

UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE INGENIERÍA METALÚRGICA Y DE MATERIALES
SANTIAGO - CHILE



SELECCIONADOR DE PISO ÓPTIMO ECONÓMICO PARA EXPLOTACIÓN POR HUNDIMIENTO DE BLOQUES DESARROLLADO BAJO PYTHON

VALENTINA PAZ NÚÑEZ NÚÑEZ

**MEMORIA DE TITULACIÓN PARA OPTAR AL TÍTULO DE INGENIERA
CIVIL DE MINAS**

PROFESOR GUÍA
VICTOR ENCINA MONTENEGRO
PROFESOR CO-REFERENTE
ROLANDO CARRASCO

OCTUBRE 2022

RESUMEN

La minería subterránea consiste en la extracción de minerales de interés económico a profundidades donde la aplicación de minería a cielo abierto ya no es rentable. Para ello existen diferentes métodos, que se aplican dependiendo de las características geológicas y de disposición física del yacimiento. Uno de los métodos de mayor interés corresponde al de hundimiento por bloques debido a sus altas tasas de producción y bajos costos.

Actualmente, el mercado de los *softwares* mineros para el diseño de explotación minera es bastante diverso, pero son pocos los que poseen herramientas específicas para diseño y planificación con el método de hundimiento por bloques y particularmente, para encontrar el piso óptimo de socavación basal.

En el caso de los *softwares* de diseño minero que incluyen, entre sus herramientas, algún paquete especializado para definir el piso óptimo de socavación, se trata de productos opcionales, cuyos costos de adquisición son elevados que limitan las posibilidades de usarlo.

Como una forma de evitar esa limitación económica, el presente trabajo propone un programa desarrollado en lenguaje de programación *Python*, que facilita y extiende el alcance de uso de un procedimiento seleccionador de piso óptimo, desarrollado anteriormente en la USM, en base a planillas de cálculo.

El seleccionador de piso para explotaciones de hundimiento por bloques o paneles considera interfases desplegables, de aspecto típico de la programación por objetos, muy fáciles de operar para cargar los modelos de recursos y los parámetros necesarios para la selección del piso de socavación. También entrega distintas opciones gráficas en 2 o 3 dimensiones que permiten al usuario apreciar resultados parciales y tomar decisiones durante el proceso de selección.

Para validarlo, el programa seleccionador se aplicó a dos modelos de bloques de 22.275 y 474.848 bloques respectivamente. El modelo pequeño permitió verificar que el resultado obtenido es el mismo que se obtendría con el método original con planillas de cálculo y la aplicación al modelo grande, demostró que el código trabaja bien sin limitaciones debidas al tamaño del modelo de recursos.

ABSTRACT

Underground mining consists on the extraction of minerals of economic interest to depths when open pit mining is no longer profitable. For these cases there are different methods that are applied depending on geological features and physical disposition of the pit. One of the most interesting methods is related to those of subsidence due to their high production rates and low costs.

Currently, the mining software market is quite diverse, however, there are very few software that have the necessary tools to optimally work a block caving method, particularly on how to find an optimal soil suitable for the reservoir.

If we talk about mine design software that has a specialized set up of tools to define optimal soil, it comes as an optional tool, which high costs mean that only one group can obtain its benefits.

To avoid this economical limitation, the main objective of this work is the creation of an optimal soil selector through the *Python* programming language, that eases and extends the use of a procedure to select optimal soil, previously developed in USM, based on spreadsheets.

The software to select optimal soil for subsidence methods considers spreadable interfaces, typical of object programming, easy to operate when is needed to import resource models and necessary parameters to select optimal soil. Moreover, it gives different graphic options in 2 & 3 dimensions that allow the user to appreciate partial results and to take decisions throughout the process of selection.

To validate the created optimizer software, it was applied in two different block models. The first model has 22,275 blocks, while the second contains 474,848 blocks. The smaller model allowed to verify that the obtained result was the same as what would be obtained through the original process using spreadsheets, while the bigger model demonstrated that the code worked fine and without limitations regarding resource model size.

DEDICATORIA

Quiero dedicar esto a aquellas personas que siempre estuvieron para mí a pesar de todo.

Se lo dedico a mi familia

A ti Nory por apoyarme y esperarme siempre con comida calientita en las noches

A ti Nono por irme a dejar al paradero las mañanas e irme a buscar en las noches

Y también a mi mascota que ya no está conmigo, pero si en mi corazón, y quien me acompañó muchas noches de estudios

A ti Rita

“Cada cicatriz que tenemos es la confirmación de que las heridas sanan”

AGRADECIMIENTOS

Nunca es fácil terminar una etapa así, sin embargo, me siento satisfecha a como finalizó, y sé que esta satisfacción no sería la misma sin las personas que estuvieron presentes en el periodo en el que me transformaba en la primera universitaria de la familia. Una Ingeniera Civil de Minas.

En un comienzo quiero agradecer a mi mamá, porque siempre ella me motivo a estudiar en una universidad, también me dio su apoyo durante esta etapa

Quiero agradecer a mis abuelos que siempre mostraron su apoyo incondicional y por siempre intentar dar todo lo que tenían a su alcance para facilitarme las cosas.

Quiero agradecer a mi tío Carlos por apoyarme, por irme a buscar a la universidad solo para que alcanzara a jugar un partido. Por regalarme mis primeras zapatillas de futbol para jugar en mi etapa universitaria.

Quiero agradecer a mis amigos de universidad, especialmente a Karla y a mis chicos BNA: a Gabriel, Bryam, Cristopher, Oscar, Guillermo, Alán, Markos, Leufu. Realmente la universidad no hubiese sido lo mismo sin todos ustedes, las tardes de risa o las tardes de estudio. Nos integraron en su grupo y nos hicieron parte de ustedes y de verdad que espero que esta relación siga a través de los años. Los quiero mucho chiquillos, a todos.

Quiero agradecer a mis amigos Axel y Daniel, por esas tardes de peloteo, realmente me desestresaban, además de que las risas nunca faltaron y las motivaciones que nos dábamos, además de siempre decirnos que podíamos con todo.

Quiero agradecerle a mi amigo Martín, mi vecino, que me apoyo mucho sobre todo este último proceso.

Por último, quiero agradecer a mi profesor guía Víctor Encina, por su apoyo incondicional, y ofrecerme este tema de memoria que realmente me gusto y me dejo conforme con el resultado. Muchas gracias profe por su ayuda y guía en este proceso. No pude haber elegido un mejor profesor para finalizar esta etapa.

ÍNDICE DE CONTENIDO

I.	INTRODUCCIÓN	8
1.	Antecedentes	8
2.	Objetivo	9
2.1.	Objetivo general	9
2.2.	Objetivos específicos.....	9
II.	MARCO TEÓRICO.....	10
3.	Software Minero.....	10
3.1.	Optimización de piso económico	11
4.	Manual	12
III.	DESARROLLO DEL TEMA.....	13
5.	Metodología	13
5.1.	Definir diagrama de flujo.....	14
5.2.	Definir variables de entrada al proyecto.....	16
5.3.	Obtención de la cota optimizada por columna.....	18
5.4.	Obtención de valorización por piso.....	20
5.5.	Otras consideraciones	21
5.6.	Obtención de gráficas.....	21
5.7.	Creación de interfaz gráfica.....	23
6.	Resultados.....	23
6.1.	Ingreso de parámetros iniciales.....	23
6.2.	Interfaz gráfica	24
6.3.	Modelo 1	25
6.4.	Modelo 2.....	32
7.	Análisis	39
7.1.	Modelo 1	39
7.2.	Modelo 2.....	45
IV.	CONCLUSIONES	53
V.	REFERENCIAS	54
VI.	ANEXOS.....	55

ÍNDICE DE FIGURAS

Figura 1: Esquema de pasos a seguir para determinar el piso óptimo en Excel.	12
Figura 2: Esquema de la metodología a implementar para obtener el optimizador de piso económico.	14
Figura 3: Diagrama de flujo del proceso de programación.	15
Figura 4: Esquema representativo de un bloque perteneciente a un modelo.	16
Figura 5: Variables de entrada al optimizador.	17
Figura 6: Esquema de una columna de bloques.	18
Figura 7: Esquema de una columna de bloques con diferenciación de valores de bloques.	19
Figura 8: Ventana emergente creada para el ingreso del costo de preparación por columna.	24
Figura 9: Interfaz gráfica creada del optimizador.	25
Figura 10: Visualización tridimensional de los bloques positivos del Modelo 1.	26
Figura 11: Visualización tridimensional de los pisos óptimos por columnas del Modelo 1.	27
Figura 12: Vista planta de los óptimos por columna del Modelo 1.	28
Figura 13: Gráfica beneficio vs piso del Modelo 1.	29
Figura 14: Gráfica de beneficio vs piso del Modelo 1.	30
Figura 15: Gráfica de beneficio vs piso del Modelo 1.	31
Figura 16: Visualización de las columnas pertenecientes a cada nivel.	31
Figura 17: Visualización tridimensional de los bloques positivos del Modelo 2.	33
Figura 18: Visualización tridimensional de los pisos óptimos por columnas del Modelo 2.	34
Figura 19: Vista planta de los óptimos por columna del Modelo 2.	35
Figura 20: Gráfica de beneficio vs piso del Modelo 2.	36
Figura 21: Gráfica de beneficio vs piso del Modelo 2.	37
Figura 22: Gráfica de beneficio vs piso del Modelo 2.	38
Figura 23: Visualización de las columnas pertenecientes a cada nivel.	39
Figura 24: Extensión del Modelo 1.	40
Figura 25: Visualización tridimensional de los pisos óptimos por columna.	41
Figura 26: Piso óptimo del Modelo 1.	42
Figura 27: Curva de beneficio vs piso.	43
Figura 28: Curva de Beneficio vs piso.	43
Figura 29: Curva de beneficio vs piso.	44
Figura 30: Extensión del Modelo 2.	46
Figura 31: Visualización tridimensional de los pisos óptimos por columna.	47
Figura 32: Piso óptimo del Modelo 2.	48
Figura 33: Curva de beneficio vs piso.	49
Figura 34: Curva de beneficio vs piso.	50
Figura 35: Curva de beneficio vs piso.	50
Figura 36: Piso óptimo obtenido a través de Excel.	65

ÍNDICE DE TABLAS

Tabla 1: Variables de entrada al optimizador.	16
Tabla 2: Características del Modelo 1.	25
Tabla 3: Características del Modelo 2.	32
Tabla 4: Comparación de beneficios entre creación de un nivel y un multinivel.	45
Tabla 5: Comparación de beneficios entre creación de un nivel y un multinivel.	51
Tabla 6: Resultados obtenidos al calcular piso óptimo a través de Excel.	65
Tabla 7: Resultados obtenidos al calcular el piso óptimo a través de Excel.	66

I. INTRODUCCIÓN

1. Antecedentes

La minería subterránea corresponde a una técnica que se usa para la extracción de minerales desde depósitos que se encuentran a profundidades donde no es posible aplicar un método de minería a cielo abierto. Se estima que cerca del 20% de la minería mundial corresponde a este tipo de método siendo Chile el líder de esta producción. Dentro de este tipo de minería, existen tres tipos de método de explotación:

- Autosoportado
- Artificialmente soportados
- Transitoriamente soportados
- No soportados

Entre el último se encuentra el método de hundimiento por bloques el cual esta involucrado en el presente proyecto. Este método consiste en la generación de una excavación con dimensiones suficientes para producir el colapso del techo, ya sea de manera natural o artificial, generando un autorrelleno controlado de la cavidad con el material hundido. Esta técnica es la preferida en la minería subterránea de yacimientos de gran magnitud debido a sus bajos costos y altas tasas de productividad.

A consecuencia de esto último, son cada vez más los proyectos mineros que se atreven a ejecutar métodos de hundimiento, por lo que cada vez es más necesaria la implementación de herramientas que permitan el fácil acceso a los cálculos relacionados.

Uno de los pasos principales es la obtención del piso óptimo de la explotación, pues desde allí se comenzarán a construir los niveles necesarios para poder poner en marcha el proyecto, sin embargo, los pocos softwares especializados y los altos precios de estos motivaron la creación de un seleccionador de pisos óptimos que entregue información necesaria para determinar si un depósito posee un piso óptimo de un nivel o un multinivel.

2. Objetivo

2.1. Objetivo general

El objetivo principal del presente proyecto es el desarrollo seleccionador de piso óptimo simple y rápido para obtener un piso único o multinivel en modelos de bloque de cualquier tamaño.

2.2. Objetivos específicos

- Desarrollar un diagrama de flujo de los pasos a seguir en la elaboración del código en lenguaje *Python*.
- Obtener una valorización por piso y evidenciarla en una gráfica de beneficio vs piso.
- Comparación del beneficio entregado por el optimizador para el caso con un nivel y un multinivel.
- Desarrollar una interfaz gráfica que englobe todo lo obtenido anteriormente.

II. MARCO TEÓRICO

La minería por hundimiento de bloques corresponde a un método minero subterráneo que permite una extracción masiva dentro de un depósito generalmente de baja ley, este método requiere múltiples evaluaciones, una de las más tempranas que se lleva a cabo en la etapa conceptual corresponde a la determinación del piso económico del yacimiento, dicha estimación puede realizarse de manera manual si el modelo lo permite, o a través de softwares mineros con elevados costos. Uno de los softwares más utilizados para ello corresponde al *GEMS* de *GEOVIA* específicamente el módulo *PCBC*, y cuya función principal es la de generar incrementos económicos a las empresas mineras que lo utilizan y que poseen minas extraídas por estos métodos.

3. Software Minero

El software *GEOVIA PCBC*, desde ahora denominado *PCBC*, lleva dentro de la industria minera cerca de 25 años, siendo actualmente la herramienta más utilizada por ingenieros planificadores para modelar la minería por hundimiento de bloques.

El software define modelos numéricos que simulan las columnas de extracción sobre los puntos de extracción, además de mezclar algoritmos para modelar el comportamiento de las mezclas ocurridas en la extracción.

Dentro de las aplicaciones que posee el software *PCBC* se encuentran las siguientes:

- Modelamiento de la minería por hundimiento
- Optimización de piso óptimo
- Planificación de la producción
- Programas de producción optimizados
- Conciliación de reservas
- Observación geológica

Debido a la naturaleza de este proyecto, es que la aplicación del software que se verá a mayor profundidad es la de la optimización de piso óptimo.

3.1. Optimización de piso económico

Footprint Finder corresponde a una herramienta específica dentro del *software PCBC* que es utilizada para determinar el piso óptimo más económico del yacimiento y lleva siendo utilizada cerca de 10 años. Dentro de las variables de entrada que posee, se encuentran las siguientes:

- Modelo de bloques
- Parámetros económicos
- Restricciones (Opcionales)

Esta herramienta se usa dentro de las primeras etapas de evaluación de un yacimiento, es decir fase conceptual o de factibilidad, y sus recientes mejoras lograron evaluación mucho más detalladas en términos de restricciones.

Teniendo en cuenta la poca cantidad de minas de hundimiento por bloques que hay actualmente, hay un considerable aumento en el número de evaluaciones que se llevan a cabo por este método, esto ocurre debido a la exploración de cada vez más depósitos de gran escala a mayores profundidades.

El foco principal de esta herramienta es la de entregar valores económicos de cada columna con un rango de altura de columna específico.

El programa usa directamente el modelo de bloques importado para crear un acumulativo de valores para cada columna de bloques y posteriormente realiza el cálculo para cada cota del depósito, entregando finalmente el valor económico por piso, el área involucrada y su tonelaje, sin embargo, la gran limitación que posee es que trabaja con la optimización de un solo nivel.

4. Manual

Otra forma de poder obtener el polígono correspondiente al piso óptimo del recurso de interés es mediante el trabajo del modelo a través de plataformas como *Excel*, sin embargo, dichas plataformas son muy limitantes en relación a la cantidad de datos con los que es posible trabajar, además de lo lento que puede llegar a ser.

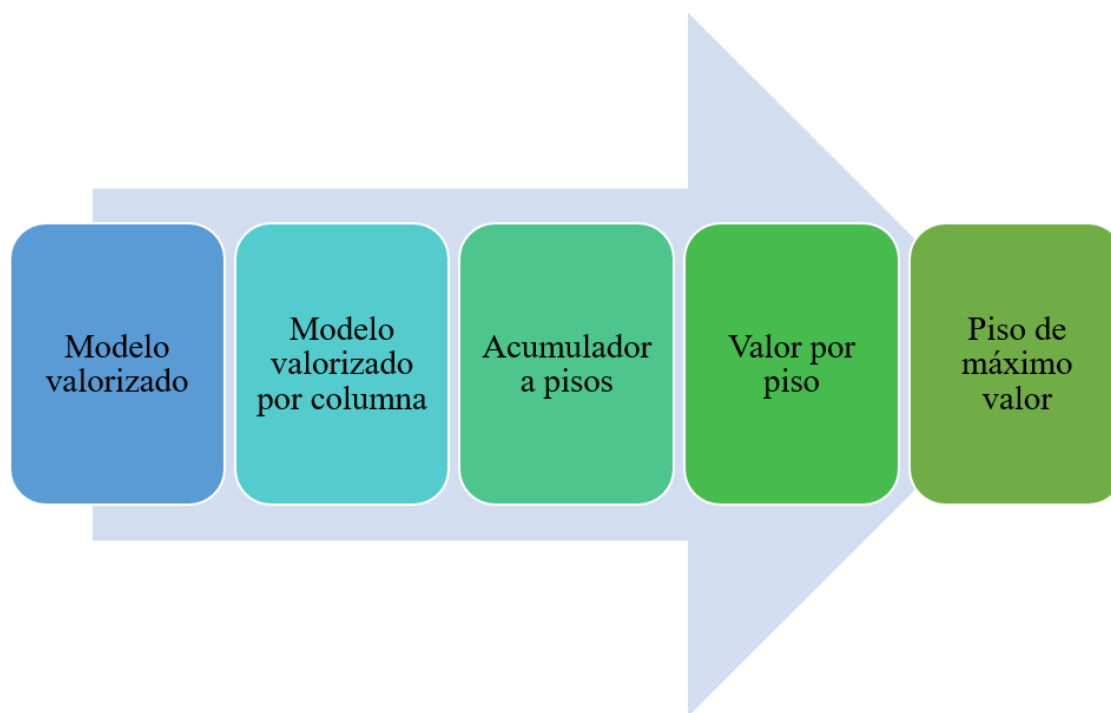


Figura 1: Esquema de pasos a seguir para determinar el piso óptimo en Excel.

