el metodo "ciclo_de_taladro"*

## Cambio de herramienta

```

```

''' python

maquina.change_tool(
    x=0.5, #posicion X del siguiente corte
    y=1.5, #posicion Y del siguiente corte
    herramientas = 'T50', #buscador a traves de id, nombre de la herramienta o el
label
    refrigerante = True #Se requiere refrigerante
)

'''

## Vaciados
''' python

maquina.empty_circle(
    x=1, #Coordenada en X del entro del vaciado
    y=2, #Coordenada en Y del entro del vaciado
    profundidad=-0.15, #Profundidad del vaciado
    r=3/8, #Radio del vaciado
    dextrogiro=False#Corte en contra de las agujas del reloj, valor
predeterminado True (Con las agujas del reloj)
)

maquina.empty_spiral(
    x=1, #Coordenada en X del entro del vaciado
    y=2, #Coordenada en Y del entro del vaciado
    profundidad=-0.15, #Profundidad del vaciado
    I_radio_Arco=0.05, #Radio del primer arco
    Q = 0.05, #Distancia entre recorridos del espiral
    #I=Q <= Diametro REAL de la herramienta de corte
    K_radio = 3/8, #Radio del Vaciado
    dextrogiro=False,#Corte en contra de las agujas del reloj, valor
predeterminado True (Con las agujas del reloj)
)

'''

## Compensacion a la derecha/izquierda
''' python

maquina.compensacion_ala(
    derecha=False, #Tipo de compensacion de corte. Valor predeterminado: True,
argumento opcional si se requiere que sea a la derecha.
    x=-0.4, #Coordenada X del inicio de la compensacion de corte
    y=0.6 #Coordenada Y del inicio de la compensacion de corte
)

```



```

'''
## Ciclo de taladrado
*maquina.mover se utiliza con cambiar_plano y fuera_material.*
''' python

maquina.ciclo_de_taladrado(
    plano_taladro= # Plano.retorno o Plano.inicio
    r= # Altura del plano de retorno
    tipo_taladro= # CicloTaladro.centro, CicloTaladro.normal,
CicloTaladro.profundo
    x= # Coordenada X del T1
    y= # Coordenada Y del T1
    z= # Profundidad de los taladrados del ciclo -2mm, -3mm, -0.06", -0.1
    p= #se utiliza cuando el tipo_taladrado = CicloTaladro.centro
    #No se requiere en otro tipo de taladrado
    #Tiempo en segundos que la herramienta espera cuando ha llegado a la
profundidad especificada, antes de moverse al siguiente punto
    q= #se utiliza cuando el tipo_taladrado = CicloTaladro.profundo
    #No se requiere en otro tipo de taladrado
    #Diferencial de profundidad que se utiliza para descender hasta llegar a la
profundidad Z.
)
maquina.mover(...)
maquina.final_taladrado() #solo se requiere cuando el ciclo se termina
'''

## Subrutinas
''' python

maquina.subrutina_interna(
    P=25 #Linea donde se encuentra la subrutina
)

maquina.subrutina_externa(
    P='0004' #Numero identificador de proyecto
)

maquina.final_subrutina() #Fin de subrutina externa o interna.
'''

## Finalizacion de un proyecto
*Orden recomendado para el uso de subrutinas internas por la señalizacion de
cantidad de lineas*
''' python

```

```


maquina.final() #Códigos necesarios para finalizar el programa

maquina.exportar(file_name="Robot Face.nc") #Exportar en un archivo con extension
.gcode, .nc, .txt, entre otros.
print(maquina.code) #Exponer en texto plano el código G generado.
...
*También existe la opción de alternar el print primero.*


*No muestra la cantidad de líneas de código.*
``` python

maquina.final() #Códigos necesarios para finalizar el programa
print(maquina.code) #Exponer en texto plano el código G generado.

maquina.exportar(file_name="Robot Face.nc") #Exportar en un archivo con extension
.gcode, .nc, .txt, entre otros.
...
*Existe la opción de solo exportar a un archivo como solo exponerlo en texto
plano*

## Desarrollado en 

* [Python](http://www.python.org) - El lenguaje de programación usado

## Autor 

* Gonzalo Valle - Programación y Documentación -
[Nekrox00111](https://github.com/Nekrox00111)

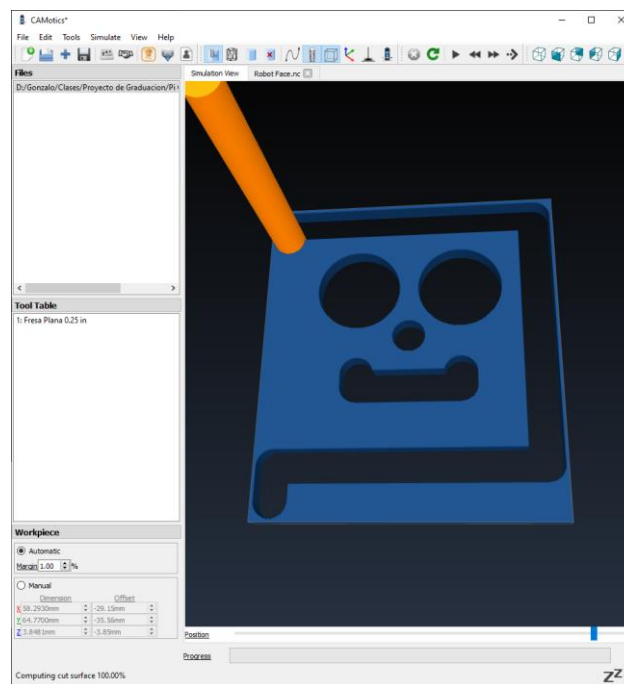
## Agradecimiento

* Isaías A. Valle - Asesor -
[guitartsword](https://github.com/guitartsword)

