

MANUAL DE USUARIO
SOFTWARE FOOTPRINT FINDER UDD

FACULTAD DE INGENIERÍA
CARRERA INGENIERÍA CIVIL EN MINERÍA

PROFESOR

Jorge Contreras R.

ALUMNOS

Nicolás Ayarza D.

Francisco Aguirre C.

Noviembre 2019

Contenido

1. Agradecimientos	4
2. Introducción y objetivos del proyecto	5
2.1. Introducción	5
2.2. Objetivo General.....	5
2.3. Objetivos Específicos	5
3. Metodología de cálculo	6
3.1. Valorización Nominal de las UBC.....	6
3.1.1. Ingresos.....	6
3.1.2. Gastos Operación	7
3.2. Periodo de extracción.....	7
3.3. Beneficio Actualizado	7
3.4. Gasto de preparación de área y valorización acumulada por UBC	7
3.5. Altura Económica Máxima de la Columna.....	8
4. Instrucciones de ejecución del programa	9
4.1. Softwares requeridos	9
4.1.1. NetBeans	9
4.1.2. OpenScad.....	9
4.2. Instalación del código	9
4.3. Modelo de Bloques.....	11
4.4. Ejecución del programa	12
4.5. Salidas de NetBeans	14
4.6. Visualización en OpenScad	15
5. Código del algoritmo	18
5.1. Captura de información y valorización nominal.....	18
5.1.1. Instrucciones para llamar a las bibliotecas de JAVA para ser usados en el programa	18
5.1.2. Instrucciones para activar la interfaz y leer desde el teclado el nombre del modelo de bloques	18
5.1.3. Separar las filas del modelo de bloques	18
5.1.4. Instrucciones para separar en columnas.....	18
5.1.5. Contador y asignación de puntos de extracción	19
5.1.6. Creación del archivo ID_PUNTOS.txt.....	19
5.1.7. Valorización Nominal.....	19
5.2. Máx VAN y Altura Columna	20
5.2.1. Parámetros de entrada.....	20
5.2.2. Contar número de Filas	21
5.2.3. Separar filas	21
5.2.4. Contar número de caracteres por fila	21
5.2.5. Contar caracteres por fila (Para efectos de buscar el Separador).....	21
5.2.6. Matriz de datos, utilizando los caracteres por fila, para efectos de lectura	21
5.2.7. Valor actualizado por UBC asumiendo un periodo fijo de extracción por bloque.....	22
5.2.8. Inicialización de Variables.....	22
5.2.9. VAN Acumulado por punto de extracción.....	22
5.3. Búsqueda de índice que maximiza VAN en columna	22

Tabla de Ecuaciones

Ecuación 3-1 Cobre fino mina UBC.....6
 Ecuación 3-2 Cobre fino a concentrado UBC.....6
 Ecuación 3-3 Ingresos por UBC.....6
 Ecuación 3-4 Gasto operación mina y planta7
 Ecuación 3-5 Beneficio nominal UBC.....7
 Ecuación 3-6 Beneficio Actualizado UBC.....7
 Ecuación 3-7 Gasto por concepto de preparación de área7

Tabla de Ilustraciones

Ilustración 3-1 Resumen metodología – Fuente (Contreras, 2017)6
 Ilustración 3-2 Altura económica que maximiza el van para 3 puntos de extracción.....8
 Ilustración 4-1 Despliegue del menú inicial de Netbeans 10
 Ilustración 4-2 Selección del nombre del proyecto en NetBeans 10
 Ilustración 4-3 Despliegue de la plataforma de trabajo de NetBeans 11
 Ilustración 4-4 Inserción del código del programa 11
 Ilustración 4-5 Ejemplo de formato del modelo de bloques..... 12
 Ilustración 4-6 Carpeta Raíz del Proyecto 12
 Ilustración 4-7 Ejecución del código 13
 Ilustración 4-8 Interfaz con el usuario NetBeans 13
 Ilustración 4-9 Ingreso de parámetros técnicos y económicos 14
 Ilustración 4-10 Archivos de Salida del Programa 14

1. Agradecimientos

En primer lugar, quisiera dar las gracias al equipo de Ingeniería Civil en Minería de la Universidad del Desarrollo, por dar el espacio para desarrollar este humilde pero útil programa para los alumnos de los últimos años de carrera. También agradecer al Centro de Innovación Docente de la Universidad por el financiamiento y la excelente iniciativa de fomentar este tipo de concursos para mejorar la docencia.

Agradecer a mis alumnos Nicolás Ayarza y Francisco Aguirre por las horas dedicadas a este proyecto.

El código está abierto y permite ir mejorando esta primera versión, por lo cual errores o simplificaciones del código, por favor hacerlos llegar.

2. Introducción y objetivos del proyecto

2.1. Introducción

El proyecto desarrollo Software “Footprint Finder UDD”, corresponde a la implementación y programación en software de código libre (gratuitos), de algoritmos que se encuentran en el estado del arte para la selección económica de reservas mineras en minas explotadas por el método de Panel Caving. (Método más productivo y futuro de toda mina a cielo abierto que desee migrar a minería Subterránea).

El concepto base, es poder a partir de un modelo de bloques diluido dedicar más tiempo en análisis de resultados, que en el desarrollo de los cálculos que actualmente se usan en plataformas como Excel. Por otra parte, la posibilidad de desplegar en 3D los resultados, permitirán al estudiante obtener conclusiones más fácilmente que en un modelo 2D. Estos softwares hoy existen en la industria, pero con un valor por sobre los US\$ 40.000 al año de licencias, y sin la posibilidad de modificar el código para poder entender lo que está detrás del software (Efecto Caja Negra). Es por lo anterior que se desea desarrollar una herramienta gratis, fácil y que permita a los alumnos poder dedicar tiempo al análisis, en vez de solamente realizar cálculos.

2.2. Objetivo General

Desarrollar un software de selección económica de reservas para minas subterráneas explotadas por block/panel Caving.

2.3. Objetivos Específicos

- ❖ Calcular la altura económica que maximiza el VAN (Valor Actual Neto) para cada punto de extracción en función de parámetros técnicos y económicos
- ❖ Minimizar el tiempo de cálculo en diferentes actividades docentes en post de mayor tiempo en análisis de resultados
- ❖ Analizar los resultados como habilidad superior de aprendizaje, por sobre el cálculo de ecuaciones
- ❖ Establecer un cambio en la metodología de trabajo en clases bajo el uso de nuevas herramientas computacionales.