El esquema anterior muestra los pasos principales que deben seguirse para determinar el piso óptimo de un yacimiento a través de una plataforma como *Excel*.

Cada columna perteneciente al modelo se encuentra conformada por bloques verticales, donde se destaca la existencia de un piso y de una altura óptima que maximiza el valor de esta. Al realizar las acumulaciones correspondientes de los valores, se obtiene un valor por piso, consiguiéndose un gráfico que muestra el beneficio total vs el piso del cual se puede obtener un óptimo que significa que, al realizar un único nivel, en este piso ocurre la maximización del beneficio.

III. DESARROLLO DEL TEMA

Con la finalidad de satisfacer tanto el objetivo principal como los objetivos específicos de este trabajo, es que se hace necesaria la creación de una metodología que indique los pasos a seguir para lograr la creación de un optimizador y así determinar el piso óptimo económico de un yacimiento a través del lenguaje de programación *Python*.

5. Metodología

La metodología creada para implementar el optimizador se clasifico principalmente en 3 etapas. La primera etapa consiste en el desarrollo de un diagrama de flujo simple para enfocar los pasos principales a seguir, sin embargo, se hace necesario el desglose de este diagrama de flujo para facilitar el código a crear, así como la definición de las variables de entrada que se utilizaran para poder trabajar el código en el lenguaje de *Python*,

La segunda etapa de la metodología consiste en los hitos que se deben alcanzar desde el optimizador para asegurar el cumplimiento de los objetivos específicos. Esta etapa está enfocada a obtener resultados simples, pero ya desde el optimizador.

Finalmente, la tercera etapa consiste en la creación de una interfaz gráfica que tiene como finalidad aglomerar los resultados obtenidos anteriormente y transformarlo en una ventana visualmente atractiva y sencilla de utilizar.

A continuación, se puede observar un esquema de la metodología propuesta para atrapar los objetivos planteados inicialmente:

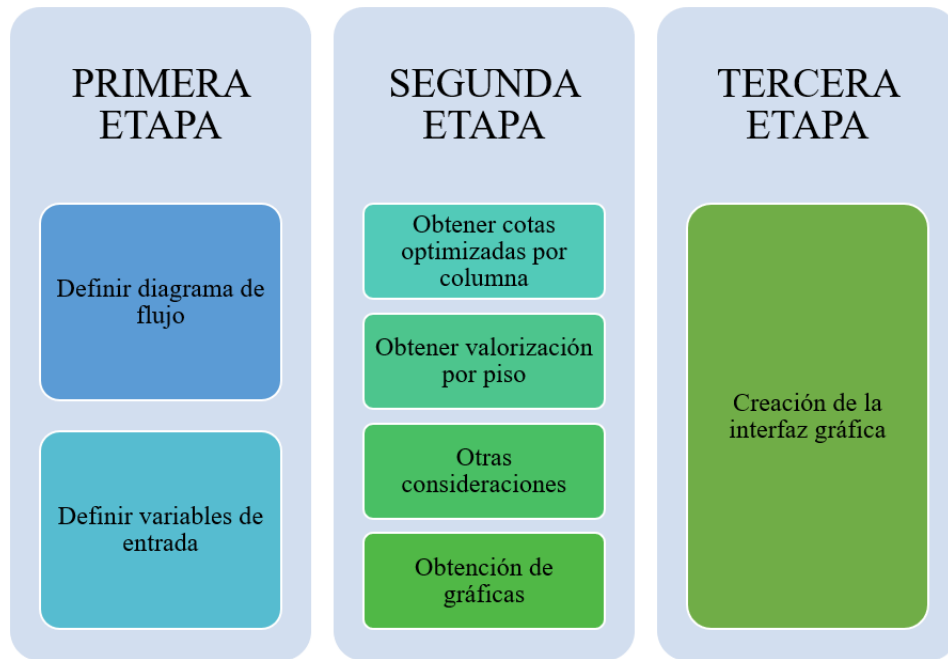


Figura 2: Esquema de la metodología a implementar para obtener el optimizador de piso económico.

El esquema anterior será detallado en los siguientes puntos:

5.1. Definir diagrama de flujo

La definición del diagrama de flujo es fundamental para trabajar con un lenguaje de programación, debido a que permitirá un mayor desglosamiento de los pasos a seguir para facilitar el código a crear.

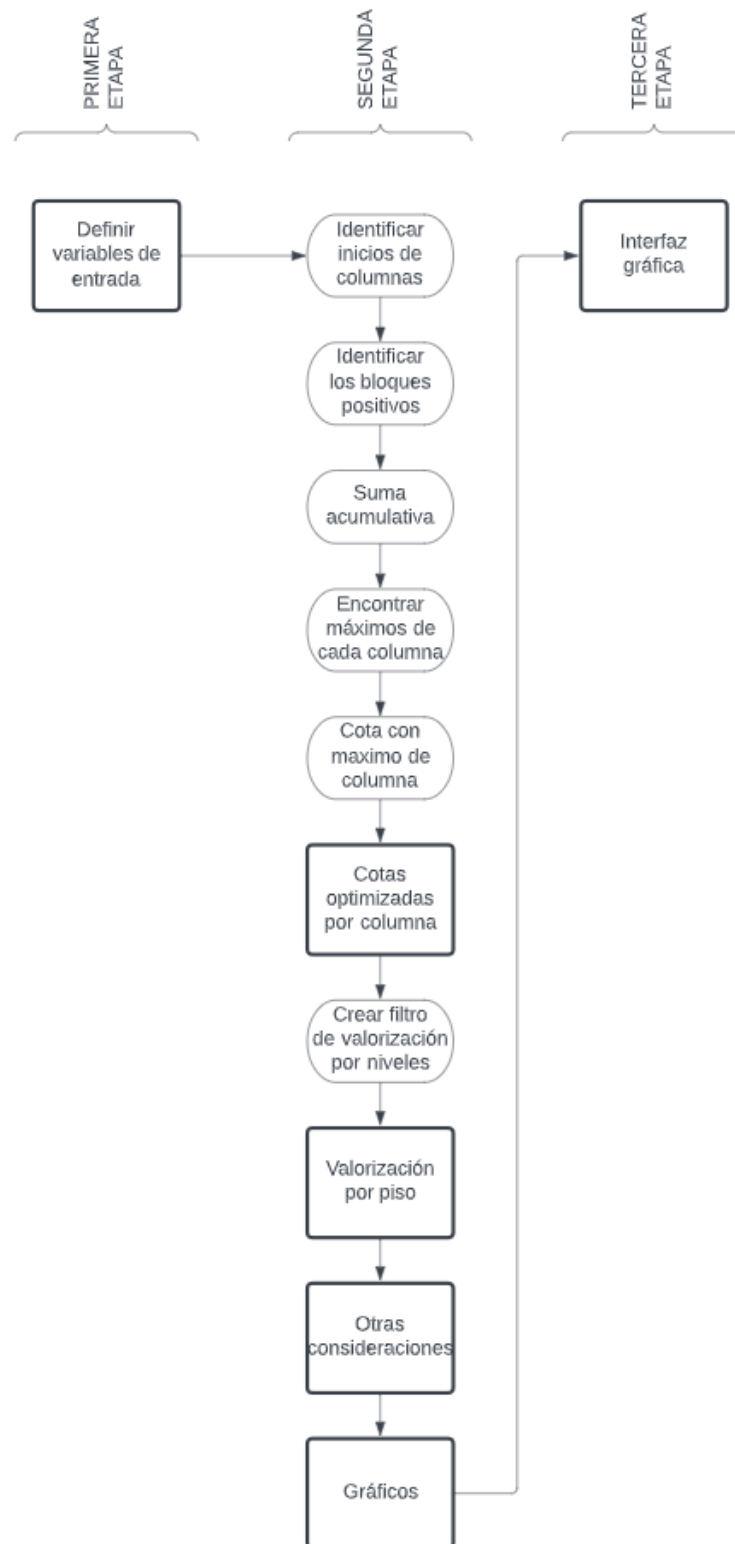


Figura 3: Diagrama de flujo del proceso de programación.

5.2. Definir variables de entrada al proyecto

La definición de estas variables tiene que ver con la necesidad de precisar aquellas que servirán como entrada para poder concretar el objetivo final del trabajo. Debido a que la finalidad es encontrar el piso óptimo para cada modelo geológico analizado, es que es importante que se considere las variables espaciales como entrada pues cada modelo estará asociado a un distinto espacio geográfico.

Las principales variables de entrada que debe contener la base de datos corresponden a Norte, Este y cota.

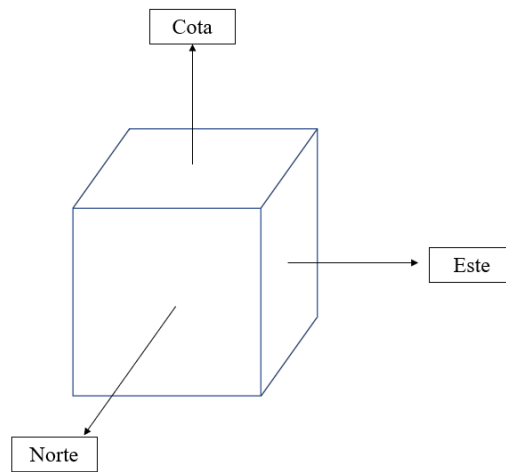


Figura 4: Esquema representativo de un bloque perteneciente a un modelo.

También se considera el ingreso de una cuarta variable de entrada, la cual debe indicar el valor del bloque valorizado, esto quiere decir, que se debe realizar una preparación de la base de datos para poder ingresarla al optimizador.

Tabla 1: Variables de entrada al optimizador.

Variables de espacio			Variable de bloque valorizado
Este	Norte	Cota	VB

El orden en que deben encontrarse las variables de espacio es la siguiente:

- Este: Ascendente

- Norte: Ascendente
- Cota: Descendente

	A	B	C	D	E
1	Este	Norte	Cota	VB	
2	-130	380	1650	-77	
3	-130	380	1630	-77	
4	-130	380	1610	-77	

Figura 5: Variables de entrada al optimizador.

Las variables antes mencionadas deben ser ingresadas a partir de la base de datos, la cual debe poseer el nombre de “Modelo” y su extensión debe ser “.csv”, además de poseer el delimitador “;”.

Por último, se debe de ingresar una quinta variable al optimizador, la cual indicará el valor del Costo de la preparación de la columna, cuyo valor dependerá del costo de preparación en dólares, y del área de la columna en metros.

Dado que el ingreso de este valor dependerá del modelo de bloques a utilizar, es que se llevará a cabo la implementación de una ventana gráfica para permitir la entrada de este parámetro.

La creación del código de la ventana gráfica emergente que permitirá el ingreso del costo de preparación por columna se puede visualizar en el Anexo 1.

El parámetro a ingresar en la ventana antes mencionada se configuró en las unidades de US\$/Columna.

Teniendo definidas cuáles serán las variables de entrada del proyecto, es que se hace necesario entrar a la siguiente fase de la metodología.

5.3. Obtención de la cota optimizada por columna

La segunda etapa de la metodología se dedica a la implementación del código en el lenguaje *Python* para a través desde el cual se obtendrá la cota optimizada por columna, para ello se aplicarán diferentes pasos, los cuales se detallarán a continuación:

Paso 1: Como primera etapa, es importante crear un identificador de cada columna existente dentro de la base de datos. Para ello es que se creará una variable denominada “Nueva Columna” cuya finalidad es asignarles a los bloques un valor entero que oscila entre 1 y 0. El valor 1 se le asignara al bloque que inicia la columna, y los bloques 0 a los que se encuentran bajo este de manera como se aprecia a continuación:

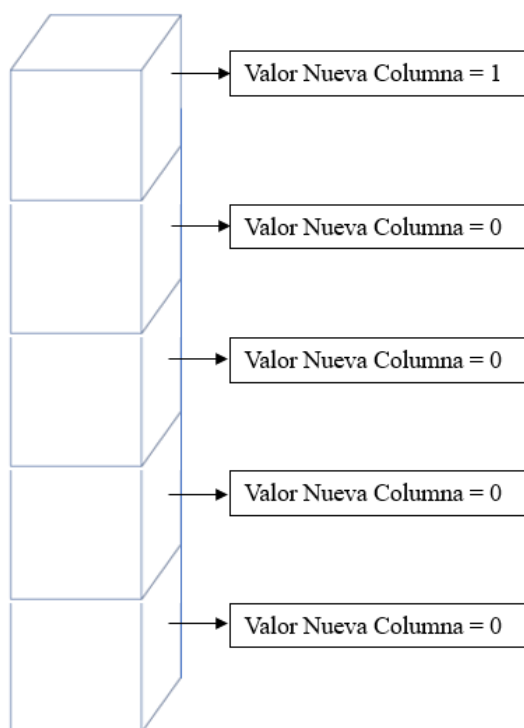


Figura 6: Esquema de una columna de bloques.

La creación del código para lograr asignar los valores descritos anteriormente puede visualizarse en Anexo 2.

Paso 2: El siguiente paso a realizar corresponde a la identificación la cota de los bloques que poseen un valor positivo, para ello se creara la variable “Cota Positiva”. La variable mencionada anteriormente será creada mediante una función que indicará la cota del bloque si el valor de este es positivo, en caso contrario, la variable creada indicará 0. Un ejemplo de lo anterior es lo que se puede evidenciar en la siguiente figura a modo de ejemplo:

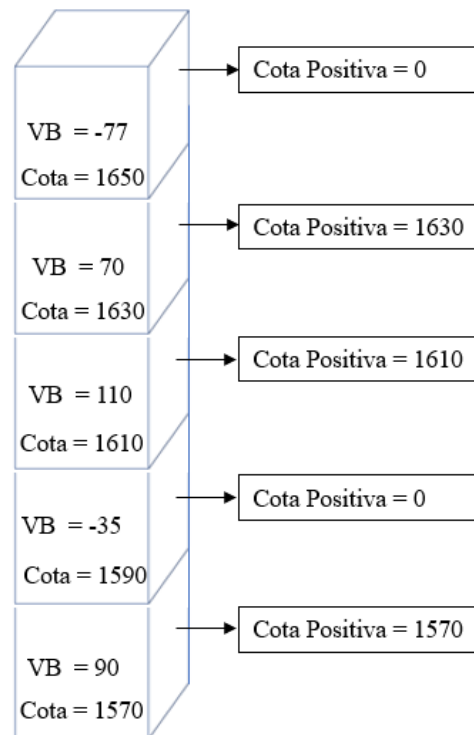


Figura 7: Esquema de una columna de bloques con diferenciación de valores de bloques.

De acuerdo con la indicado anteriormente y para lograr implementar la creación se implementará el código ubicado en Anexo 3.

Paso 3: La tercera etapa consiste en la creación de una nueva variable denominada “Suma Abajo”, cuyo objetivo es realizar una suma acumulativa por columna, en esta suma se deben incluir tanto bloques positivos como bloques negativos.

El código que implementa el paso descrito se encuentra en Anexo 4.

Paso 4: El cuarto paso implica la creación de una nueva variable llamada “Índice optimizador”, el cual, en base a una función aplicada, entrega el máximo valor acumulado de una columna, ingresando un 1 cuando este valor es encontrado, y un 0 cuando es otro valor.

La implementación de esta etapa se aplicó en dos pasos principales, primero se crea una lista para obtener los máximos de cada columna, mientras que el segundo paso es realizar la comparación entre la columna creada en el paso anterior y la lista.

En Anexo 5 se puede observar el código desarrollado para cumplir con este paso.

Paso 5: La quinta etapa consiste en la creación de una variable nombrada “Valor óptimo”, y entrega la diferencia entre el máximo valor acumulado y el costo de preparación por columna, éste último valor es ingresado manualmente por el usuario al ejecutar el optimizador a través de la ventana gráfica.

En Anexo 6 se puede encontrar el código implementado para conseguir completar el paso descrito.

Paso 6: Por último, se crea una nueva variable denominada “Piso óptimo” la cuál a través de una función entregará la cota donde se encuentra ubicado el valor máximo de cada columna.

En Anexo 7 se puede encontrar el código implementado para conseguir completar el paso descrito.

5.4. Obtención de valorización por piso

En el avance anterior, se logró obtener la cota óptima para cada columna, sin embargo, es necesario crear un filtro que permita entregar la valorización por piso para así determinar un piso óptimo para el yacimiento. Cabe destacar que, hasta este punto, la evaluación desarrollada considera el cálculo de solo un piso óptimo para todo el depósito.

