# Instrucciones para el uso de los programas desarrollados en Python

Los 3 programas desarrollados en el Python fueron hechos con la versión de Python 3.9.7 la cual se puede descargar desde el sitio oficial o desde la tienda de Microsoft.

Para que cada uno de los programas funcione de manera correcta, requiere que Python tenga instalada las siguientes 7 librerías:

* Numpy
* Pandas
* Matplotlib
* Sklearn
* Scipy
* Sympy
* Tkinter

Para poder instalar estas librerías, es preciso dirigirse al menú de búsqueda y escribir cmd:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Se abre esta aplicación y aparecerá el siguiente cuadro:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

En este cuadro se escriben las siguientes sentencias para instalar cada librería y después de cada sentencia se da un enter, cada paquete se debe instalar de uno en uno, es decir se debe escribir “pip install numpy” y se da enter, después aparecerá una barra que se esta descargando esta librería, una vez que esta descargada aparecerá que ya esta hecho el proceso, cuando esto ocurra, escribir en la misma ventana CMD “pip install pandas” y se instalara la librería Pandas y así con todas las 7 librerías que requieren los programas para correr.

Las sentencias para escribir en CMD son las siguientes:

* pip install numpy
* pip install pandas
* pip install matplotlib
* pip install sklearn
* pip install scipy
* pip install sympy
* pip install tkinter

Una vez instalado estos programas, Python será capaz de correr todas las líneas de código de cada uno de los 3 programas desarrollados, para el caso del programa “Algoritmo de Lane” Python demora un tiempo estimado de 3 a 4 minutos, pero para el “algoritmo GRG Iterativo” y “algoritmo GRG Multi Año” Python puede demorar mas de 2 horas, por lo cual, para acelerar la velocidad y tiempo de procesamiento se recomienda la instalación de un software que ayuda a Python, mejorando la velocidad cuando la ejecución de los códigos de programación son muy extensos, para ello se ocupara el software Visual Studio Code, el cual puede ser descargado desde el siguiente link:

<https://code.visualstudio.com/>

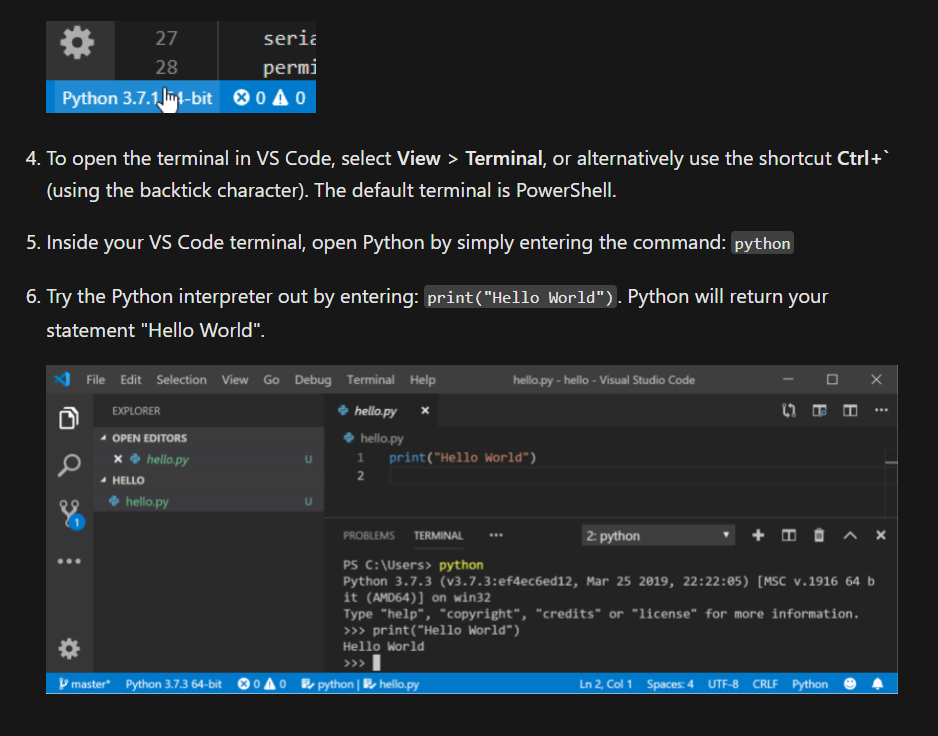
Este software ayuda muchísimo a acelerar la velocidad de procesamiento de los códigos de programación, por ejemplo, utilizando este software para correr el programa “algoritmo GRG Iterativo” y “algoritmo GRG Multi Año” este nuevo software se demora solo “10 MINUTOS” utilizando un pc con un procesador Intel i5 windows 10 con 2.3 GHz y con 8 Gb de RAM instalada.