3. Metodología de cálculo

La Ilustración 3-1 corresponde a un resumen de la lógica de valorización y definición de la altura que maximiza el VAN (Valor Actual Neto) para cada columna de extracción del modelo de bloques.

Leyes de Cu por UBC	Valorización KUS\$	Periodo de Extracción	Valor Actualizado KUS\$	Valor Acumulado KUS\$
0.41%	-122	2	-104	-97
0.55%	-29	2	-24	6
0.6%	5	2	4	31
0.61%	12	1	10	27
0.77%	118	1	109	16
0.92%	218	1	202	-93
1.02%	285	1	264	-295
1.17%	385	0	385	-559
1.25%	439	0	-944	-944
	<ul style="list-style-type: none"> Ingresos Costos 	$1/(1+tasa)^i$	Inversión agregada en UBC 1	MAX VAN 31 KUS\$ - 140 [m]

Ilustración 3-1 Resumen metodología – Fuente (Contreras, 2017)

Es posible describir 3 aspectos generales:

- ❖ Valorización Nominal de las UBC's (Beneficio económico de explotar un bloque)
- ❖ Valorización Actualizada de las UBC's (Considera la temporalidad con las que saldrá una UBC, en función de su altura y parámetros tales como la velocidad de extracción, disponibilidad de área, etc.)
- ❖ Búsqueda del valor acumulado actualizado que maximiza el VAN (Valor Actual Neto), y por ende la altura de columna para dicho punto de extracción.

3.1. Valorización Nominal de las UBC

3.1.1. Ingresos

$$Fino Cu_{MINA}[t] = Ton_{UBC}[t] \times Ley_{Cu}[\%]$$

Ecuación 3-1 Cobre fino mina UBC

Ton_{UBC} : Tonelaje [t] de una UBC

Ley_{Cu} : Ley de Cobre en porcentaje

$$Fino Cu_{CONCENTRADORA}[lb] = Fino Cu_{MINA}[t] \times R_{Metalúrgica}[\%] \times 2.204,62[lb/t]$$

Ecuación 3-2 Cobre fino a concentrado UBC

$R_{Metalúrgica}$: Recuperación metalúrgica planta concentradora en porcentaje

$$Ingreso[KUS\$] = \frac{Fino Cu_{CONCENTRADORA}[lb] \times (P_{Cu}[US\$/lb] - D_{Cu}[US\$/lb])}{1.000}$$

Ecuación 3-3 Ingresos por UBC

P_{Cu} : Precio del Cobre en [US\$/lb]

D_{Cu} : Descuento a Concentrado en [US\$/lb]

3.1.2. Gastos Operación

$$Gastos[KUS\$] = \frac{Ton_{UBC}[t] \times (C_{Mina}[US\$/t] + C_{Planta}[US\$/t])}{1.000}$$

Ecuación 3-4 Gasto operación mina y planta

C_{Mina} : Costo Mina en [US\$/t]

C_{Planta} : Costo Planta en [US\$/t]

Beneficio Nominal

$$Beneficio\ UBC_i[KUS\$] = Ingresos[KUS\$] - Gastos[KUS\$]$$

Ecuación 3-5 Beneficio nominal UBC

3.2. Periodo de extracción

De acuerdo con la publicación de Contreras, Cornejo & Caviedes (2016) existen ecuaciones que gobiernan la altura máxima a extraer por periodo (año) para una mina explotada por panel Caving. En particular las características geotécnicas del macizo podrán imponer condiciones de “velocidades de extracción” diferenciadas en función de la altura de columna extraída con el objetivo de disminuir el riesgo sísmico asociada a roca primaria.

Para efectos de esta primera versión del programa, los valores vienen por defecto y corresponde a:

- ❖ **Año 0** : No hay extracción, solo preparación del punto de extracción
- ❖ **Año 1** : Extracción de las UBC's 1 y 2
- ❖ **Año 2** : Extracción de las UBC's 3,4 y 5
- ❖ **Del año 3 en adelante se extraerán 4 UBC's por periodo**

3.3. Beneficio Actualizado

En función del periodo de extracción definido para cada UBC de la columna, se determinará su valor actual neto mediante la siguiente expresión:

$$Beneficio\ Actualizado\ UBC_i\ [KUS\$] = \frac{Beneficio\ UBC_i[KUS\$]}{(1 + r)^j}$$

Ecuación 3-6 Beneficio Actualizado UBC

r : Tasa de descuento

j : Periodo de actualización (año), $j = 1, \dots, n$

3.4. Gasto de preparación de área y valorización acumulada por UBC

$$Gasto\ Preparación[KUS\$] = Área\ PE[m^2] \times Costo_{Incorporación}[KUS\$/m^2]$$

Ecuación 3-7 Gasto por concepto de preparación de área

Es importante destacar que el algoritmo supone que un PE de extracción se construye un año antes de iniciar la extracción.

3.5. Altura Económica Máxima de la Columna

$$MÁX: - \text{Gasto Preparación [KUS\$]} + \sum_{i=1}^x \text{Beneficio Actualizado } UBC_i \text{ [KUS\$]}$$

S. A.

$$x \leq \text{Altura Máxima}[m]/h_{UBC}$$

h_{UBC} : Altura del modelo de bloque (UBC)

La altura máxima corresponde a un parámetro que el usuario deberá definir

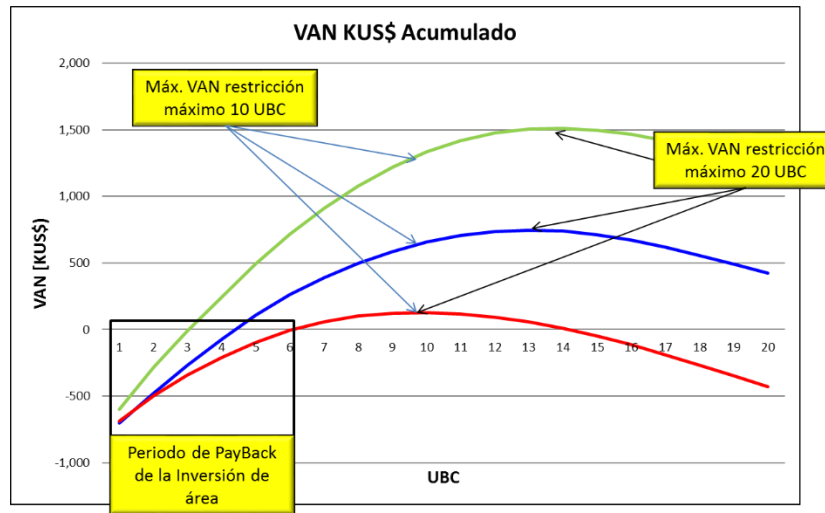


Ilustración 3-2 Altura económica que maximiza el van para 3 puntos de extracción

4. Instrucciones de ejecución del programa

4.1. Softwares requeridos

4.1.1. NetBeans

NetBeans IDE 8.2 Download

8.1 | 8.2 | Development | Archive

Email address (optional):