Es posible encontrar el desarrollo del código de ambos pasos en Anexo 8.

5.5. Otras consideraciones

Con la finalidad de cumplir a cabalidad el objetivo planteado, es que se hace necesario realizar consideraciones dentro del código, pues como se menciona anteriormente, la optimización desarrollada, entrega un piso económico asociado a un nivel único, y dada la variedad de yacimientos que existen, esto puede no adaptarse de forma universal. Por ello, para solucionar este problema, se propone un método que será capaz de comparar el beneficio entregado por un único nivel frente al beneficio obtenido al desarrollar dos niveles.

Para que el seleccionador sea capaz de determinar la existencia de un multinivel, se debe realizar la evaluación del modelo, determinando el nivel óptimo de este, y localizando la cota donde se encuentra, y usándola como punto de quiebre. Este punto de quiebre determina un límite en el modelo, y se utilizará para obtener dos niveles óptimos. El primer nivel se obtiene con la evaluación de las columnas que posee un piso óptimo por sobre este límite, calculando nuevamente un óptimo para estas columnas, mientras que el segundo nivel realiza el mismo procedimiento con la diferencia de que la evaluación se realiza con las columnas que poseen un piso óptimo bajo el punto de quiebre.

5.6. Obtención de gráficas

El último paso de la segunda etapa consiste en la obtención de las gráficas que permitirán realizar la comparación de los beneficios obtenidos por cada piso, así como la visualización de los bloques positivos del modelo y una vista en tridimensional del piso óptimo de cada columna. A continuación, se presentarán los gráficos incorporados:

- a) Modelo de bloques con bloques con beneficio positivo

Este gráfico será representado a través de una vista tridimensional, en donde se podrá evidenciar el cuerpo de interés, pues solo se desplegarán en el visualizador aquellos bloques que posean un beneficio positivo.

b) Visualización tridimensional de los pisos óptimos por columnas

Esta gráfica será representada a través de una vista tridimensional, en la cual se podrá observar de manera tridimensional el piso óptimo para cada columna.

c) Piso óptimo

La gráfica de piso óptimo corresponderá a una gráfica en dos dimensiones, la cual mostrará la información de la cota a través de un gradiente de calor.

d) Beneficio en función del piso

Esta gráfica será representada en una vista en dos dimensiones, donde el eje de las abscisas estará indicado por el piso, mientras que el eje de las ordenadas estará representado por el beneficio.

e) Visualización de las columnas pertenecientes a cada nivel

Esta gráfica corresponde a una visualización de las columnas pertenecientes a los multiniveles.

El código implementado para obtener estas gráficas se encuentra en Anexo 9.

5.7. Creación de interfaz gráfica

La creación de una interfaz gráfica es necesaria para lograr reunir todo lo realizado anteriormente y permitir que el usuario sea el que seleccione que información quiere ver. A raíz de lo expresado anteriormente, la interfaz gráfica creada debe presentar botones que permitan al consumidor acceder rápidamente la información.

El código implementando para la creación de la interfaz gráfica se puede encontrar en el Anexo 11.

6. Resultados

Para comprobar la funcionalidad del código creado, es que se evidenciaran los resultados obtenidos en dos modelos de bloques de prueba distintos, uno con una poca cantidad de bloques y otro con una gran cantidad (Esto permite comprobar que el código funciona independientemente del tamaño del modelo).

6.1. Ingreso de parámetros iniciales

A continuación, se encuentra la ventana emergente creada para el ingreso del costo de preparación por columna. Como indica la figura, este debe ser ingresado en las respectivas unidades (US\$/COL) y luego debe ser presionado el botón “*Ingresar*” para que el optimizador reconozca el parámetro ingresado y continúe realizando el cálculo del piso óptimo.

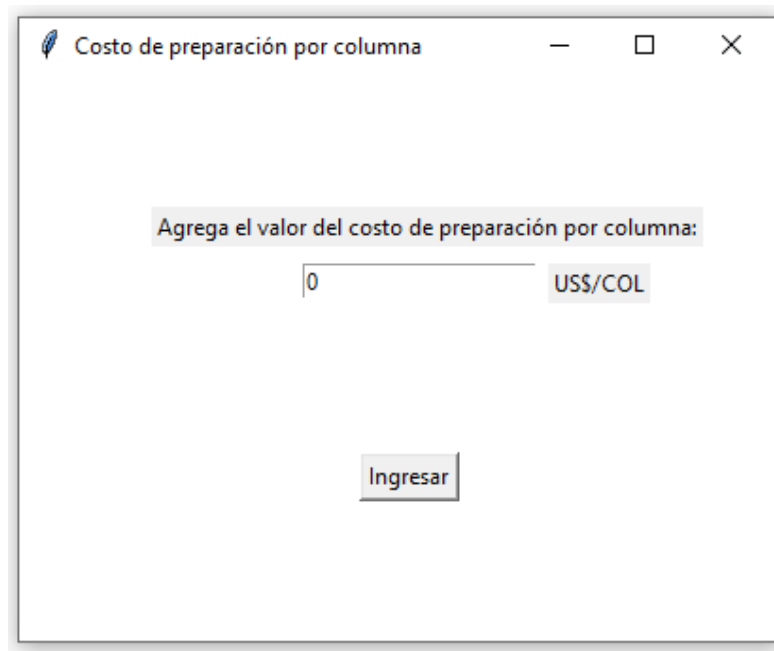


Figura 8: Ventana emergente creada para el ingreso del costo de preparación por columna.

6.2. Interfaz gráfica

A continuación, se puede observar la interfaz gráfica creada para organizar el optimizador de pisos óptimos. En el centro de la interfaz, se puede apreciar un visualizador, donde se podrán apreciar los gráficos de beneficio en función del piso obtenidos al oprimir los botones ubicados en la zona derecha de la ventana gráfica. Por otro lado, al presionar los botones localizados en el sector izquierdo de la ventana, se desplegará una ventana emergente con la información.



Figura 9: Interfaz gráfica creada del optimizador.

6.3. Modelo 1

Como se mencionó anteriormente, el optimizador de piso óptimo se verificará en dos modelos distintos para evaluar su funcionalidad. El primer modelo posee las siguientes características:

Tabla 2: Características del Modelo 1.

N° bloques 22.275		
Modelo	Mínimo	Máximo
Este	-130	510
Norte	380	900
Cota	1170	1650

6.3.1. Modelo

A continuación, se presenta una vista tridimensional de los bloques del modelo que poseen un beneficio positivo. Esta visualización se puede obtener al seleccionar el botón denominado “Modelo”, ubicado en el sector izquierdo de la ventana gráfica.

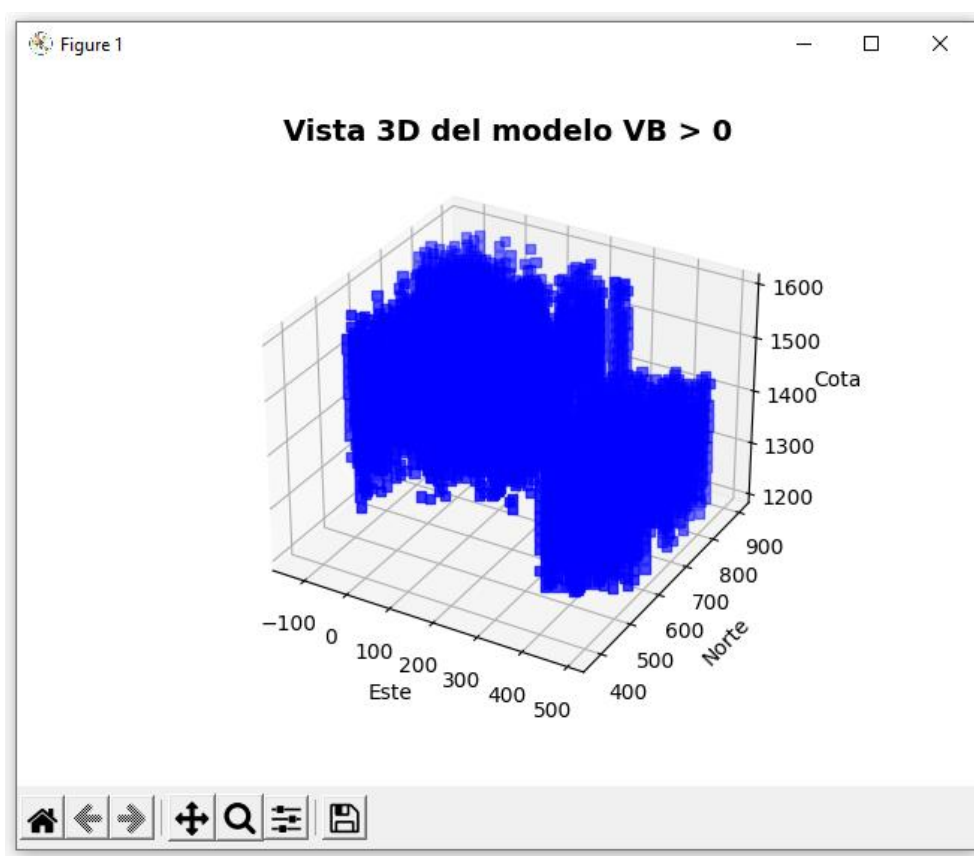


Figura 10: Visualización tridimensional de los bloques positivos del Modelo 1.

6.3.2. Óptimos por columna

A continuación, se presenta el gráfico que posee la visualización tridimensional que muestra el punto óptimo por columna. Para visualizar esta gráfica, se debe presionar el botón denominado “3D”.

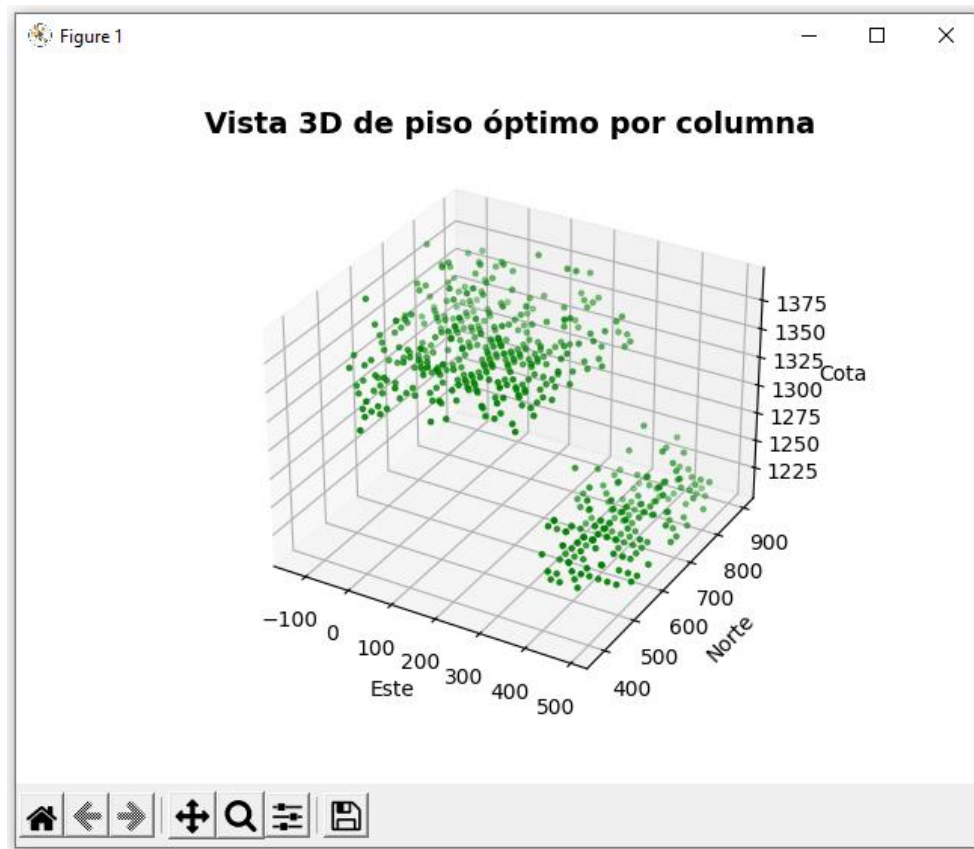


Figura 11: Visualización tridimensional de los pisos óptimos por columnas del Modelo 1.

6.3.3. Piso óptimo

En la siguiente figura, es posible observar una gráfica en dos dimensiones que posee una gradiente de calor que entrega información relacionada al nivel donde se ubica el piso óptimo de cada columna del modelo analizado. Para lograr visualizar esta gráfica, se debe presionar el botón denominado “Piso óptimo”.

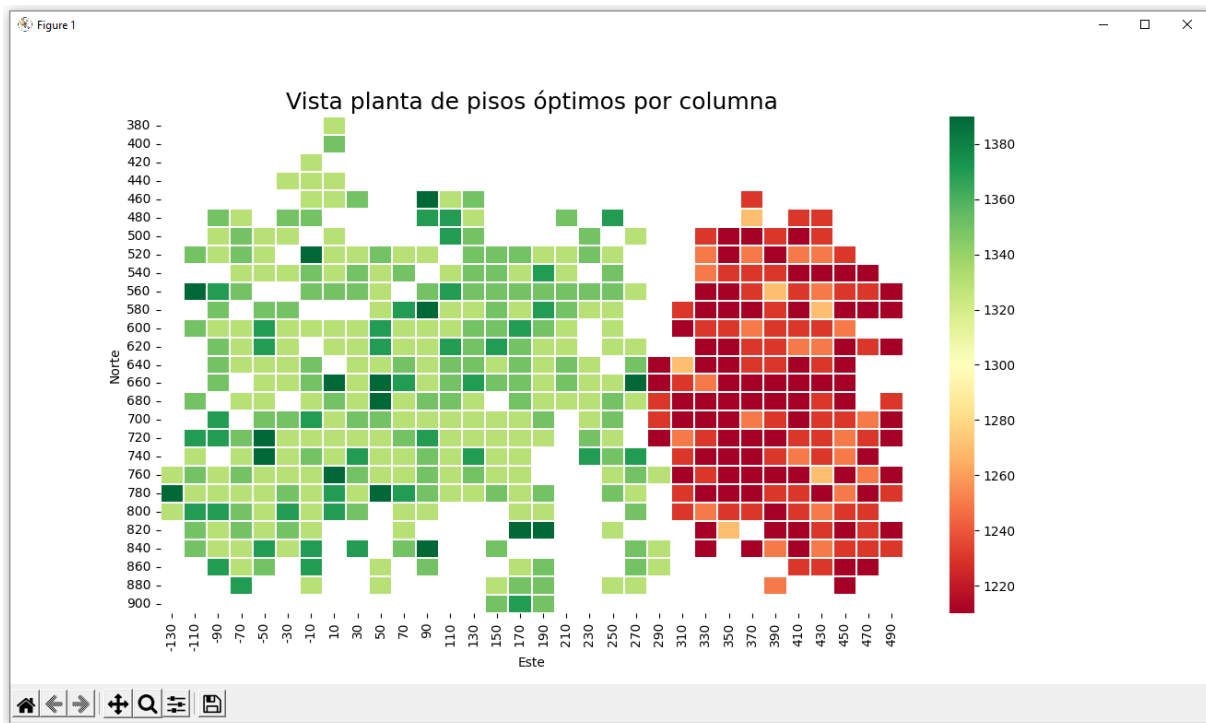


Figura 12: Vista planta de los óptimos por columna del Modelo 1.

6.3.4. Beneficio en función del piso

Para poder obtener el máximo beneficio del yacimiento, es que se hizo la evaluación en dos escenarios posibles.

El primer escenario realiza la evaluación económica del yacimiento considerando un nivel único de piso óptimo, para ello se obtendrá una sola gráfica de beneficios en funciones del piso. El segundo escenario considera la creación de dos niveles óptimos y se obtiene una gráfica para cada uno.

A continuación, se realizará la evaluación de ambos escenarios indicados.

a) Nivel único

La curva obtenida al realizar la evaluación del yacimiento considerando un único nivel óptimo es la que se presenta a continuación:

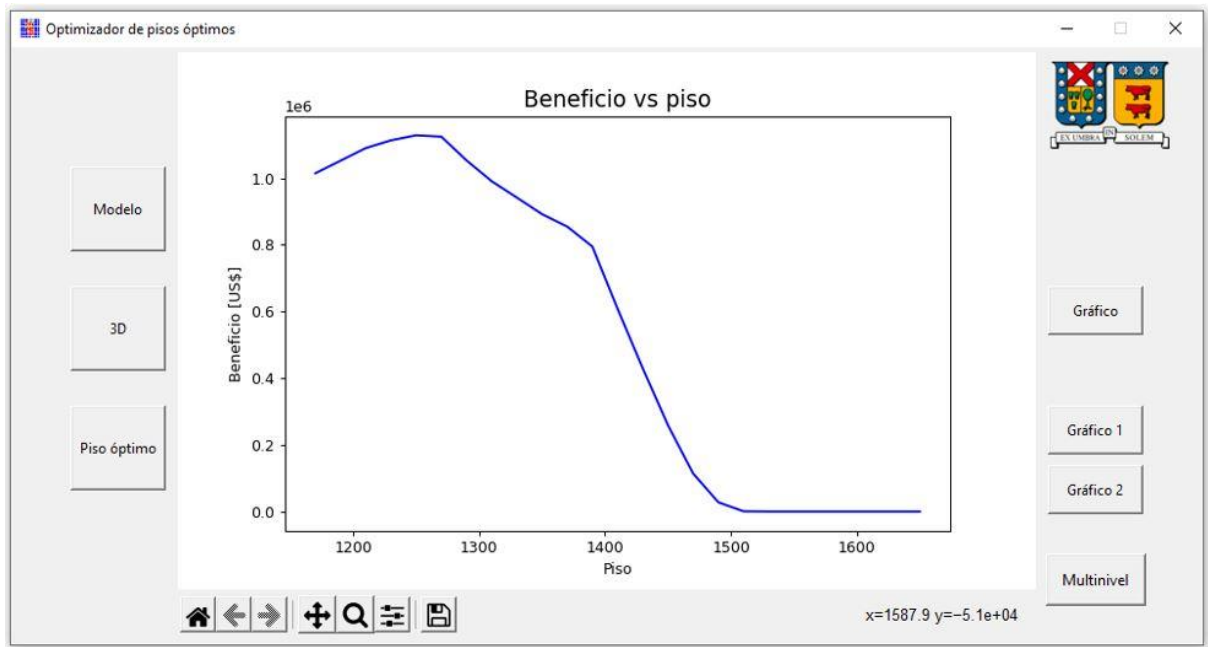


Figura 13: Gráfica beneficio vs piso del Modelo 1.