Las instrucciones para instalar este software de ayuda y aceleración para códigos de programación están en el siguiente link:

<https://docs.microsoft.com/en-us/windows/python/beginners>

Texto

Descripción generada automáticamente



La idea es que, una vez instalado este nuevo software, a este software se le debe añadir un lenguaje de programación y allí uno debe buscar en la lupa y escribir “Python” donde aparecerá el lenguaje de programación listo para ser incorporado a este nuevo software.

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Una vez instalado y listo el software de ayuda y aceleración para los programas en Python, es posible correr de forma más amena y adecuada cada uno de los 3 programas desarrollados.

## Algoritmo de Lane

Cuando se corra este algoritmo aparecerá la siguiente ventana interactiva donde se deben colocar los parámetros económicos y la distribución de leyes del elemento de interés.

Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente

En esta pantalla aparecen una serie de datos que el programa necesita para ser corrido, como parámetros económicos y la distribución de leyes, por ejemplo, para el desarrollo del trabajo de titulación se contemplaron los siguientes datos inputs:

|  |  |  |
| --- | --- | --- |
| Ley Mínima % | Ley Máxima % | Tonelaje Millones de Toneladas |
| 0 | 0,2 | 60 |
| 0,2 | 0,4 | 52 |
| 0,4 | 0,6 | 23 |
| 0,6 | 0,8 | 17 |
| 0,8 | 1 | 12 |
| 1 | 1,2 | 9.7 |
| 1,2 | 1,4 | 6.7 |
| 1,4 | 1,6 | 8.9 |
| 1,6 | 1,8 | 4 |
| 1,8 | 3 | 5.6 |

|  |  |  |
| --- | --- | --- |
| Parámetros Económicos | Valor | Unidad |
| Costo Mina | 3,5 | usd/ton de material |
| Costo Planta | 6,5 | usd/ton de mineral |
| Costo de Comercialización Cátodo | 1.100 | usd/ton de cobre fino |
| Precio del Cobre | 300 | cusd/lb de cobre |
| Tasa de Descuento | 10 | % |
| Costo Fijo Anual | 20.000.000 | usd/año |
| Costo de Rehabilitación del Botadero | 0,027306 | usd/ton de estéril |
| Capacidad de la Planta Concentradora | 15.000.000 | Ton de Mineral |
| **Parámetros Económicos** | **Valor** | **Unidad** |
| Capacidad de producción de cátodos de la refinería | 100.000 | Ton de Cobre Refinado |
| Recuperación de la Planta Concentradora | 88 | % |
| Ley del Concentrado de Cobre | 30 | % |
| Perdida Metalúrgica de la Fundición y Refinería | 3,5 | % |
| Cargo TC | 126 | usd/TMS de concentrado |
| Cargo RC | 12,6 | cusd/lb de cobre |

Cabe destacar que cada uno de los valores numéricos ingresados al sistema debe ser con su separador decimal en punto y no con la coma, ya que Python conoce que el separador decimal es el punto y no la coma, por lo que **TODOS LOS DATOS INGRESADOS SI SON DECIMALES, SU SEPARADOR ES EL PUNTO DECIMAL** **Y NO LA COMA**

Otra consideración importante es que, en el menú inicial, pregunta si deseo incorporar las economías de escalas, si quiero incorporar una diferenciación entre el costo de extracción del mineral según la tabla inputs es 3,5 usd/ton de mineral, y el costo de extracción del estéril, debo escribir en este cuadrado la palabra SI en mayúscula o minúscula.

Texto

Descripción generada automáticamente

Y después el programa me pedida que ingrese el costo de extracción para el estéril. Que para el caso del trabajo de titulación se ha asignado de 2.5 usd/ton de estéril.