Subscribe to newsletters: Monthly Weekly

NetBeans can contact me at this address

IDE Language: Platform:

Note: Greyed out technologies are not supported for this platform.

Supported technologies *	Java SE	Java EE	HTML5/JavaScript	PHP	C/C++	All
<input type="checkbox"/> NetBeans Platform SDK	•	•				•
<input type="checkbox"/> Java SE	•	•				•
<input type="checkbox"/> Java FX	•	•				•
<input type="checkbox"/> Java EE		•				•
<input type="checkbox"/> Java ME						•
<input type="checkbox"/> HTML5/JavaScript		•	•	•		•
<input type="checkbox"/> PHP				•		•
<input type="checkbox"/> C/C++					•	•
<input type="checkbox"/> Groovy						•
<input type="checkbox"/> Java Card™ 3 Connected						•
<input type="checkbox"/> Bundled servers						•
<input type="checkbox"/> GlassFish Server Open Source Edition 4.1.1		•				•
<input type="checkbox"/> Apache Tomcat 8.0.27		•				•

Download x86 Download x86 Download x86 Download x86 Download x86

Download x64 Download x64 Download x64 Download x64

Free, 95 MB Free, 197 MB Free, 108 - 112 MB Free, 108 - 112 MB Free, 107 - 110 MB Free, 221 MB

<https://netbeans.org/downloads/8.2/>

4.1.2. OpenScad

OpenSCAD

The Programmers Solid 3D CAD Modeller

home about news downloads documentation gallery community github

Downloads

Mac OS X

Windows

Linux

Other Systems

Source Code

Development Snapshots

Prior Releases

Mac OS X

System requirements: Mac OS X 10.7 or newer

OpenSCAD 2015.03-3

64 bit intel - dmg package - 23 MB

OpenSCAD is also available on MacPorts:

```
$ sudo port install openscad
```

Windows

System requirements: Windows XP or newer on x86 32/64 bit

OpenSCAD 2015.03-2 x86 (32-bit) exe installer - 14 MB

OpenSCAD 2015.03-2 x86 (32-bit) zip package - 14 MB

OpenSCAD 2015.03-2 x86 (64-bit) exe installer - 14 MB

OpenSCAD 2015.03-2 x86 (64-bit) zip package - 14 MB

<http://www.openscad.org/downloads.html>

4.2. Instalación del código

Una vez instalado el software Netbeans IDE 8.2 o versión superior se desplegará el siguiente menú

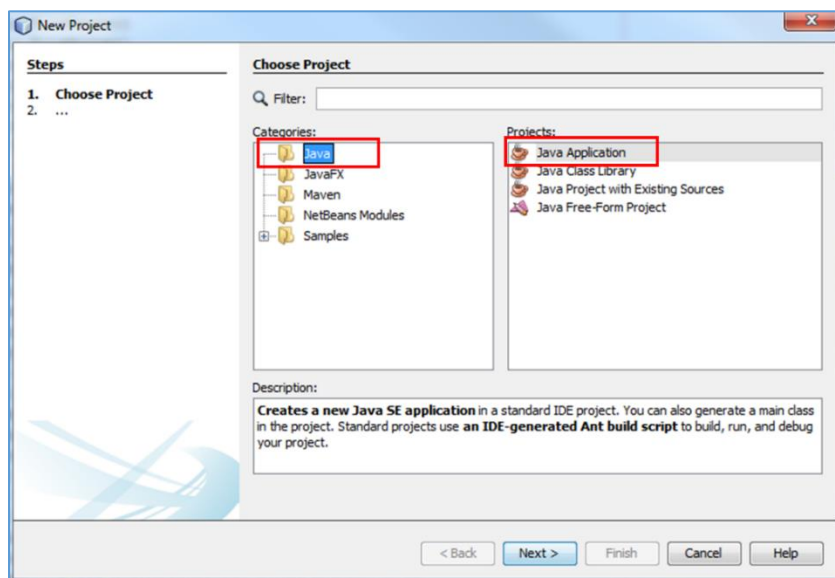


Ilustración 4-1 Despliegue del menú inicial de Netbeans

En “Categories” se debe escoger **Java** y en “Projects” **Java Application**

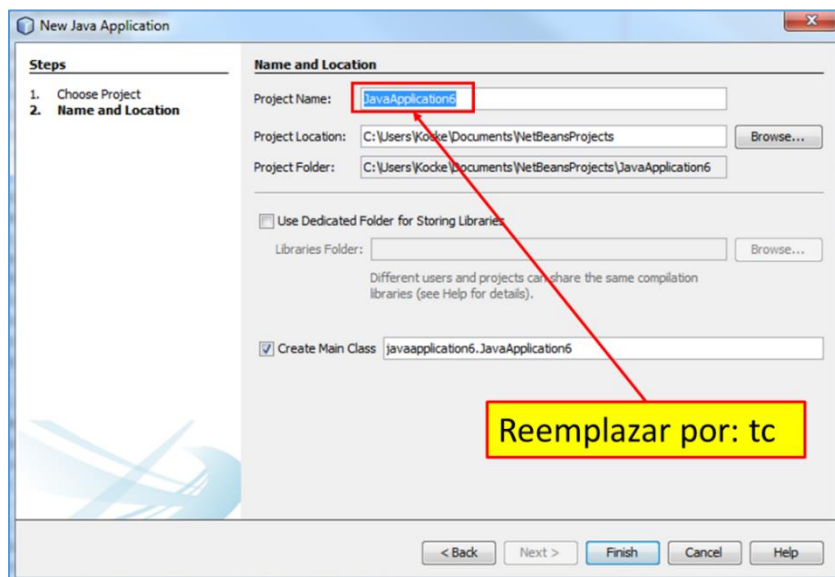


Ilustración 4-2 Selección del nombre del proyecto en NetBeans

En “Project Name” se debe reemplazar el nombre por **tc** (este nombre es básicamente debido a que se asignó estas siglas para la creación del proyecto y a su vez para la clase (en java) del proyecto – puede ser reemplazado, pero es importante modificar las partes del código que posean o llamen a tc).