La figura anterior presenta una curva que alcanza su máximo en la cota 1250 y donde posee un beneficio de 1.128.554 [US\$].

b) Multinivel

Posteriormente se realiza la evaluación del yacimiento con dos niveles óptimos donde se obtienen las siguientes cifras.

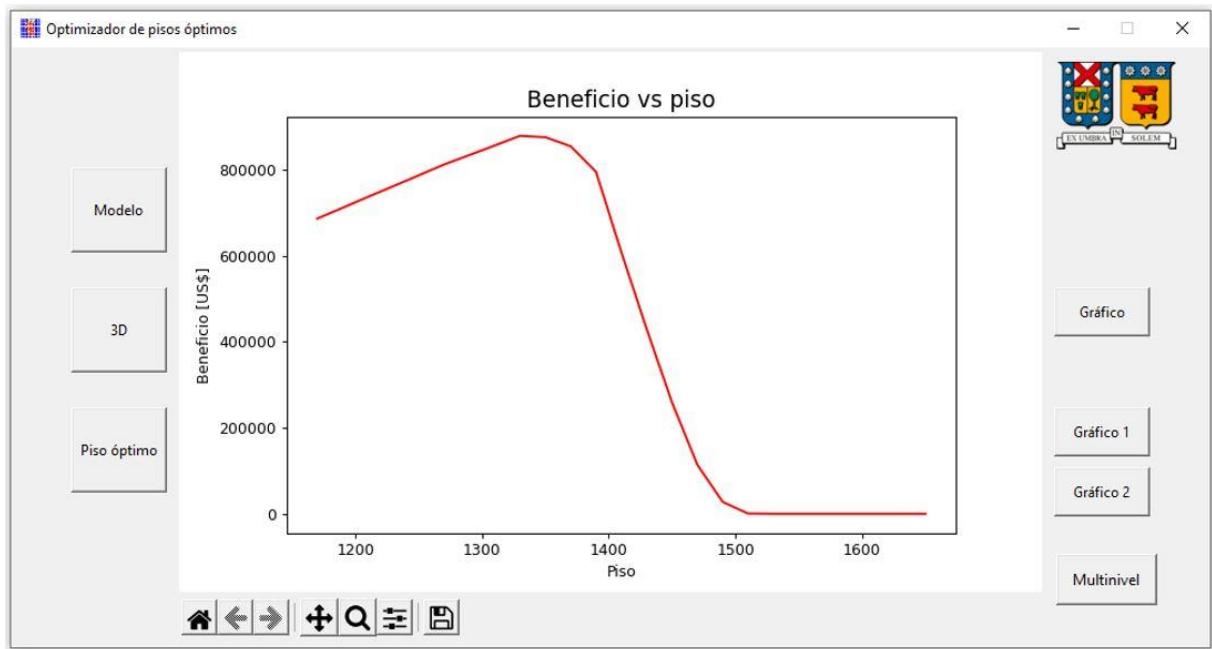


Figura 14: Gráfica de beneficio vs piso del Modelo 1.

La figura anterior presenta el primer nivel considerado en el cálculo y el cual presenta un máximo en el nivel 1330 y posee un beneficio de 878.436 [US\$].

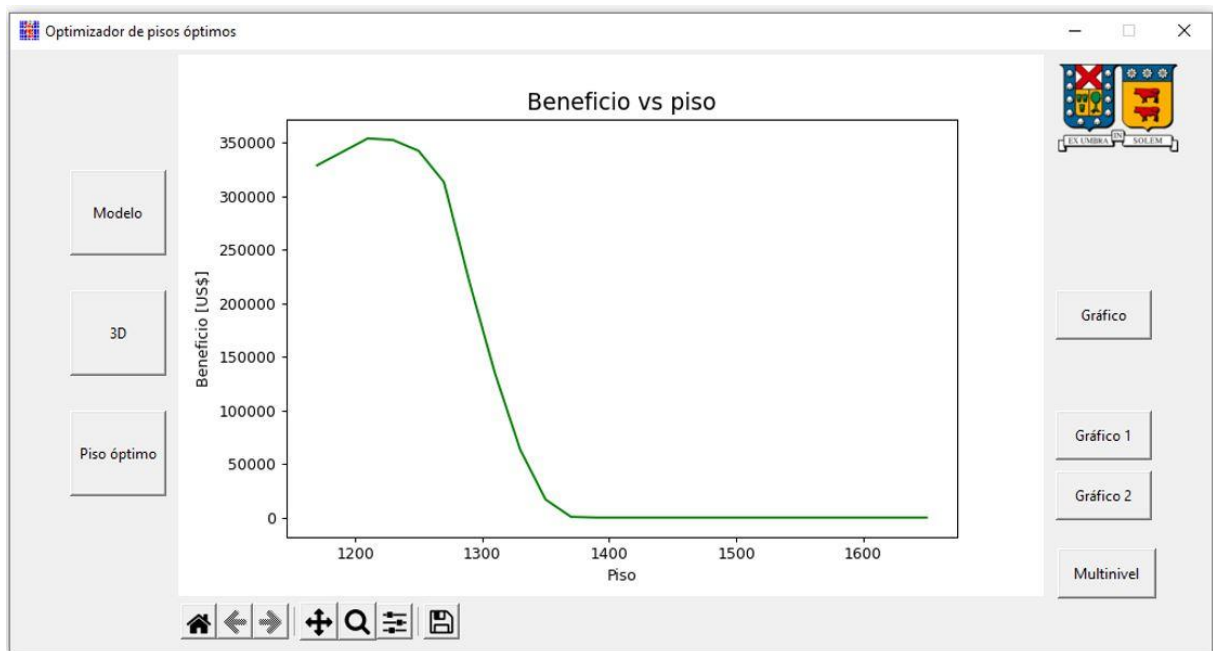


Figura 15: Gráfica de beneficio vs piso del Modelo 1.

La figura anterior presenta el segundo nivel considerado en el cálculo y el cual presenta un máximo en el nivel 1210 y posee un beneficio de 353.665 [US\$].

La figura a continuación representa las columnas pertenecientes a cada nivel.

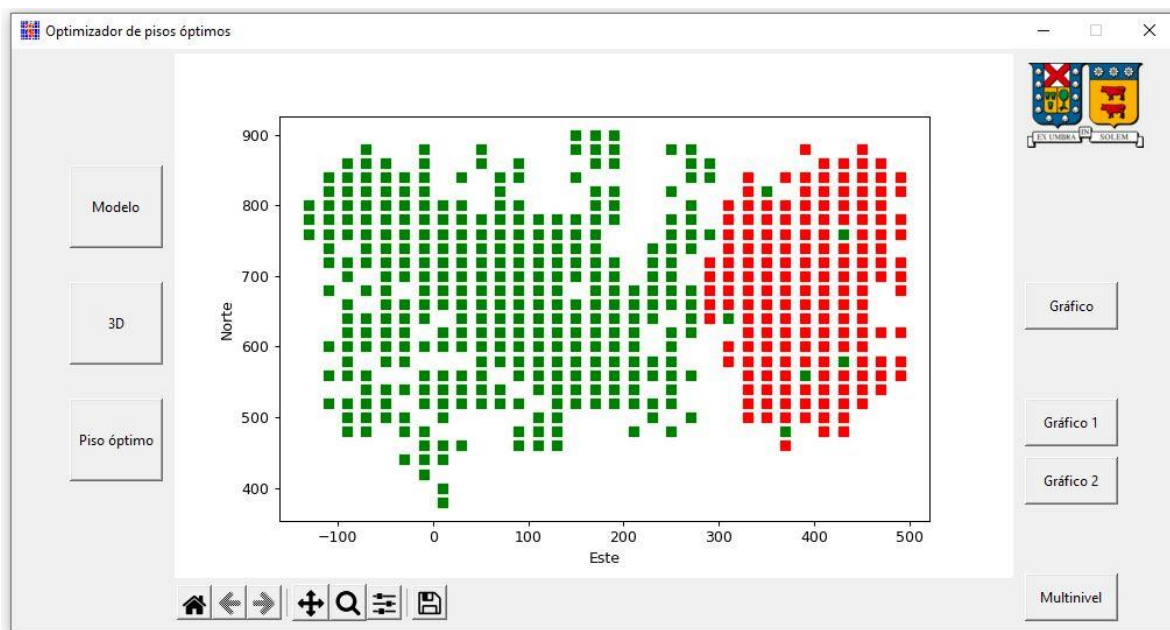


Figura 16: Visualización de las columnas pertenecientes a cada nivel.

6.4. Modelo 2

El segundo modelo que será evaluado mediante el optimizador de pisos óptimos posee las siguientes características:

Tabla 3: Características del Modelo 2.

N° bloques		
474.848		
Modelo	Mínimo	Máximo
Este	410610	412110
Norte	7039810	7041240
Cota	805	1675

6.4.1. Modelo

La figura a continuación presenta una vista tridimensional de los bloques positivos del modelo.

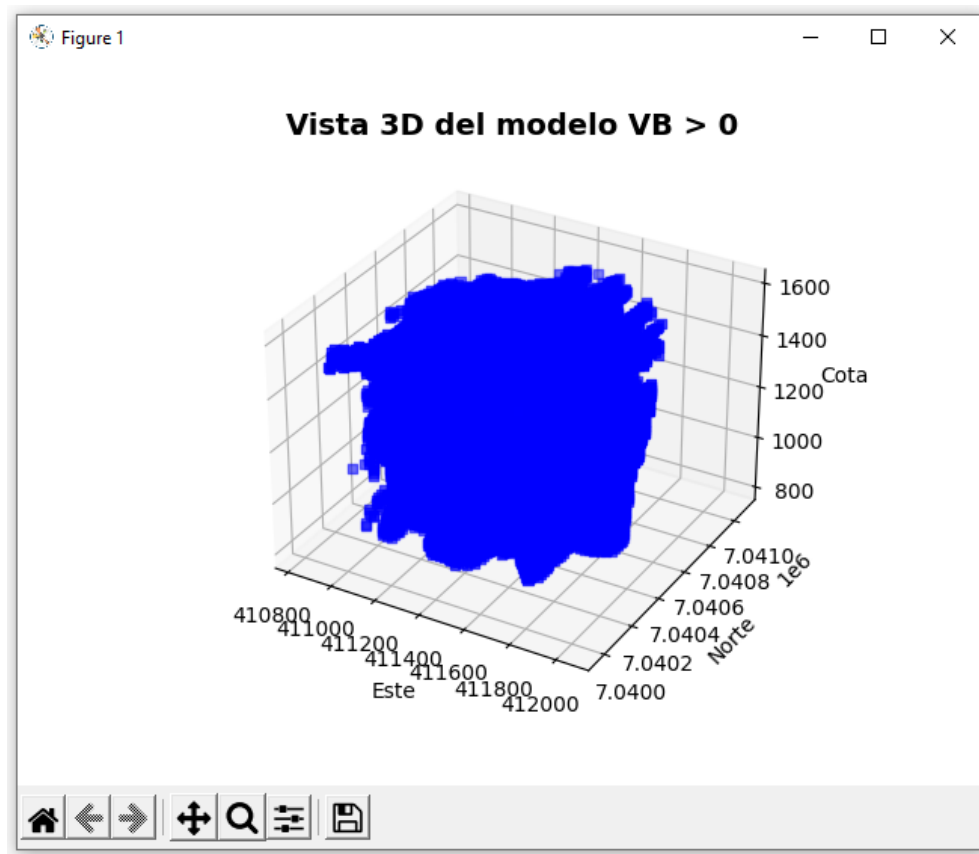


Figura 17: Visualización tridimensional de los bloques positivos del Modelo 2.

6.4.2. Óptimos por columna

La Figura 18 entrega una visualización en tres dimensiones de los óptimos por columna que presenta el modelo.

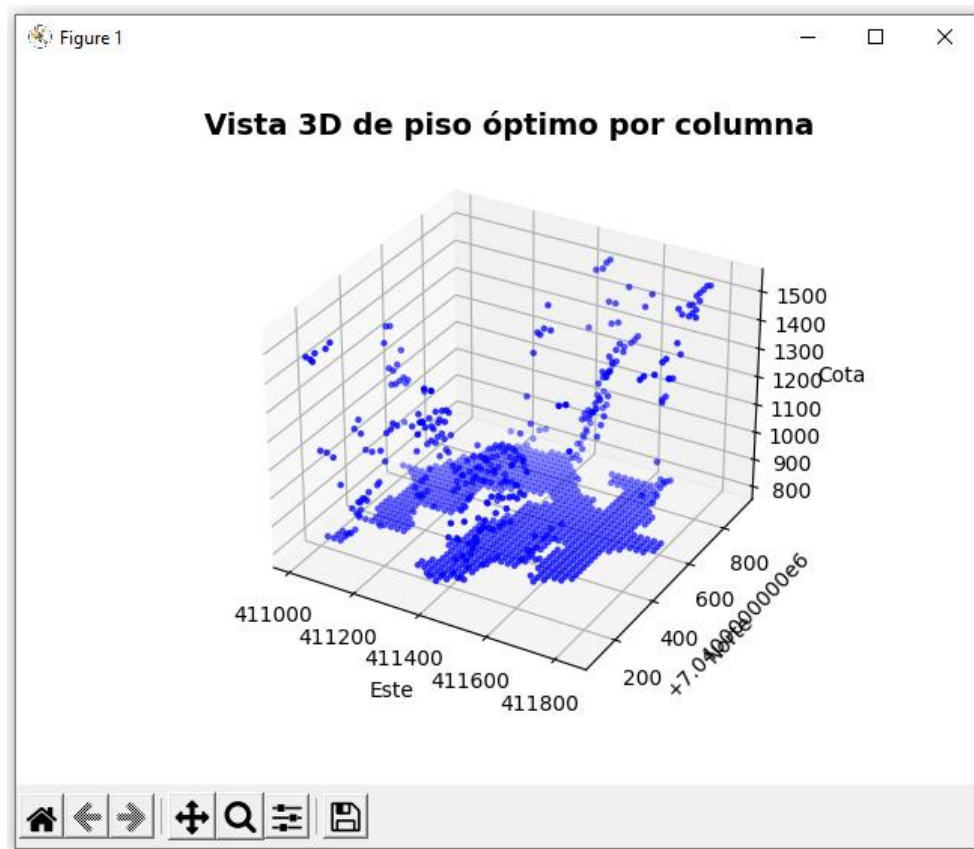


Figura 18: Visualización tridimensional de los pisos óptimos por columnas del Modelo 2.

6.4.3. Piso óptimo

En la siguiente figura es posible observar una gráfica en dos dimensiones que posee una gradiente de calor que entrega información relacionada con el nivel donde se ubica el piso óptimo de la columna del modelo analizado.