Texto

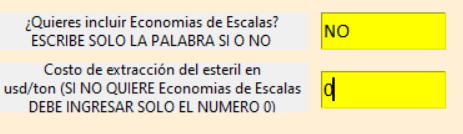
Descripción generada automáticamente

En el caso de que no deseo incluir esta economía de escalas, y que el costo de extracción del material (mineral más estéril) solo posee un único costo, se debe escribir la palabra NO en mayúscula o minúscula.

Texto

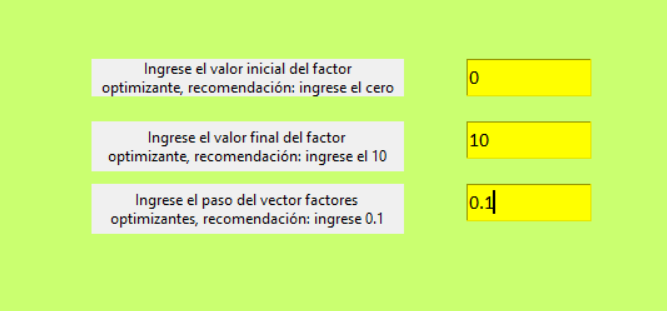
Descripción generada automáticamente con confianza media

Y SI SE COLOCO LA OPCION “NO” SE DEBE ESCRIBIR UN COSTO DE EXTRACCION DEL ESTERIL DE CERO, NO DEJAR ESA CASILLA EN BLANCO.



## Algoritmo GRG Iterativo

Las instrucciones para este programa son las mismas que para el programa “Algoritmo de Lane” anterior, solo con la salvedad de que ahora el usuario debe ingresar los datos del vector optimizante que hace crecer el VAN del “Algoritmo de Lane” siempre que sea posible, ya que existen casos que debido a la distribución de leyes y a los parámetros que este programa “Algoritmo GRG Iterativo” no podrá hacer crecer mas el VAN del “Algoritmo de Lane”, ya que no es posible matemáticamente.



La imagen anterior ilustra los mejores valores que puede tener el vector factores optimizante y los que recomienda su creador, ya que debe partir desde el 0 hasta llegar al 10 y con pasos de 0.1, si el paso fuese de 0.01 la solución sería más exacta, pero se demoraría 100 veces mas en llegar a la solución. (recordemos que este algoritmo parte de la solución que ofrece Lane y esta solución la hace crecer hasta encontrar aquel valor especifico (factor optimizante) que hace que el VAN sea máximo.

## Algoritmo GRG Multi Año

Las instrucciones para este programa son las mismas que para el programa algoritmo de Lane, solo que ahora se agrega un apartado nuevo en donde se debe colocar los valores numéricos del factor optimizante, así como decidir el tipo de solución que se desea obtener.

Texto

Descripción generada automáticamente con confianza media

Del mismo modo que como ocurrió con el “Algoritmo GRG Iterativo”, se debe colocar los valores numéricos del factor optimizante, solo que este vez el creador recomienda que este vector comience desde el cero hasta llegar al numero 30 con pasos de 0.01, esto es debido a que la solución de arranque es la ley de corte marginal y esta ley por lo general es menor que la ley de corte ofrecida por la solución de Lane y con la que comienza el “Algoritmo GRG Iterativo”, por lo tanto este vector debe abarcar un mayor número de elementos numéricos, el paso es mas pequeño, debido a que se evalúa en cada año cada una de estas leyes de corte calculadas con el factor optimizante a diferencia del “Algoritmo GRG Iterativo” donde el factor optimizante, se hace variar para cada plan pero año a año se mantiene fijo.

Además, es posible observar que existe un cuarto ingreso de dato, el cual, corresponde al tipo de plan que desea el usuario, el cual puede ser de dos tipos:

* Maximizando el VAN Total: si el usuario desea encontrar la solución de la política de leyes de corte que permitan consumir todos los recursos mineros, pero que al mismo tiempo entreguen el mayor VAN posible, debe seleccionar esta opción. (pero no asegura que el vector de leyes de corte optimo sea constante o uniforme), para esto debe escribir la letra A.
* Priorizando el Movimiento Mina Parejo: si el usuario desea encontrar una solución que entregue el vector de leyes de corte optimas, pero que al mismo tiempo el movimiento mina sea constante a lo largo de todo el periodo de producción hasta descender cuando se agoten los recursos, debe escribir la letra “B”. esta opción no asegura que el VAN obtenido sea el mayor, pero si que el movimiento mina sea parejo.