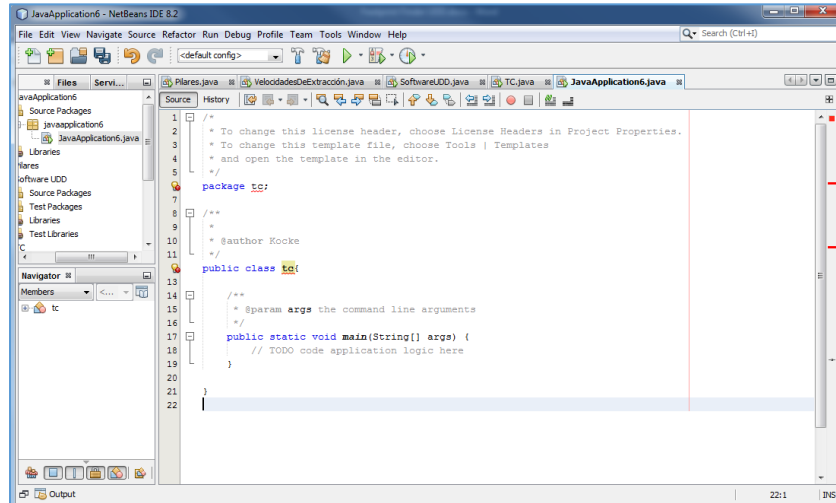


Ilustración 4-3 Despliegue de la plataforma de trabajo de NetBeans

Una vez realizado lo anterior, se desplegará la ventana de la Ilustración 4-4. Todo el contenido debe ser borrado, para ser reemplazado por el texto del archivo **Código.txt**

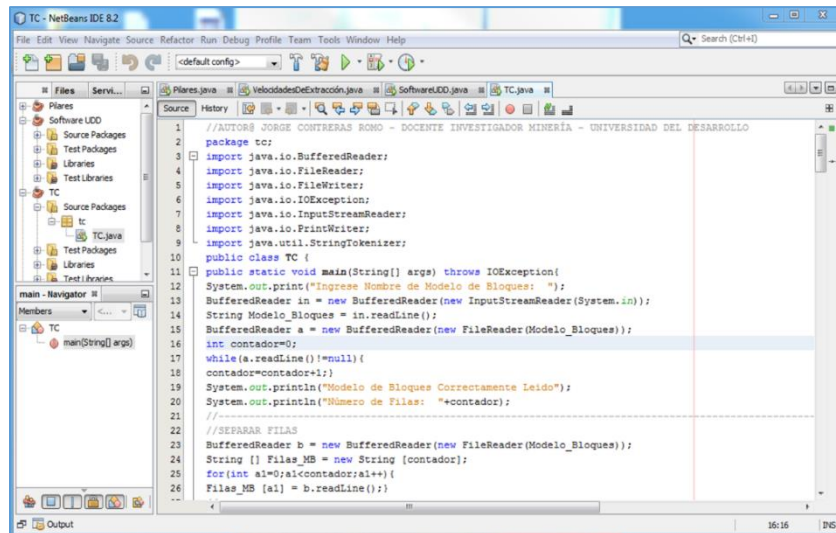


Ilustración 4-5 Inserción del código del programa

En este punto se debe guardar los cambios, y el programa queda listo para ser ejecutado

4.3. Modelo de Bloques

La estructura que debe ser utilizada en esta primera versión del **Footprint finder UDD** viene dada por los siguientes atributos:

ID_Pto; Coord_X; Coord_Y; Pos_Col; Tonelaje; CuT%

```

Modelo.txt: Bloc de notas
Archivo  Edición  Formato  Ver  Ayuda
ID_Pto;Coord_X;Coord_Y;Pos_Col;TonaLaje;Cut%
02_00N;321.9559;-74.1125;1;11720.19003;0.46891318
02_00N;321.9559;-74.1125;2;15626.92004;0.44484916
02_00N;321.9559;-74.1125;3;15626.92004;0.433814828
02_00N;321.9559;-74.1125;4;15626.92004;0.429189293
02_00N;321.9559;-74.1125;5;15626.92004;0.428942011
02_00N;321.9559;-74.1125;6;15626.92004;0.429440413
02_00N;321.9559;-74.1125;7;15626.92004;0.430374565
02_00N;321.9559;-74.1125;8;15626.92004;0.433306256
02_00N;321.9559;-74.1125;9;15587.61038;0.439067821
02_00N;321.9559;-74.1125;10;15552.49582;0.442934257
02_00N;321.9559;-74.1125;11;15526.90124;0.44131227
02_00N;321.9559;-74.1125;12;15501.30666;0.429069771
02_00N;321.9559;-74.1125;13;15475.71208;0.417821779
02_00N;321.9559;-74.1125;14;15450.1175;0.407624875
02_00N;321.9559;-74.1125;15;15424.52291;0.398829624
02_00N;321.9559;-74.1125;16;15398.92833;0.390572618
02_00N;321.9559;-74.1125;17;15373.33375;0.382920131

```

Ilustración 4-6 Ejemplo de formato del modelo de bloques

El atributo más relevante para el buen funcionamiento del algoritmo corresponde a la posición de la columna (Pos_Col). En la lógica de explotación de un Block/Panel Caving, la posición de columna da cuenta del orden de salida, desde el Piso del UCL a la topografía.