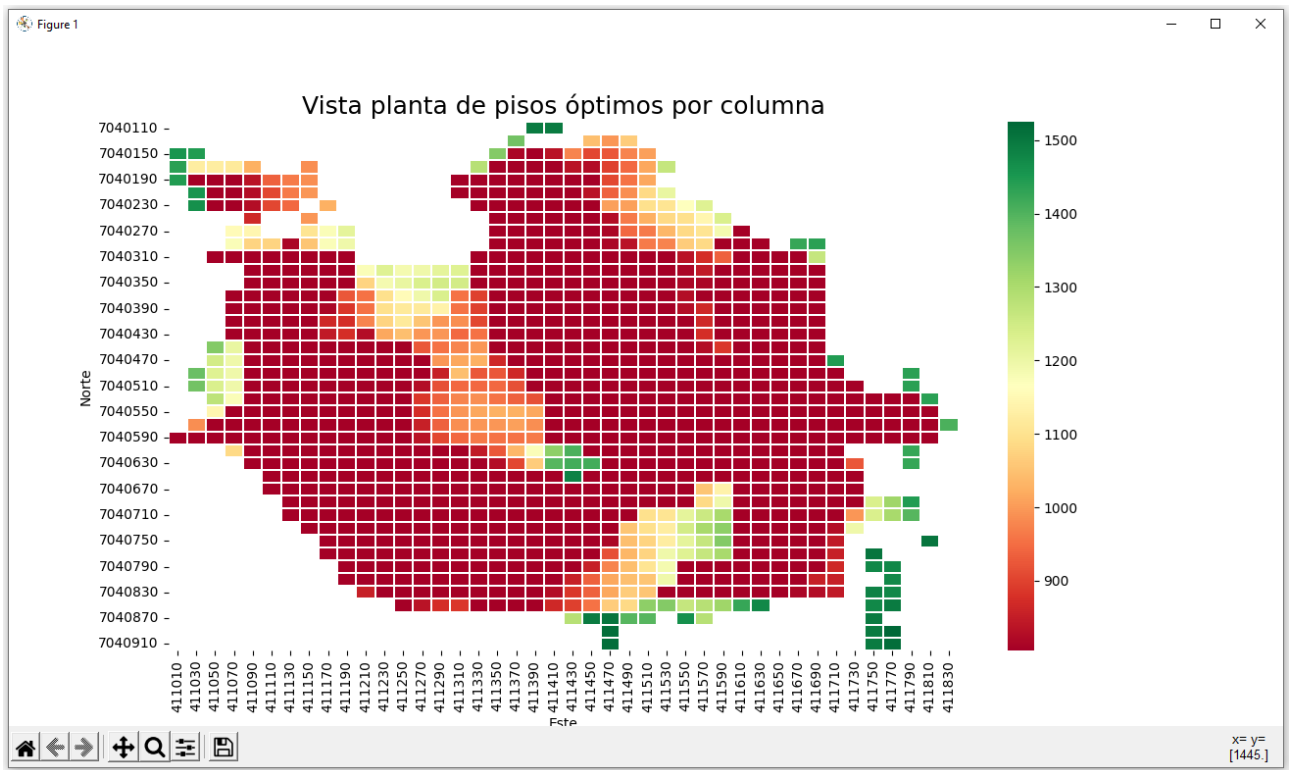


Figura 19: Vista planta de los óptimos por columna del Modelo 2.

6.4.4. Beneficio en función del piso

Al igual que en el modelo analizado anteriormente, para poder determinar el máximo beneficio del yacimiento, es que se realiza la evaluación en dos escenarios posibles.

El primer escenario realiza la evaluación económica del yacimiento considerando un nivel único de piso óptimo, para ello se obtendrá una sola gráfica de beneficios en funciones del piso. El segundo escenario considera la creación de dos niveles óptimos y se obtiene una gráfica para cada uno.

A continuación, se realizará la evaluación de ambos escenarios indicados.

a) Nivel único

La curva obtenida al realizar la evaluación del yacimiento considerando un único nivel es la siguiente:

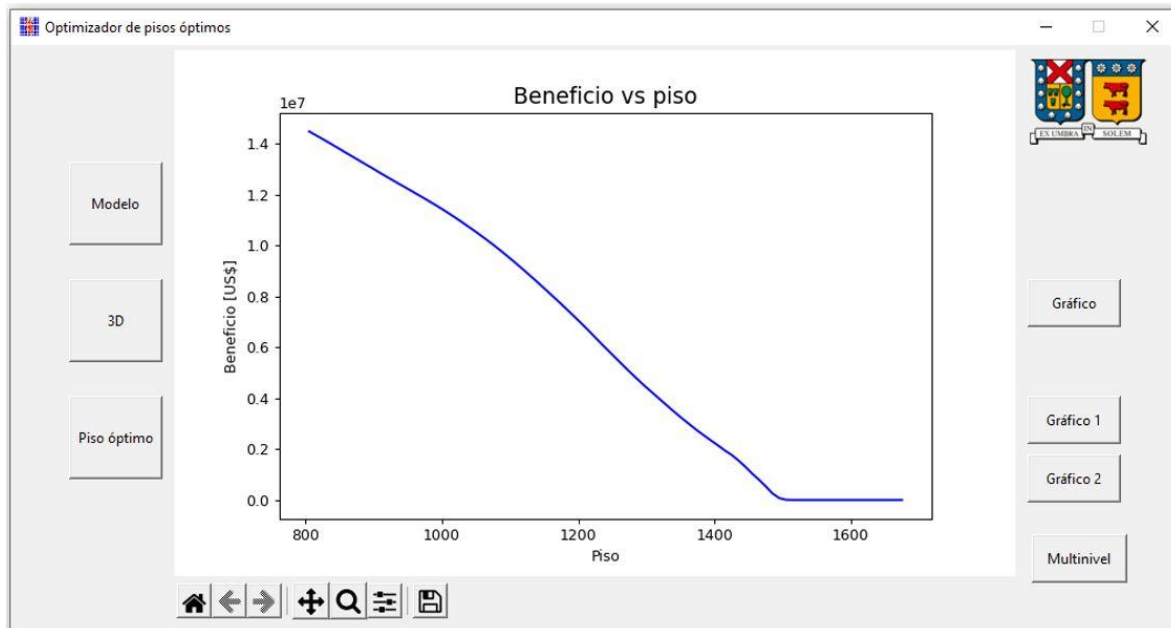


Figura 20: Gráfica de beneficio vs piso del Modelo 2.

La figura anterior presenta el segundo nivel considerado en el cálculo y el cual presenta un máximo en el nivel 805 y posee un beneficio de 14.475.885 [US\$].

b) Multinivel

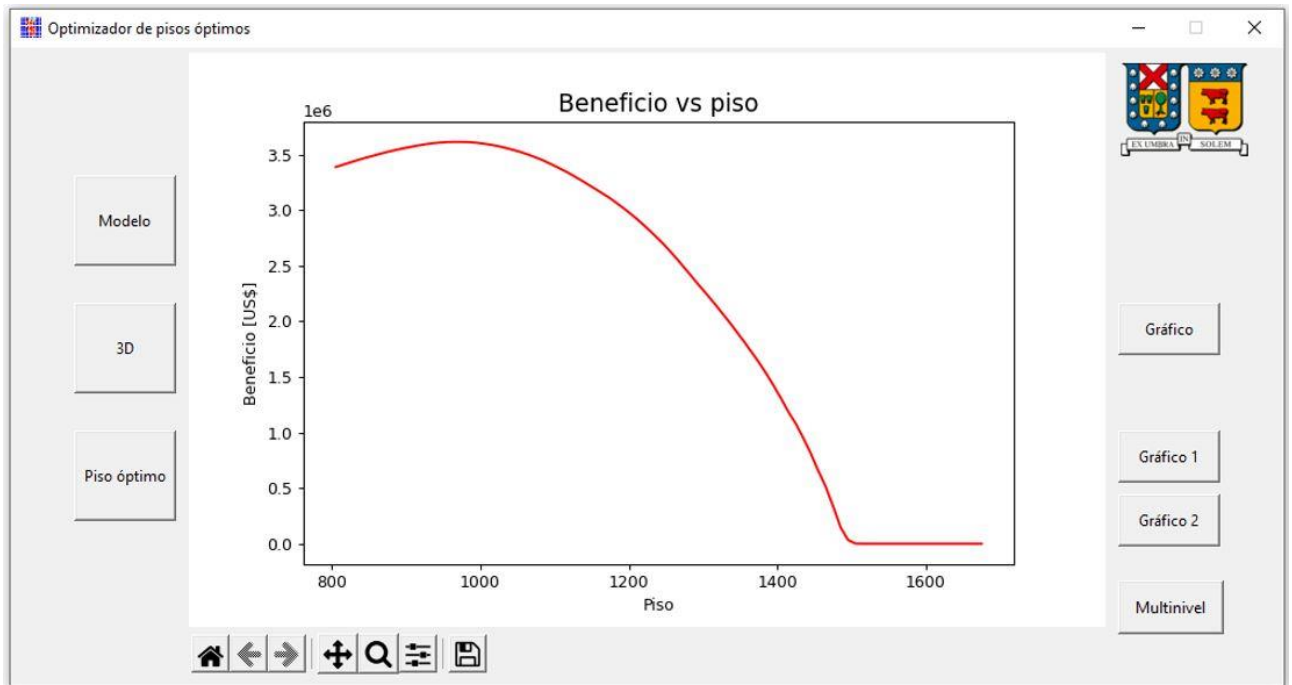


Figura 21: Gráfica de beneficio vs piso del Modelo 2.

La figura anterior presenta el segundo nivel considerado en el cálculo y el cual presenta un máximo en el nivel 965 y posee un beneficio de 3.615.460 [US\$].

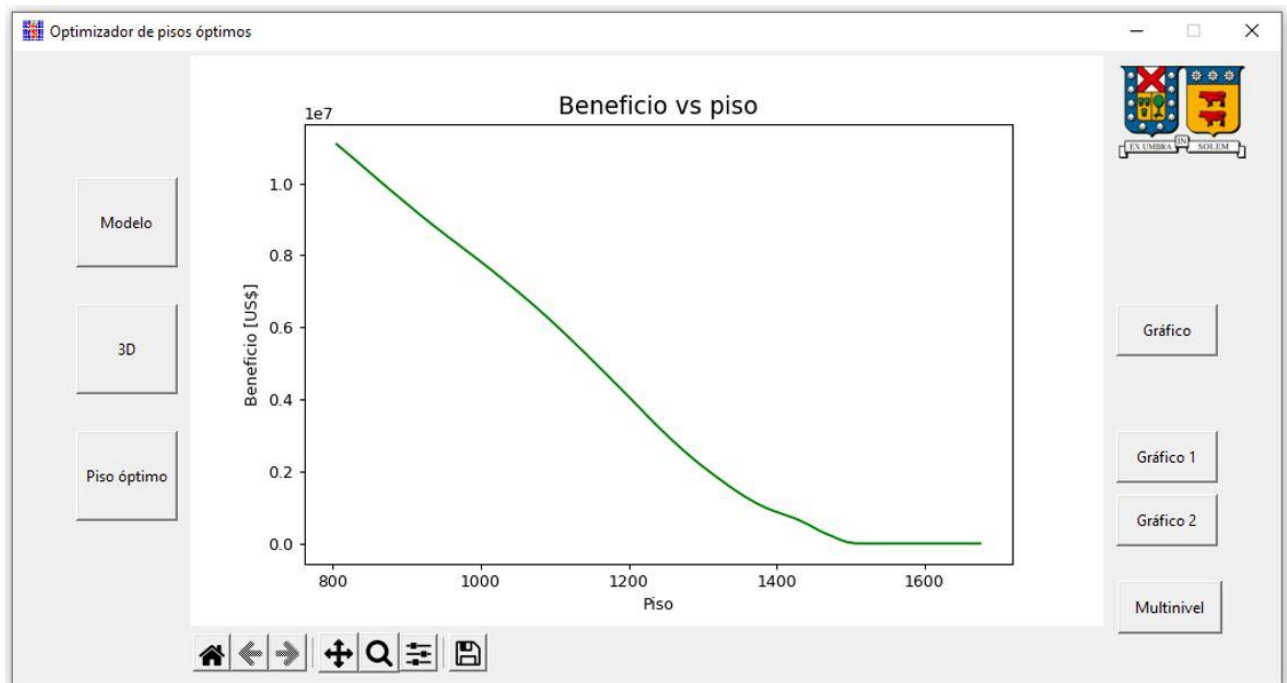


Figura 22: Gráfica de beneficio vs piso del Modelo 2.

La figura anterior presenta el segundo nivel considerado en el cálculo y el cual presenta un máximo en el nivel 805 y posee un beneficio de 11.086.470 [US\$].

La figura a continuación representa las columnas pertenecientes a cada nivel.

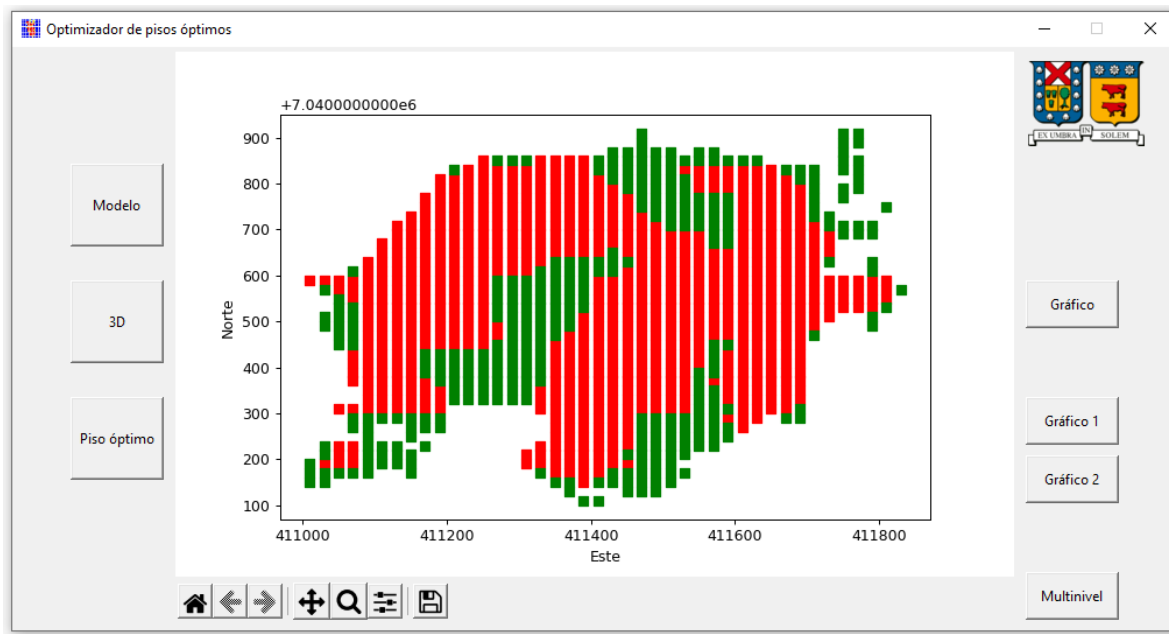


Figura 23: Visualización de las columnas pertenecientes a cada nivel.

7. Análisis

A partir de los resultados obtenidos en ambos modelos evaluados en el optimizador de pisos óptimos, es necesario realizar un análisis más detallado para definir finalmente el piso óptimo con un único nivel o multinivel.

7.1. Modelo 1

El primer modelo evaluado, desde este momento denominado en un futuro como Modelo 1, contaba con una extensión de 640 metros en el plano este, 520 metros de extensión norte y 480 metros de extensión en cota. El Modelo 1 posee 22.275 bloques con dimensiones de 20 x 20 x 20.

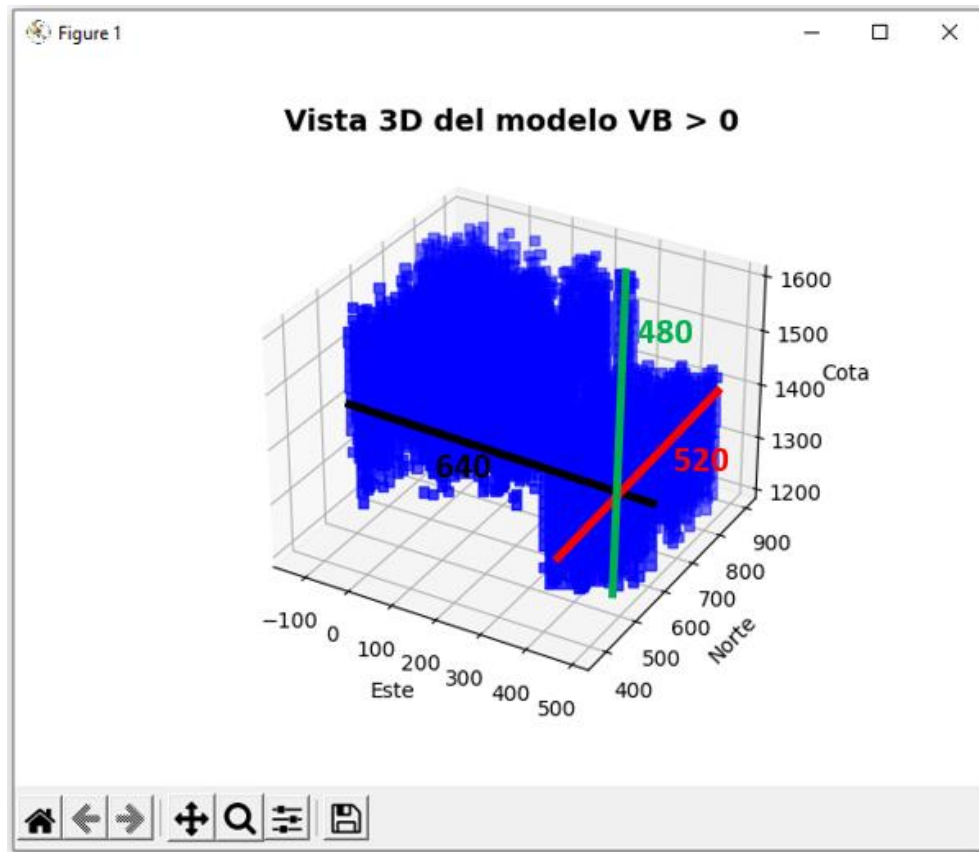


Figura 24: Extensión del Modelo 1.

En la Figura 24 se puede observar una primera visualización del modelo que se está trabajando. Esta visualización involucra a todos los bloques que poseen un valor positivo.