```

## 5.5 PRUEBAS EN SIMULADOR DE CNC

Al utilizar el generador tiene un comando para poder exportar el archivo con el nombre y con una extensión gcode como predeterminado, tiene la opción también de ponerle su propia extensión preferiblemente con extensiones como: txt, nc, ngc, mpt, mpf. En la ilustración 13 muestra una prueba de nuestro código exportado probado en el simulador de CAMotics generado el "Robot Face.nc". En la ilustración 13 se analiza que la realización del código generado fue exitosa, por el hecho de que el código se ha ejecutado bien.



**Ilustración 13 Simulación del archivo "Robot Face.nc" generado del "Robot Face.py"**

Fuente: (Elaboración Propia)

## VI CONCLUSIONES

- Se logró generar el código G mediante el lenguaje de programación Python creando una librería específica para poder traducir los códigos G más comunes para fresadora CNC.
- Se desarrolló un sistema de programación en Python de tal manera que fuese más amigable y entendible para el programador de forma que no sea confuso utilizando la filosofía código limpio.
- Se creó una documentación bien organizada de la función de cada uno de los métodos creados en la librería en un repositorio de GitHub para lograr utilizarla correctamente y sin problemas.
- Se desarrolló una función específica para poder exportar nuestro código G generado por la librería en un archivo legible para una operación en una fresadora CNC.
- Se expuso un texto plano a través de una de las funciones principales la cual es llamada "print" siempre utilizando el import de la librería.
- Se lograron exitosas pruebas ya que no fallaron en la exportación y generación de Python con el texto plano generado en código G mediante el software CAMotics para fresadora de CNC.

## VII RECOMENDACIONES

- Se recomienda tener experiencia en el lenguaje de Python, de igual manera conocer a detalle los comando en Código G.
- Se recomienda conocer y aplicar la filosofía de código limpio.
- Al realizar una documentación es recomendable analizar los métodos establecidos en la librería para poder describir su función a su cabalidad.
- Se recomiendo exportar en un archivo capaz de ser legible para la máquina CNC deseada.
- Se recomienda que al exponer el código G el programador solo requiera de utilizar print para su mayor facilidad.
- Se recomienda el uso de un simulador multiplataforma para realizar las pruebas ya que se pueden realizar en distintos sistemas operativos y poder analizar si tiene alguna falla nuestro programa.

## VIII BIBLIOGRAFÍA

Asociación Española de Fabricantes de Máquinas-Herramienta. (2022, abril 20). *¿Qué es una máquina-herramienta?* Canales sectoriales Interempresas. <https://www.interempresas.net/Deformacion-y-chapa/Articulos/30677-Que-es-una-maquina-herramienta.html>

Bitfab. (s. f.). Mega guía de fresadoras – Qué son, tipos y características. *Mega guía de fresadoras – Qué son, tipos y características*. <https://bitfab.io/es/blog/fresadoras/>

CAMotics. (2019). *Camotics*. <https://camotics.org/>

Carlos Eduardo Plasencia Prado. (2021, septiembre). Diccionarios en Python. *Diccionarios en Python*. <https://devcode.la/tutoriales/diccionarios-en-python/>

De máquinas y herramientas. (2012, julio 5). *¿Qué son y cómo se usan las Fresas?* <https://www.demaquinasyherramientas.com/mecanizado/fresas-tipos-y-usos>

De máquinas y herramientas. (2015, marzo 7). *¿Qué tener en cuenta al elegir una fresadora? ¿Qué tener en cuenta al elegir una fresadora?* <https://www.demaquinasyherramientas.com/mecanizado/elegir-comprar-fresadora>

devCamp. (2020). *Qué es una librería en programación*. <https://devcamp.es/que-es-libreria-programacion/#:~:text=%C2%BFQu%C3%A9%20es%20una%20librer%C3%ADa%20en%20programaci%C3%B3n%3F&text=En%20este%20sentido%2C%20una%20librer%C3%ADa,llanamente%2C%20es%20un%20archivo%20importable.>

GitHub, Inc. (2022). *Github*. GitHub Docs. <https://docs.github.com/es/get-started/quickstart/hello-world>

Haas F1 Team. (2022a). *Qué son los códigos G*. ¿Que son los codigos G? <https://www.haascnc.com/es/service/service-content/guide-procedures/what-are-g-codes.html>

Haas F1 Team. (2022b). *Qué son los códigos M*. ¿Que son los codigos M? <https://www.haascnc.com/es/service/service-content/guide-procedures/what-are-m->



Stanser. (2018, septiembre 26). *¿CÓMO FUNCIONAN LAS FRESADORAS CNC?* Stanser Maquinaria CNC. <https://www.stanser.com/como-funcionan-las-fresadoras-cnc/>

Steve F. Krar. (2001). *Tecnología de las Máquinas Herramienta* (Quinta). McGRAW-HILL.

Sublime Text, un sofisticado editor de código multiplataforma. (2012, febrero 9). *Sublime Text, un sofisticado editor de código multiplataforma.* <https://www.genbeta.com/herramientas/sublime-text-un-sofisticado-editor-de-codigo-multiplataforma>