- ❖ La separación entre columnas del archivo debe ser con ; (punto y coma)
- ❖ No hay limitación respecto al número de decimales para las diferentes variables
- ❖ La ley de cobre debe estar en porcentaje

4.4. Ejecución del programa

- ❖ Dejar en la carpeta raíz del proyecto el archivo asociado al modelo de bloques. Para efectos de este manual utilizaremos el archivo **B.txt**

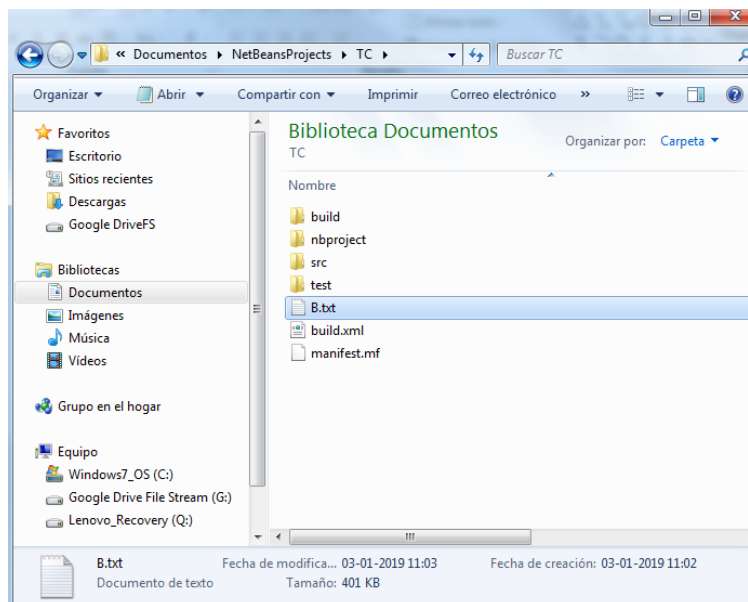


Ilustración 4-7 Carpeta Raíz del Proyecto

❖ Ejecutar el código del programa en NetBeans

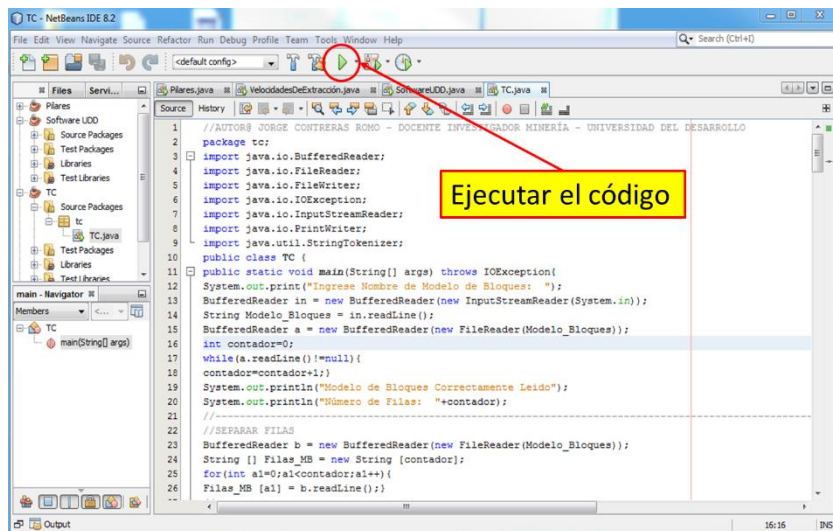


Ilustración 4-8 Ejecución del código

❖ Activación de la interfaz con el usuario

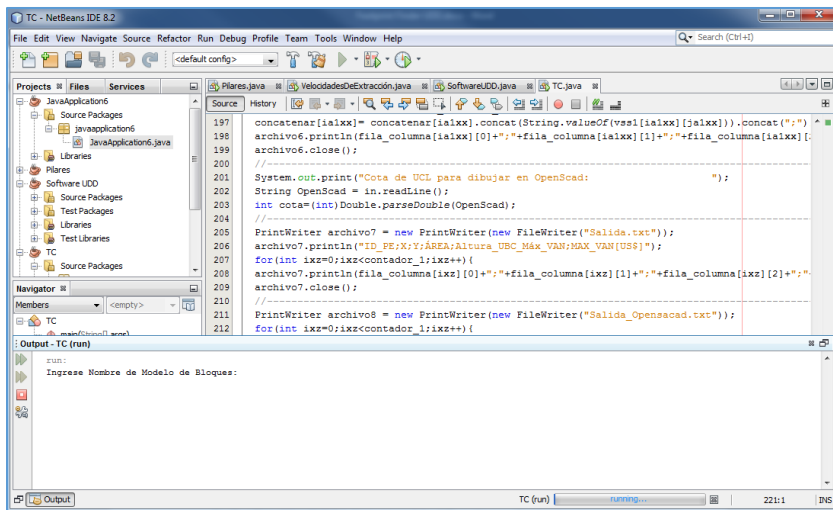


Ilustración 4-9 Interfaz con el usuario NetBeans

❖ Se debe ingresar el nombre del modelo de bloques con su extensión *.txt

```

Output - TC (run)
run:
Ingreso Nombre de Modelo de Bloques:                b.txt
Modelo de Bloques Correctamente Leido
Número de Filas:                                    8075
Número de Puntos de Extracción en Modelo de Bloques: 486
Ingreso área del punto de extracción en [m2]:        350
Ingreso Precio del Cobre en [US$/lb]:                2.8
Ingreso Descuento del Cobre a concentrado [US$/lb]:  0.4
Recuperación Metalúrgica Concentradora [%]:          88
Costo Mina + Costo Planta [US$/t]:                   20
Costo Preparación de área [US$/m2]:                  2950
-----
Ley de Corte Critica[%]:                             0.429
-----
-----Valorización Nominal Finalizada con Exito-----
-----Código para calcular VAN Máximo Columna-----
-----
Ingreso tasa de descuento en % :                     8
Ingreso altura de UBC en [m]:                        20
Ingreso máxima altura de columna en [m]:             400
-----Archivo Correctamente Leido-----
Número de Filas:                                    486
Cota de UCL para dibujar en OpenScad:                 2265
BUILD SUCCESSFUL (total time: 47 seconds)
  
```

Ilustración 4-10 Ingreso de parámetros técnicos y económicos

- ❖ Si el código y el modelo de bloque son correctos, el programa comenzará a solicitar los parámetros para la evaluación técnica económica