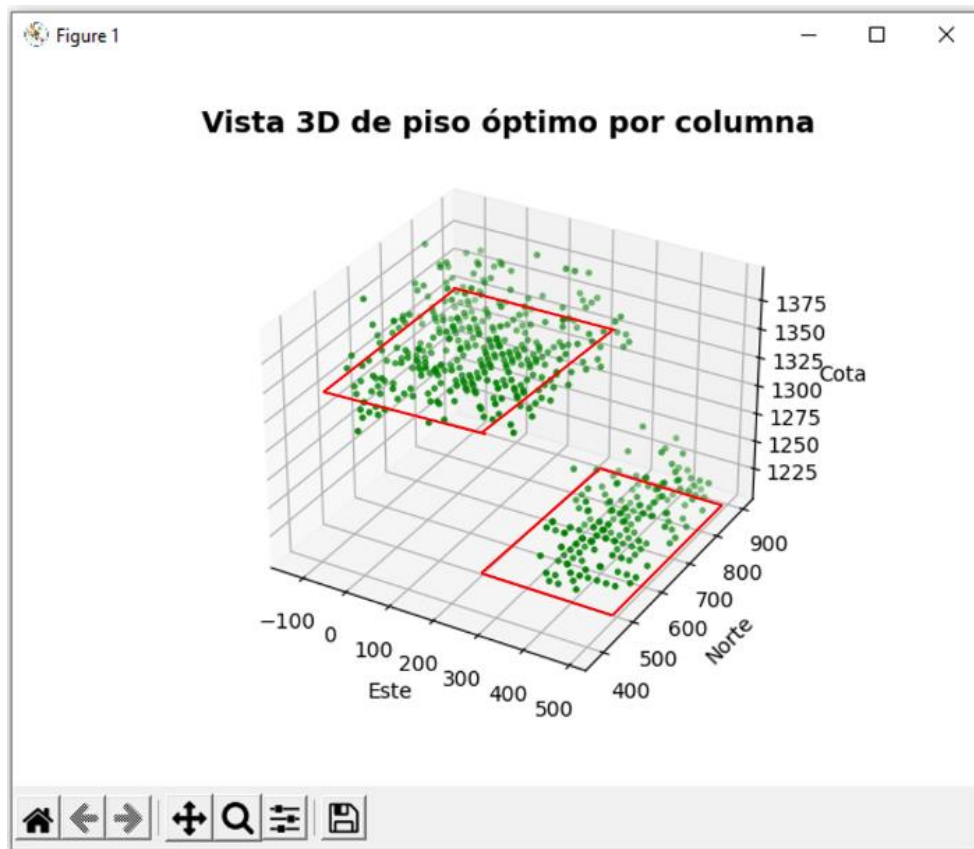


Figura 25: Visualización tridimensional de los pisos óptimos por columna.

La siguiente visualización correspondió a la Figura 25, la cual corresponde al primer acercamiento al resultado final que se espera obtener. Esta gráfica permite observar en conjunto y en tres dimensiones los óptimos de cada columna. Particularmente en este caso, desde esta vista, se pueden identificar dos niveles en los que los óptimos poseen una tendencia, sin embargo, no es posible identificar exactamente a que niveles corresponden, pues para ello se requiere de un análisis numérico.

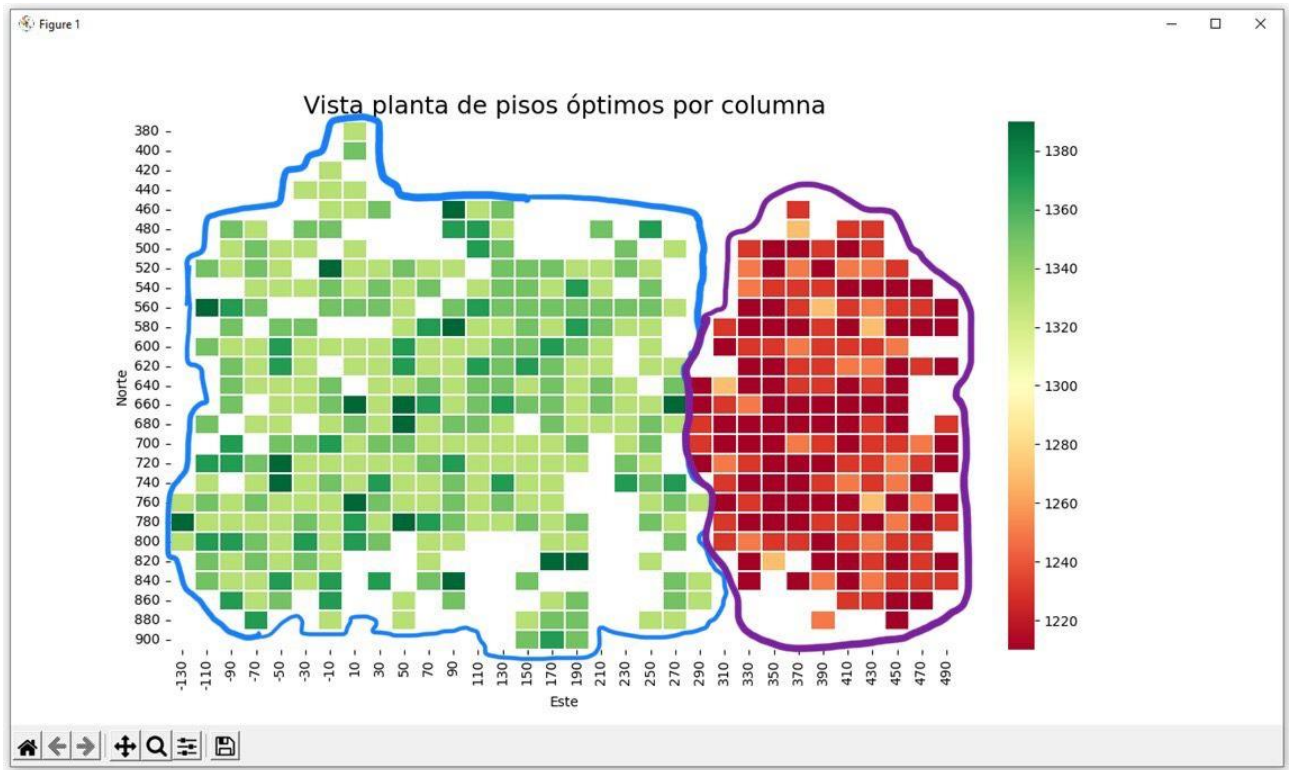


Figura 26: Piso óptimo del Modelo 1.

La tercera visualización correspondiente a la Figura 26, muestra una vista en dos dimensiones, de las cotas óptimas por columnas y entrega esta información en una gradiente de color, en este gráfico se puede identificar la preferencia a los dos niveles, pues están claramente definidos en el color verde y el color rojo, sin embargo, para poder determinar la existencia de estos dos niveles se debe de realizar la evaluación numérica en ambos casos, además de un análisis para determinar si es factible desarrollar los dos niveles.

La evaluación numérica desarrollada se llevó a cabo en los dos contextos mencionado anteriormente. La primera evaluación considera un piso óptimo de un solo nivel, mientras que la segunda evaluación considera la existencia de dos niveles.

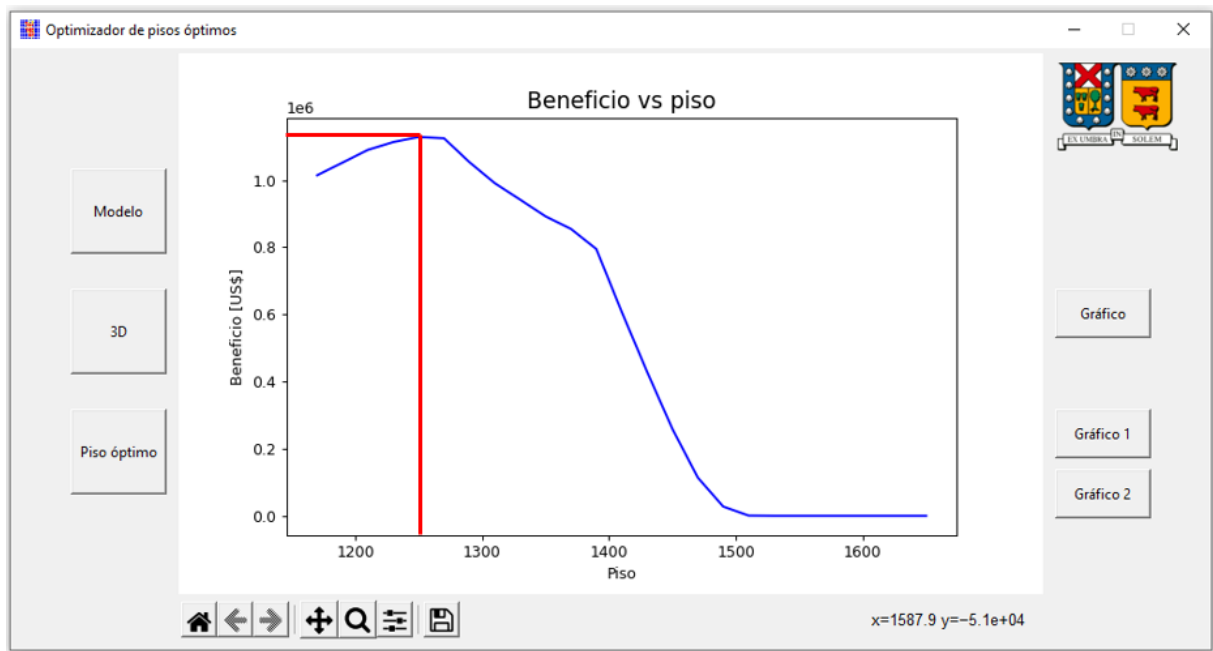


Figura 27: Curva de beneficio vs piso.

La Figura 27 entrega la curva del yacimiento generada al contemplar un único piso, esta curva presenta la cúspide en la cota 1250, entregando un valor de beneficio de 1.128.554 [US\$]. Es decir, si se considera únicamente la construcción de un nivel, este debe realizarse en el piso 1250 para obtener el máximo beneficio posible.

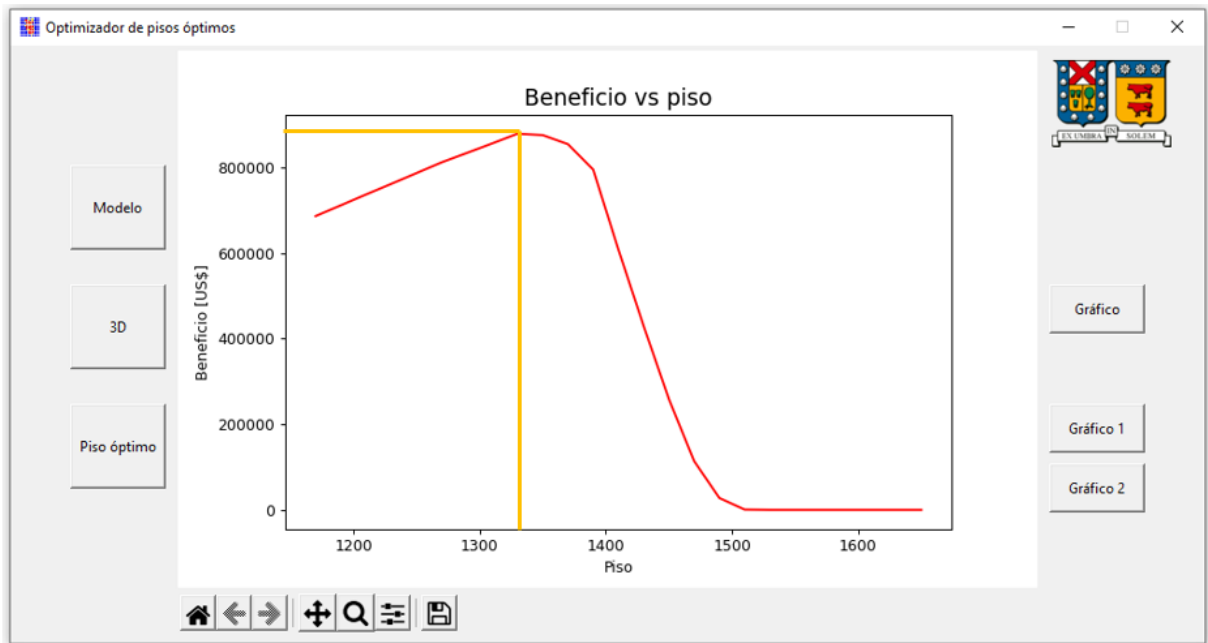


Figura 28: Curva de Beneficio vs piso.

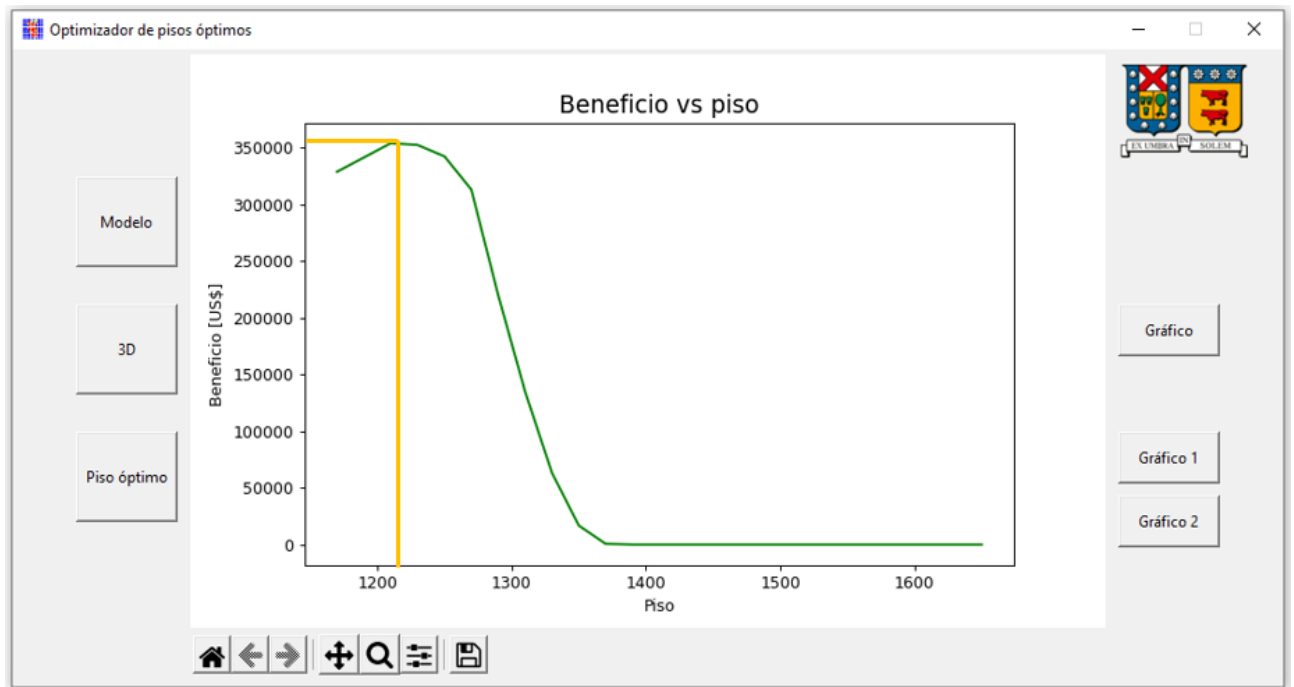


Figura 29: Curva de beneficio vs piso.

Por otra parte, tanto la Figura 28 como la Figura 29 evalúan la posibilidad de desarrollar dos niveles, ambas presentando la curva de su nivel respectivo.

La Figura 28 entrega la primera curva, la cual y de acuerdo con las consideraciones tomadas, entrega una evaluación realizada solo a aquellas columnas que poseen un piso óptimo que se encuentra por sobre la cota 1250. Esta gráfica alcanza su máximo en la cota 1330 y posee un beneficio de 878.436 [US\$].

Por el contrario, la Figura 29 entrega la evaluación de las columnas que poseen un piso óptimo por debajo de los 1250. La gráfica presentada en dicha figura alcanza su punto óptimo en la cota 1210 y posee un beneficio de 353.665 [US\$].

Para determinar si este modelo posee un piso óptimo de un solo nivel o multinivel, se deben realizar dos pasos fundamentales. El primer paso es realizar la comparación entre los beneficios obtenidos en ambos casos. El segundo paso es verificar la compatibilidad del desarrollo de los niveles.

Tabla 4: Comparación de beneficios entre creación de un nivel y un multinivel.

Modelo 1	Nivel único	Multinivel	
Cota	1250	1250	1330
Beneficio [US\$]	1.128.554	1.232.199	

El piso óptimo obtenido en la primera evaluación, es decir, considerando un solo nivel, es de 1.128.554 [US\$], mientras que el beneficio obtenido evaluando el yacimiento como multinivel es de 1.232.199 [US\$].

De acuerdo con los resultados anteriores, ejecutar el piso óptimo multinivel entrega un beneficio mayor, sin embargo, también la realización de estos debe ser factible, para ello se debe evaluar la Figura 26, y donde se ve claramente la predominancia de un nivel en el sector izquierdo del modelo, así como ocurre lo mismo en el sector derecho del modelo con el otro nivel.

Tomando en consideración lo mencionado anteriormente, es factible desarrollar dos niveles en el Modelo 1. El primer nivel estará localizado en la cota 1330, mientras que el segundo nivel se encontrará en la cota 1210.

En Anexo 12 se puede encontrar los resultados obtenidos al realizar la obtención del piso óptimo a través de una plataforma manual como *Excel*, y donde puede apreciarse una diferencia cercana al 1% entre lo obtenido de forma manual y con el seleccionador de piso óptimo.