4.5. Salidas de NetBeans

- ❖ En la carpeta raíz, se encontrarán los siguientes archivos

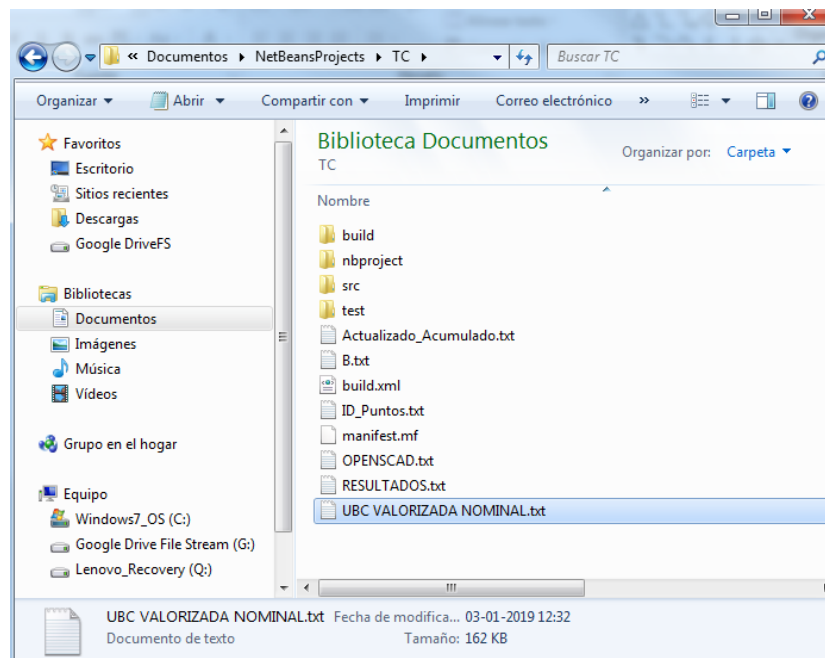


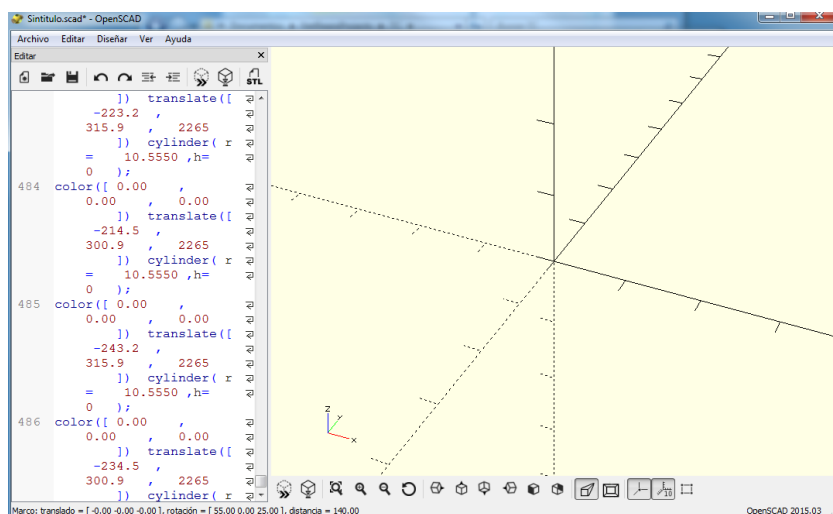
Ilustración 4-11 Archivos de Salida del Programa

ID_PUNTOS.txt	: Nombre y coordenadas en planta de cada punto de extracción
UBC VALORIZADA NOMINAL.txt	: Nombre PE, Coordenadas y Valor nominal por UBC en altura
OPENSCAD.txt	: Parámetros de entrada a OpenScad
RESULTADOS.txt	: Nombre PE, Coordenadas, VAN Máximo por PE y cota asociada

Cada uno de estos archivos permite al usuario trabajar información de detalle en Excel para efectos de análisis estadísticos más profundos. Donde el archivo resultados permite obtener el nombre de punto de extracción, las coordenadas, altura que maximiza el VAN y por ende el VAN Máximo factible de obtener.

4.6. Visualización en OpenScad

Con el archivo de salida OPENSCAD.txt es posible copiar el contenido de este (block de notas) y copiarlo directamente en la venta de interfaz de Openscad donde al apretar ENTER inmediatamente comienza a graficar a escala la altura de columna asociada a cada punto de extracción, y el color representar el rango de VAN [MUS\$]



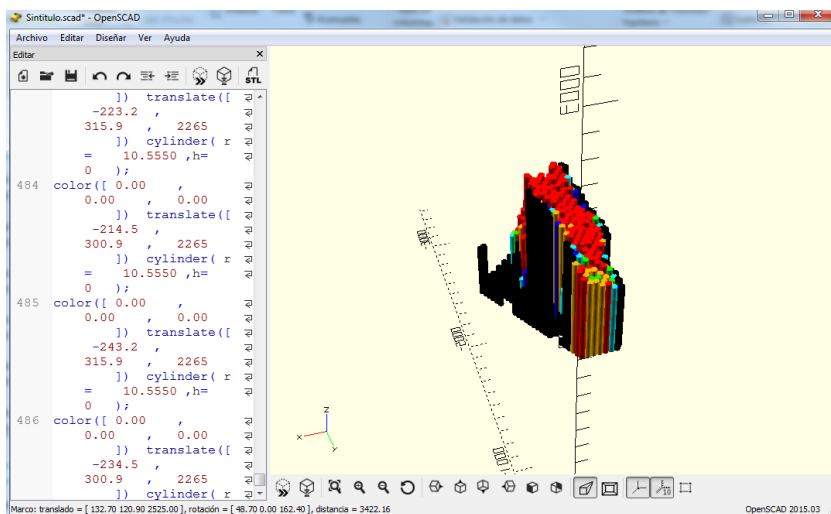


Ilustración 4-12 Visualización en 3D alturas de Columnas.

5. Bibliografía y recursos

- I. NetBeans (Plataforma para programar en lenguaje Java) <https://netbeans.org/downloads/8.2/>
- II. Software Openscad <http://www.openscad.org/downloads.html>
- III. ARANEDA, Octavio and GAETE, Sergio. Continuous modeling for caving exploitation. En: MassMin 2004 (2004, Santiago, Chile). in proceedings. Santiago, Chile, MassMin 2004, 2004. pp. 505 – 507.
- IV. CONTRERAS, Jorge. Metodología para determinar el valor en riesgo de un proyecto minero, mediante la aplicación de un modelo matemático de optimización. En: Universidad del Desarrollo (2016, Santiago, Chile). Tesis de grado de magíster en ingeniería industrial y de sistemas.
- V. CONTRERAS, Jorge, CORNEJO Javier and CAVIEDES, Christian, 2016. “Metodología para estimación de la tasa de incorporación de área en panel caving, Codelco – División El Teniente”, en U-Mining 2016: Primer congreso internacional en minería subterránea, ed’s: R Castro and M Valencia, Universidad de Chile, Santiago, Chile, pp. 452-460.

ANEXOS

Código del algoritmo

Captura de información y valorización nominal

Instrucciones para llamar a las bibliotecas de JAVA para ser usados en el programa

```
package tc;
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.PrintWriter;
import java.util.StringTokenizer;
```

Instrucciones para activar la interfaz y leer desde el teclado el nombre del modelo de bloques

```
public class TC {
public static void main(String[] args) throws IOException{
System.out.print("Ingrese Nombre de Modelo de Bloques: ");
BufferedReader in = new BufferedReader(new InputStreamReader(System.in));
String Modelo_Bloques = in.readLine();
BufferedReader a = new BufferedReader(new FileReader(Modelo_Bloques));
int contador=0;
while(a.readLine()!=null){
contador=contador+1;
}
System.out.println("Modelo de Bloques Correctamente Leido ");
System.out.println("Número de Filas: "+contador);
```

Separar las filas del modelo de bloques

```
BufferedReader b = new BufferedReader(new FileReader(Modelo_Bloques));
String [] Filas_MB = new String [contador];
for(int a1=0;a1<contador;a1++){
Filas_MB [a1] = b.readLine();
}
```

Instrucciones para separar en columnas

```
String[][]fila_columna_MB = new String[contador][6];
for(int a2=0;a2<contador;a2++){
StringTokenizer c = new StringTokenizer(Filas_MB[a2],",");
```

```

for(int a3=0;a3<6;a3++){
fila_columna_MB[a2][a3]=c.nextToken();
}
}

```

Contador y asignación de puntos de extracción

```

int contador_puntos=0;
for(int a4=0;a4<contador;a4++){
if(fila_columna_MB[a4][3].equals("1")==true){
contador_puntos=contador_puntos+1;
}
else
contador_puntos=contador_puntos+0;
}
System.out.println("Número de Puntos de Extracción en Modelo de Bloques: "+contador_puntos);

```

Creación del archivo ID_PUNTOS.txt

```

PrintWriter archivo = new PrintWriter(new FileWriter("ID_Puntos.txt"));
archivo.println("ID_Puntos;X;Y");
int j=0;
String [] Codigo_Puntos = new String [contador_puntos];
String [] X = new String [contador_puntos];
String [] Y = new String [contador_puntos];
String [] Z = new String [contador_puntos];
System.out.print("Ingrese área del punto de extracción en [m2]: ");
for(int i=1;i<contador;i++){
if(fila_columna_MB[i-1][0].equals(fila_columna_MB[i][0])==false){
Codigo_Puntos [j] = fila_columna_MB [i][0];
X [j]=fila_columna_MB[i][1];
Y [j]=fila_columna_MB[i][2];
Z [j]= Area;
j++;}}
for(int k=0;k<contador_puntos;k++){
archivo.println(Codigo_Puntos [k]+";"+X[k]+";"+Y[k]+";"+Z[k]);}
archivo.close();

```

Valorización Nominal

```

double [] Valor_UBC = new double [contador];
Valor_UBC [0]=0;
System.out.print("Ingrese Precio del Cobre en [US$/lb]: ");
String A1 = in.readLine();
double precio=Double.parseDouble(A1);
System.out.print("Ingrese Descuento del Cobre a concentrado [US$/lb]: ");

```

```

String A2 = in.readLine();
double descuento=Double.parseDouble(A2);
System.out.print("Recuperación Metalúrgica Concentradora [%]:      ");
String A3 = in.readLine();
double recuperación=Double.parseDouble(A3)/100;
System.out.print("Costo Mina + Costo Planta [US$/t]:                ");
String A4 = in.readLine();
double costos=Double.parseDouble(A4);
System.out.print("Costo Preparación de área [US$/m2]:            ");
String A5 = in.readLine();
double inversion =Double.parseDouble(A5);
System.out.println("-----");
Double ley = (costos/((precio-descuento)*recuperación*2204.62))*100;
int ley1 =(int)((int)1000*ley);
Double ley2 = (double)ley1/1000;
System.out.println("-----");
System.out.println("Ley de Corte Critica[%]:                "+ ley2);
System.out.println("-----");
PrintWriter archivo1 = new PrintWriter(new FileWriter("UBC VALORIZADA NOMINAL.txt"));
for(int z=1;z<contador;z++){
Valor_UBC[z]=
Double.parseDouble(fila_columna_MB[z][4])*(Double.parseDouble(fila_columna_MB[z][5])*(precio-
descuento)*2204.62*recuperación/100-costos);}
String [] A = new String [contador_puntos];
for(int tttt=0;tttt<contador_puntos;tttt++){
A[tttt]="";}
for(int y=0;y<contador_puntos;y++){
String Busqueda =Codigo_Puntos[y];
for(int w=1;w<contador;w++){
if(Busqueda.equals(fila_columna_MB[w][0])==true){
A[y]= A[y].concat(String.valueOf(Valor_UBC[w])).concat(";");}
archivo1.println(Codigo_Puntos[y]+";"+X[y]+";"+Y[y]+";"+Z[y]+";"+A[y]);}
archivo1.close();
System.out.println("-----Valorización Nominal Finalizada con Exito-----");

```

Máx. VAN y Altura Columna

Parámetros de entrada

```

System.out.println("-----Código para calcular VAN Máximo Columna-----");
System.out.println("-----");
System.out.print("Ingrese tasa de descuento en % :                ");
String A3XXXX = in.readLine();
double taxax=Double.parseDouble(A3XXXX)/100;
System.out.print("Ingrese altura de UBC en [m]:                ");
String Altura = in.readLine();
double Altura1=Double.parseDouble(Altura);
System.out.print("Ingrese máxima altura de columna en [m]:            ");

```

```
String Altura_max = in.readLine();
double Altura1_max=Double.parseDouble(Altura_max);
int limite = (int) ((int) Altura1_max/Altura1);
```