7.2. Modelo 2

El segundo modelo evaluado, desde ahora en adelante denominado Modelo 2 contaba con una extensión de 1500 en el plano este, 1400 metros de extensión norte y 870 de extensión en la cota. El Modelo 2 posee 474.848 bloques con dimensiones de 20 x 20 x 10.

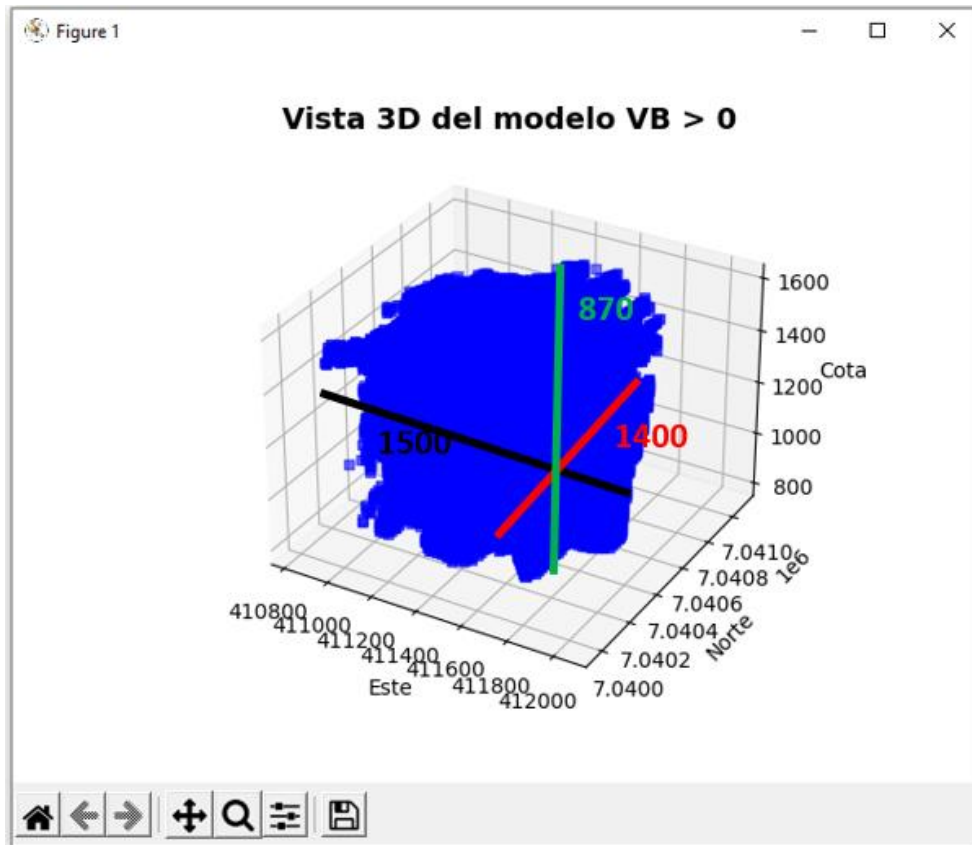


Figura 30: Extensión del Modelo 2.

En la Figura 30 es posible apreciar una vista tridimensional de los bloques que poseen un valor positivo dentro del modelo.

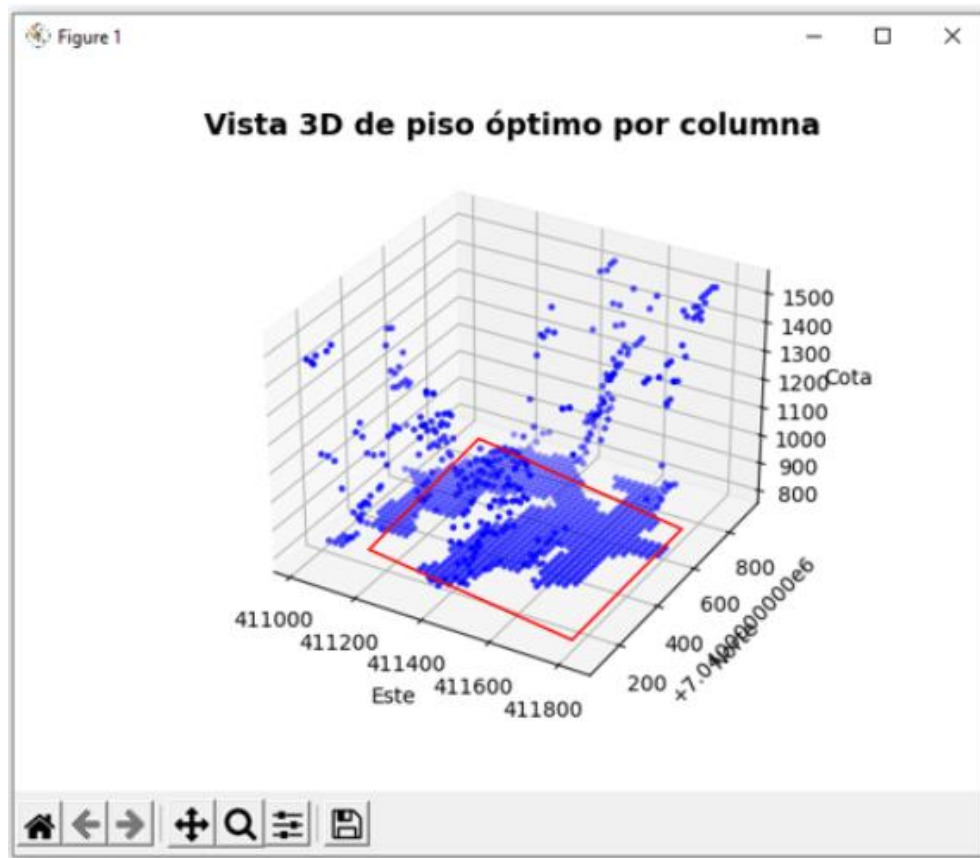


Figura 31: Visualización tridimensional de los pisos óptimos por columna.

A continuación, en la Figura 31 es posible contemplar nuevamente en una vista tridimensional los pisos óptimos por cada columna, en la cual se puede apreciar una tendencia a presentar un solo piso óptimo ubicado en la zona más baja del modelo, sin embargo, no se podría determinar con esta vista la cota exacta del piso.

Una tercera visualización se puede encontrar en la Figura 32, en donde se observa una vista en dos dimensiones de las columnas con su cota óptima respectiva, dicha información esta proporcionada a través de una gradiente de calor. Particularmente en este modelo se puede apreciar una preferencia por el color rojo, lo cual indica que la mayoría de los pisos óptimos se encontraría en la zona baja del modelo.

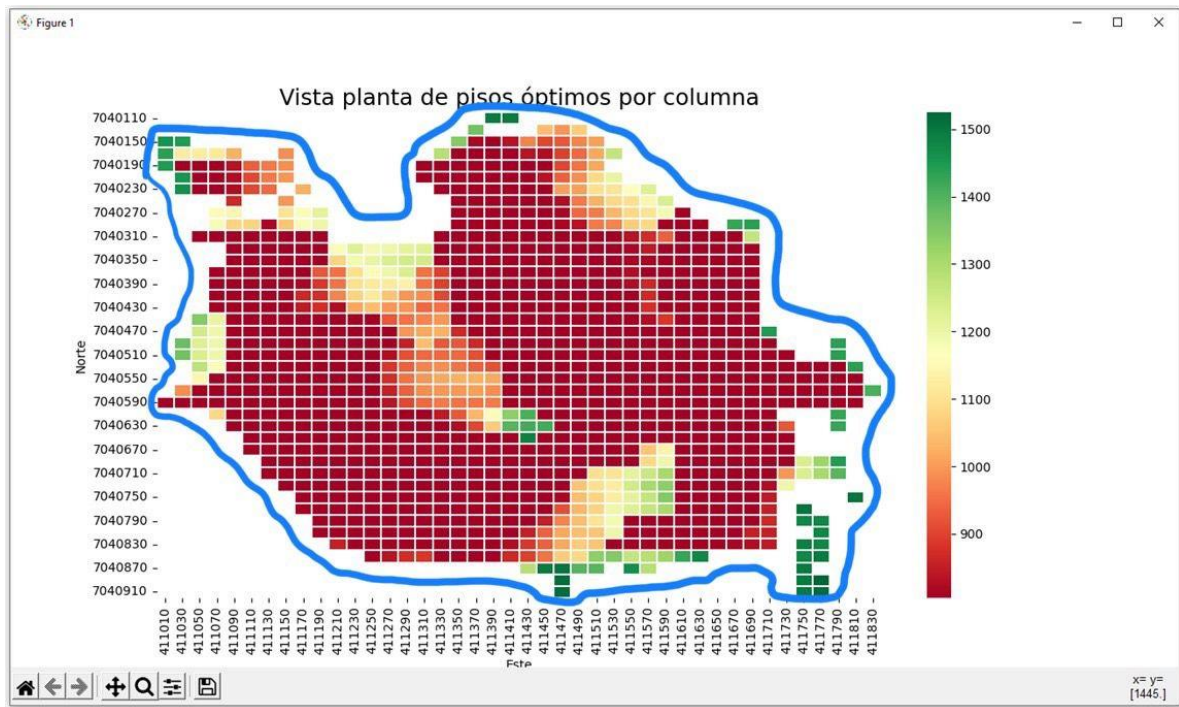


Figura 32: Piso óptimo del Modelo 2.

Para lograr definir si el yacimiento presenta uno o dos niveles como piso óptimo, es que se debe realizar una evaluación numérica. Esta evaluación se realizará bajo dos escenarios, al igual que se realizó en el Modelo 1. El primer escenario involucra una evaluación considerando un único piso, mientras que el segundo realiza la misma evaluación considerando dos niveles.

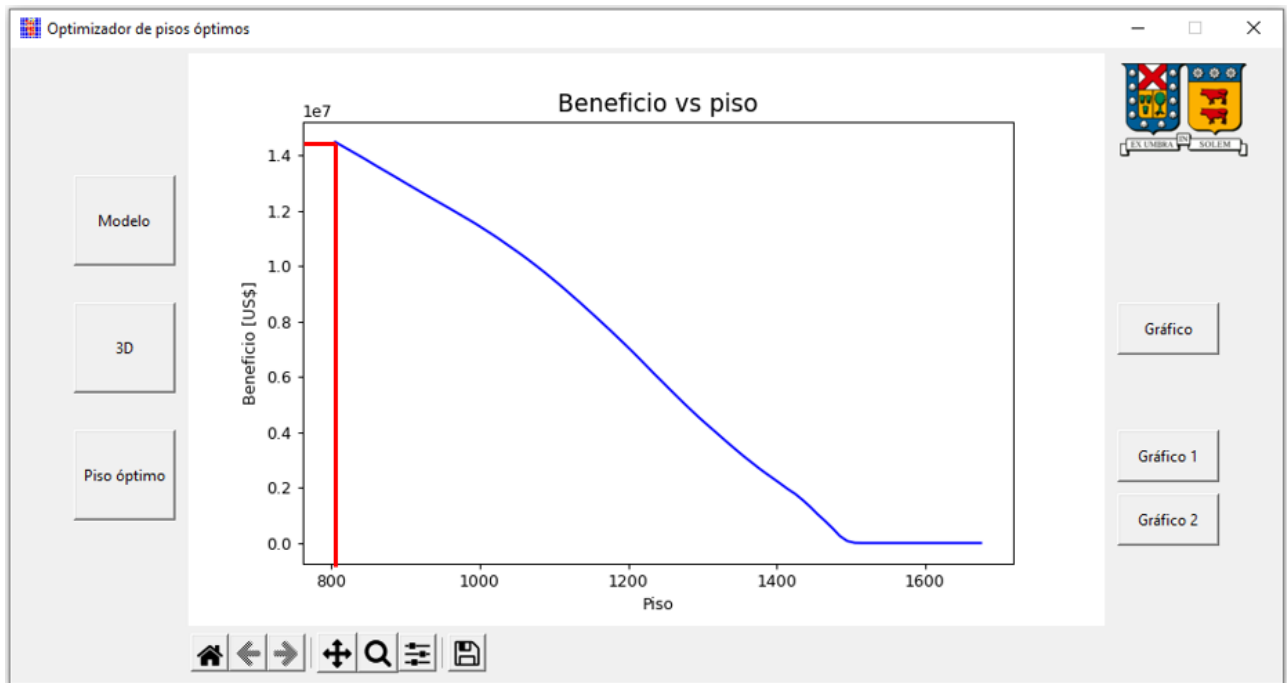


Figura 33: Curva de beneficio vs piso.

En la Figura 33 se puede observar la gráfica obtenida luego de evaluar el primer escenario, es decir contemplando el uso de un único nivel como piso óptimo, esta curva presenta un máximo en la cota 805 y posee un beneficio asociado de 14.475.885 [US\$].

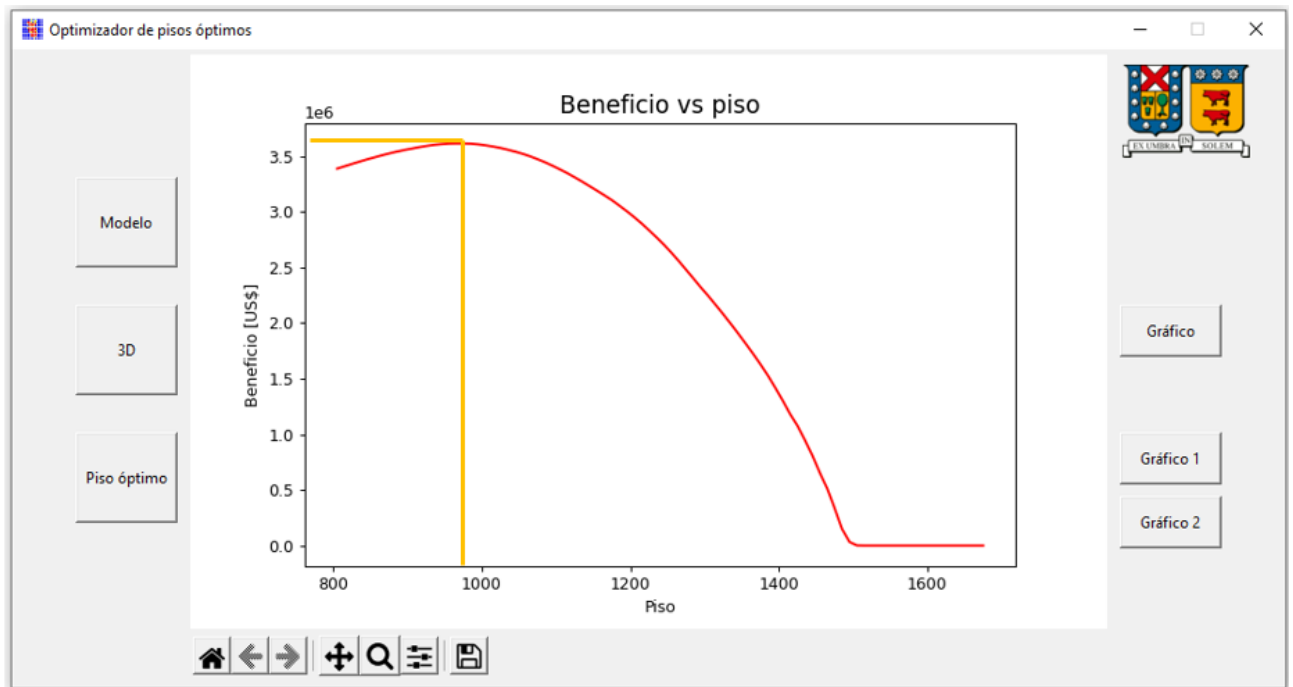


Figura 34: Curva de beneficio vs piso.

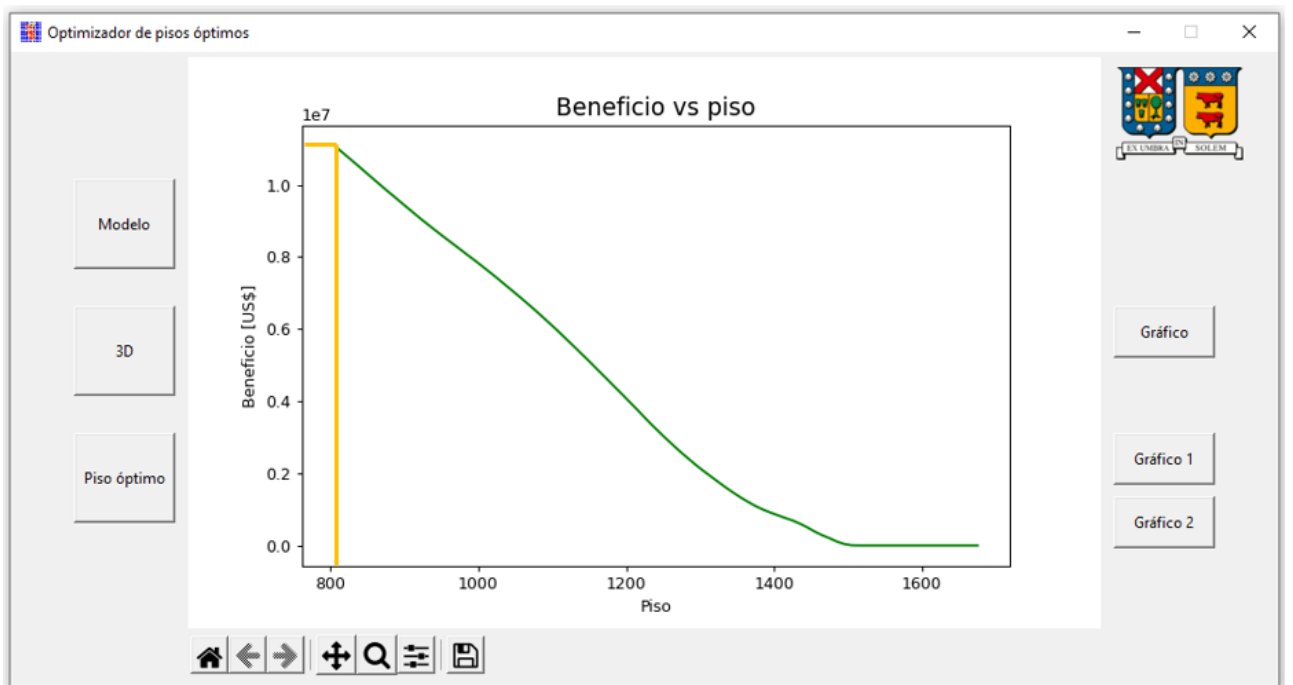


Figura 35: Curva de beneficio vs piso.