Contar número de Filas

```
BufferedReader abc = new BufferedReader(new FileReader("UBC VALORIZADA NOMINAL.txt"));
int contador_1=0;
while(abc.readLine()!=null){
contador_1=contador_1+1;}
System.out.println("-----Archivo Correctamente Leido-----");
System.out.println("Número de Filas:                "+contador_1);
```

Separar filas

```
BufferedReader bcd = new BufferedReader(new FileReader("UBC VALORIZADA NOMINAL.txt"));
String [] Filas = new String [contador_1];
for(int aaa1=0;aaa1<contador_1;aaa1++){
Filas [aaa1] = bcd.readLine();
}
```

Contar número de caracteres por fila

```
//CONTAR NÚMERO DE CARACTERES POR FILA
int caracteres_fila[]=new int[contador_1];
for(int ss=0;ss<contador_1;ss++){
caracteres_fila[ss]=Filas[ss].length();
}
```

Contar caracteres por fila (Para efectos de buscar el Separador)

```
int contador_columnas_fila []= new int[contador_1];
for(int bss1=0;bss1<contador_1;bss1++){
for(int bss2=0;bss2<caracteres_fila[bss1];bss2++){
if(";" .equals(Filas[bss1].substring(bss2,bss2+1))){
contador_columnas_fila[bss1]=contador_columnas_fila[bss1]+1;
}
}
}
```

Matriz de datos, utilizando los caracteres por fila, para efectos de lectura

```
//MATRIZ DE DATOS, UTILIZANDO LOS CARACTERES POR FILA, PARA EFECTOS DE LA LECTURA
String[][]fila_columna = new String[contador_1][100];
```

```

for(int ass2=0;ass2<contador_1;ass2++){
StringTokenizer css = new StringTokenizer(Filas[ass2],",");
for(int ass3=0;ass3<contador_columnas_fila[ass2];ass3++){
fila_columna[ass2][ass3]=css.nextToken();
}
}

```

Valor actualizado por UBC asumiendo un periodo fijo de extracción por bloque

```

int pos[]=new int[100];
pos[0]=1;pos[1]=1;pos[2]=2;pos[3]=2;pos[4]=2;pos[5]=3;pos[6]=3;pos[7]=3;pos[8]=3;pos[9]=4;pos[10]=4;pos[11]=4;
pos[12]=4;pos[13]=5;pos[14]=5;pos[15]=5;pos[16]=5;pos[17]=6;pos[18]=6;pos[19]=6;pos[20]=6;pos[21]=7;pos[22]=7;pos[23]=7;pos[24]=7;
pos[25]=8;pos[26]=8;pos[27]=8;pos[28]=8;pos[29]=9;pos[30]=9;pos[31]=9;pos[32]=9;pos[33]=10;pos[34]=10;pos[35]=10;pos[36]=10;
for(int i=37;i<100;i++){
pos[i]=11;
}
double valor_Act[][]=new double[contador_1][100];
for(int iiss1=0;iiss1<contador_1;iiss1++){
for(int jjss1=4;jjss1<contador_columnas_fila[iiss1];jjss1++){
valor_Act[iiss1][jjss1-4]=Double.parseDouble(fila_columna[iiss1][jjss1])/Math.pow(1+tasax,pos[jjss1-4]);
}
}

```

Inicialización de Variables

```

String[] concatenar=new String[contador_1];
for(int tss=0;tss<contador_1;tss++){
concatenar [tss]="";
}

```

VAN Acumulado por punto de extracción

```

double vss1[][]= new double[contador_1][100];
for(int ix=0;ix<contador_1;ix++){
for(int jax=1;jax<contador_columnas_fila[ix]+1;jax++){
vss1[ix][0]=-inversion*Double.parseDouble(fila_columna[ix][3])+valor_Act[ix][jax-1];
vss1[ix][jax]=vss1[ix][jax-1]+valor_Act[ix][jax];
}
}

```

Búsqueda de índice que maximiza VAN en columna

```

double maxt[]=new double[contador_1];

```

```

for(int alpha=0;alpha<contador_1;alpha++){
maxt[alpha]=-99999999;}
int indice[]=new int[contador_1];
for(int ixxx=0;ixxx<contador_1;ixxx++){
for(int jaxx=0;jaxx<Math.min(contador_columnas_fila[ixxx],limite);jaxx++){
maxt[ixxx]=Math.max(maxt[ixxx],Math.max(vss1[ixxx][jaxx],vss1[ixxx][jaxx+1]));
if(String.valueOf(maxt[ixxx]).equals(String.valueOf(vss1[ixxx][jaxx]))){
indice[ixxx]=jaxx;
}
}
}
PrintWriter archivo2 = new PrintWriter(new FileWriter("Actualizado_Acumulado.txt"));
for(int ia1xx=0;ia1xx<contador_1;ia1xx++){
for(int ja1xx=0;ja1xx<contador_columnas_fila[ia1xx]-4;ja1xx++){
concatenar[ia1xx]= concatenar[ia1xx].concat(String.valueOf(vss1[ia1xx][ja1xx])).concat(";");
}
archivo2.println(fila_columna[ia1xx][0]+";"+fila_columna[ia1xx][1]+";"+fila_columna[ia1xx][2]+";"+fila_columna[ia1xx][3]+";"+concatenar[ia1xx]);}
archivo2.close();
System.out.print("Cota de UCL para dibujar en OpenScad:          ");
String OpenScad = in.readLine();
int cota=(int)Double.parseDouble(OpenScad);
PrintWriter archivo3 = new PrintWriter(new FileWriter("RESULTADOS.txt"));
archivo3.println("ID_PE;X;Y;ÁREA;Altura_UBC_Máx_VAN;MAX_VAN[US$]");
for(int ixz=0;ixz<contador_1;ixz++){
archivo3.println(fila_columna[ixz][0]+";"+fila_columna[ixz][1]+";"+fila_columna[ixz][2]+";"+fila_columna[ixz][3]+";"+indice[ixz]*Altura1+";"+maxt[ixz]);
}
archivo3.close();
PrintWriter archivo4 = new PrintWriter(new FileWriter("OPENSCAD.txt"));
for(int ixz=0;ixz<contador_1;ixz++){
archivo4.println(fila_columna[ixz][1]+";"+fila_columna[ixz][2]+";"+fila_columna[ixz][3]+";"+cota+";"+indice[ixz]*Altura1+";"+maxt[ixz]);
}
archivo4.close();
}
}
}

```