Los resultados de la evaluación en el segundo escenario pueden verse tanto en la Figura 34 como en la Figura 35.

La Figura 34 entrega la primera curva, la cual de acuerdo con las consideraciones que fueron tomadas inicialmente, hace una evaluación solo de aquellas columnas que posee un piso óptimo que se encuentra por sobre la cota 805. La gráfica obtenida presenta en la figura alcanza su punto máximo en la cota 965 y posee un beneficio de 3.615.460 [US\$].

Por el contrario, la Figura 35 entrega la segunda curva, correspondiente a la evaluación realizada en el segundo escenario, es decir, con las columnas que poseen un piso óptimo que se encuentra por debajo de la cota 805, incluyendo esta misma. La gráfica obtenida alcanza su máximo en la cota 805 y posee un beneficio de 11.086.470 [US\$].

Para determinar si lo óptimo para el modelo es tener un solo nivel o dos, se deben realizar dos pasos fundamentales. El primer paso corresponde a la comparación entre los beneficios de ambos casos. El segundo paso corresponde a la compatibilidad del desarrollo de los niveles.

Tabla 5: Comparación de beneficios entre creación de un nivel y un multinivel.

Modelo 2	Nivel único			Multinivel	
Cota	805			805	965
Beneficio [US\$]	14.475.885			14.071.930	

El piso óptimo obtenido en la primera evaluación, considerando un nivel único corresponde a 14.475.885 [US\$], mientras que el beneficio obtenido considerando dos niveles, es de 14.701.930 [US\$].

Al ejecutar el segundo paso, es decir comprobar en la Figura 32 si el desarrollo de dos niveles es viable, es que se concluye que en el Modelo 2 se debe construir un piso óptimo de un único nivel, pues el desarrollo de más niveles no sería factible, esto debido a que la zona en la que se desarrollaría un segundo nivel es pequeña y se encuentra dividida principalmente en dos zonas.

Tomando en cuenta lo anterior, es que en el modelo 2 es más factible realizar un piso óptimo de un solo nivel, el cual se encuentra en la cota 805 y posee un beneficio asociado de 14.475.885 [US\$].

Teniendo en consideración las gráficas obtenidas en este modelo, se debe poner especial atención en las cotas inferiores a 805 debido a que dichas gráficas no presentan un máximo notorio, por ello es que se podría inferir que la zona mineralizada se extiende más abajo del modelo de bloques, lo que requiere un mayor estudio geológico para así determinar el piso óptimo que entrega el mayor beneficio posible.

Los resultados a través de Excel no lograron obtenerse debido al tamaño de la base de datos.

IV. CONCLUSIONES

El proyecto desarrollado buscaba como objetivo principal la creación de un optimizador de piso óptimo económico aplicable a yacimientos mineros, diseñado a través del lenguaje de programación *Python*. El uso de este optimizador permitirá obtener resultados en un menor tiempo de procesamiento comparado con una plataforma como Excel, además de proporcionar una herramienta específica diseñada para el cálculo de dicho nivel.

La creación del optimizador fue diseñada a través de varios pasos, destacando los siguientes puntos:

- Valorización por piso dentro del código del optimizador.
- Creación de un nivel y un multinivel para respectiva comparación de beneficios.
- Elaboración de una interfaz gráfica que permita interacción entre el usuario y el optimizador.

Al cumplir en su totalidad los puntos anteriores, el optimizador creado es capaz de entregar un resultado del piso óptimo del modelo evaluado, proporcionando información visual como lo son las vistas tridimensionales del modelo y de los óptimos por columnas, además de información gráfica de los beneficios de cada piso. También es capaz de entregar una vista en planta del modelo indicando en una vista de color los óptimos. Agrupando esta información, el usuario es capaz de determinar si el modelo tiene un óptimo económico al realizar un multinivel o un solo nivel.

Por lo tanto, se debe concluir que el optimizador diseñado cumple a cabalidad con los objetivos que fueron planteados en un principio, además de que se realizó una validación de los datos mediante la plataforma de Excel, obteniéndose diferencias en torno al 1%.

V. REFERENCIAS

- [1] Weston J., (2016). “*The Evolution of Block Caving Technology*”. Seventh International Conference & Exhibition on Mass Mining.
- [2] Diering T., (2000), “*PC-BC: A Block Cave Design and Draw Control System*”, MassMin. 2000.
- [3] Calderon, J., (2011), “*Desarrollo de una herramienta para planificación de largo plazo en Block/Panel Caving*”, Memoria para optar al título de Ingeniero Civil de Minas, Universidad de Chile, 2011.
- [4] Vergara. D., (2014), “*Altura de columna en Block Caving*”, Memoria para optar al título de ingeniera Civil de Minas, Universidad de Chile, 2014.
- [5] Brazil, M. & Grossman P. (2008), “*Access layout optimisation for underground mines*”, Australian Mining Technology Conference, 2008.
- [6] Vargas, E., Morales N., & Emery X., “*Footprint and Economic Envelope Calculation for Block/Panel Caving Mines under Geological Uncertainty*”, Universidad de Chile.
- [7] Beard, D. & Brannon C., (2018), “*Grasberg Block Cave mine: cave planning and undercut sequencing*”, Australian Centre for Geomechanics, Perth, 2018.
- [8] Bennet W., (2020), “*Long term planning for a block cave project. Is the value being missed?*”, MassMin 2020.

VI. ANEXOS

Anexo 1: Creación de la ventana gráfica

```
ventana = Tk()
ventana.title('Costo de preparación por columna')
ventana.geometry("400x300")

num = IntVar()

textoN = Label(ventana, text= "Agrega el valor del costo de preparación por columna:")
textoN.place(x=70, y =70)

textoP = Label(ventana, text = "US$/COL")
textoP.place(x=280,y=100)

caja = Entry(ventana, textvariable=num)
caja.place(x=150,y=100)

boton1 = Button(ventana, text = "Ingresar", command= Asignar)
boton1.place(x=180, y=200)
ventana.config(bg="#fff")

ventana.mainloop()

def Asignar():
    Costo_prep_column = num
    ventana.destroy()
```

Anexo 2: Nueva Columna

```
i = 1
prev_row = [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]
for row in Hoja1.iter_rows():
    espacio = 'E' + str(i)
    i = i + 1
    if (row[0].value == prev_row[0] and row[1].value == prev_row[1]):
        prev_row[0] = row[0].value
        prev_row[1] = row[1].value
        Hoja1[espacio] = int(0)

    else:
        prev_row[0] = row[0].value
        prev_row[1] = row[1].value
        Hoja1[espacio] = int(1)
i=i+2
Hoja1[espacio]=1
Hoja1['E1'] = 'NewColumn'
```

Anexo 3: Cota positiva

```
i = 1

for row in Hoja1.iter_rows():
    espacio = 'F' + str(i)
    i = i + 1
    if (row[3].value == 'VB'):
        Hoja1['F1'] = 'Zpos'

    elif (row[3].value > 0):
        Hoja1[espacio] = row[2].value

    else:
        Hoja1[espacio] = int(0)
```

Anexo 4: Suma acumulativa

```
i = 1
prev_row = [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]
for row in Hoja1.iter_rows():
    espacio = 'G' + str(i)
    i = i + 1
    if (row[4].value == 'NewColumn'):
        Hoja1['G1'] = 'SumDown'

    elif (row[4].value == 1):
        prev_row[6] = row[3].value
        Hoja1[espacio] = row[3].value

    else:
        Hoja1[espacio] = int(prev_row[6]) + int(row[3].value)
        prev_row[6] = Hoja1[espacio].value
```

Anexo 5: Índice optimizador

```
i = 1
prev_row = [0,0,0,0,0,0,0,0,0,0,0,0,0,0]
Max = []
for row in Hoja1.iter_rows():
    espacio = 'H' + str(i)
    i = i + 1
    if (row[4].value == 'NewColumn'):
        Hoja1['H1'] = 'OptIndex'

    elif (row[4].value == 1):
        Max.append(prev_row[6])
        prev_row[6] = row[6].value
        Hoja1[espacio] = row[6].value

    else:
        if (prev_row[6] < row[6].value):
            prev_row[6] = row[6].value
            Hoja1[espacio] = row[6].value
        else:
            Hoja1 [espacio] = prev_row[6]
```

```
i = 1
j = 0
x = 1
for row in Hoja1.iter_rows():
    espacio = 'T' + str(i)
    i = i+1
    if row[4].value == 'NewColumn':
        Hoja1[espacio] = 'Opt Index2'

    elif row[4].value == 1:
        j = j+1
        Hoja1[espacio] = 0
        x = 0

    else:
        if x == 0:
            if row[6].value == Max[j]:
                Hoja1[espacio] = 1
                x = 1
            else:
                Hoja1[espacio] = 0
        else:
            Hoja1[espacio] = 0
```

Anexo 6: Valor óptimo

```
i = 1
for row in Hoja1.iter_rows():
    espacio = 'J' + str(i)
    i = i + 1
    if row[8].value == 'Opt Index2':
        Hoja1[espacio] = 'Opt Value'

    elif row[8].value == 1:
        Hoja1[espacio] = row[6].value - num.get()

    else:
        Hoja1[espacio] = 0
```

Anexo 7: Piso optimo

```
i = 1
for row in Hoja1.iter_rows():
    espacio = 'K' + str(i)
    i = i + 1
    if row[8].value == 'Opt Index2':
        Hoja1[espacio] = 'Opt Floor'

    elif row[8].value == 1:
        Hoja1[espacio] = row[2].value

    else:
        Hoja1[espacio] = 0
```

Anexo 8: Filtro de Niveles

```
i = 1
prev_row = [0]
for row in Hoja1.iter_rows():
    espacio = 'L' + str(i)
    i = i + 1
    if row[8].value == 'Opt Index2':
        Hoja1[espacio] = 'Sumatoria'

    elif row[4].value == 0:
        if prev_row[0] + row[5].value == 0:
            Hoja1[espacio] = 0
        else:
            Hoja1[espacio] = prev_row[0] + row[3].value
            prev_row[0] = Hoja1[espacio].value

    else:
        Hoja1[espacio] = 0
        prev_row[0] = 0
```

```
i = 1
for row in Hoja1.iter_rows():
    espacio = 'M' + str(i)
    i = i + 1
    if row[8].value == 'Opt Index2':
        Hoja1[espacio] = 'Val Col'

    elif row[11].value < num.get():
        Hoja1[espacio] = 0

    else:
        Hoja1[espacio] = row[11].value - num.get()
```

Anexo 9: Botones

```
def Boton3D():
    x=[]
    y=[]
    z=[]
    for row in Hoja1.iter_rows():
        if row[8].value == 1:
            x.append(row[0].value)
            y.append(row[1].value)
            z.append(row[2].value)

    fig = plt.figure()
    ax1 = fig.add_subplot(111, projection='3d')

    ax1.scatter(x, y, z, c='b', marker='.')

    ax1.set_xlabel ("Este")
    ax1.set_ylabel ("Norte")
    ax1.set_zlabel ("Cota")
    ax1.set_title ("Vista 3D de piso óptimo por columna", fontsize =14, fontweight = "bold")

    plt.show()
```

```
def BotonModelo():
    x=[]
    y=[]
    z=[]
    i = 1
    for row in Hoja1.iter_rows():
        if row[0].value == "Este":
            i = i + 1
        elif row[3].value > 0:
            x.append(row[0].value)
            y.append(row[1].value)
            z.append(row[2].value)

    fig = plt.figure()
    ax1 = fig.add_subplot(111, projection='3d')
    ax1.set_xlabel ("Este")
    ax1.set_ylabel ("Norte")
    ax1.set_zlabel ("Cota")
    ax1.set_title ("Vista 3D del modelo VB > 0", fontsize =14, fontweight = "bold")
    ax1.scatter(x, y, z, c='b', marker='s')
    plt.show()
```

```

def BotonPiso():
    x=[]
    y=[]
    z=[]
    for row in Hoja1.iter_rows():
        if row[8].value == 1:
            x.append(row[0].value)
            y.append(row[1].value)
            z.append(row[2].value)
    VistaCalor=Footprint.create_sheet('Mapadecolor')
    i = 2
    VistaCalor['A1'] = "Este"
    VistaCalor['B1'] = "Norte"
    VistaCalor['C1'] = "Cota"
    for x in x:
        espacio = 'A' + str(i)
        VistaCalor[espacio]= x
        i = i+1
    i = 2
    for x in y:
        espacio = 'B' + str(i)
        VistaCalor[espacio]= x
        i = i+1
    i = 2
    for x in z:
        espacio = 'C' + str(i)
        VistaCalor[espacio]= x
        i = i+1

```

```

Footprint.save("Optimizador.xlsx")
df = pd.read_excel("Optimizador.xlsx", sheet_name = "Mapadecolor")
symbol = ((np.asarray(df['Este'])))
perchange = ((np.asarray(df['Norte'])))
result = df.pivot(index='Norte',columns='Este',values='Cota')
labels = (np.asarray(["{0} \n {1:.2f}".format(symb,value)
                    for symb, value in zip(symbol.flatten(),
                    perchange.flatten())]))

fig, ax = plt.subplots(figsize=(13,7))
title = "Vista planta de pisos óptimos por columna"
plt.title(title,fontsize=18)
ttl = ax.title
ttl.set_position([0.5,1.05])

ax.set_xticks([])
ax.set_yticks([])
sns.heatmap(result, annot = False, fmt="", cmap='RdYlGn',linewidths=0.30,ax=ax)
plt.show()

```

```

def Boton1(plot1):

    WB = openpyxl.load_workbook('Optimizador.xlsx')
    x=[]
    y=[]
    i = 1
    for row in Hoja2.iter_rows():
        if row[0].value == "Cota":
            i = i+1
        else:
            x.append(row[0].value)
            y.append(row[1].value)
    plot1.clear()
    plot1.plot(x,y, "b")
    plot1.set_ylabel('Beneficio [US$]')
    plot1.set_xlabel('Piso')
    plot1.set_title('Beneficio vs piso', fontsize=15)
    canvas.draw()
    canvas.get_tk_widget().pack()
    toolbar = NavigationToolbar2Tk(canvas, miFrame)
    toolbar.update()
    canvas.get_tk_widget().pack()

```

```

def Boton2(plot1):
    df = pd.read_excel(workbook1,sheet_name = 'Multinivel1')

    plot1.clear()

    WB = openpyxl.load_workbook('Optimizador.xlsx')
    x=[]
    y=[]
    i = 1
    for row in Multinivel1.iter_rows():
        if row[0].value == "Cota":
            i = i+1
        else:
            x.append(row[0].value)
            y.append(row[1].value)
    plot1.plot(x,y,"r")
    plot1.set_ylabel('Beneficio [US$]')
    plot1.set_xlabel('Piso')
    plot1.set_title('Beneficio vs piso', fontsize=15)
    canvas.draw()

```

```

def Boton3(plot1):
    df = pd.read_excel(workbook1,sheet_name = 'Multinivel2')
    plot1.clear()
    WB = openpyxl.load_workbook('Optimizador.xlsx')
    x = []
    y = []
    i = 1
    for row in Multinivel2.iter_rows():
        if row[0].value == "Cota":
            i = i+1
        else:
            x.append(row[0].value)
            y.append(row[1].value)

    plot1.plot(x,y,"g")
    plot1.set_ylabel('Beneficio [US$]')
    plot1.set_xlabel('Piso')

    plot1.set_title('Beneficio vs piso', fontsize=15)
    canvas.draw()

```

Anexo 10: Módulos a importar

```

import pandas as pd
import matplotlib.pyplot as plt
import openpyxl
from functools import partial
from tkinter import *
from mpl_toolkits.mplot3d import axes3d
import matplotlib
from matplotlib.pylab import mpl
from matplotlib.figure import Figure
from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg,NavigationToolbar2Tk
import os
from mpl_toolkits.mplot3d import axes3d
import numpy as np
import seaborn as sns

```

Anexo 11: Interfaz gráfica

```
workbook1 = "Optimizador.xlsx"

window = Tk()
window.title('Optimizador de pisos óptimos')
window.geometry("1000x500")
window.iconbitmap("icono.ico")
window.resizable(0,0)
img = PhotoImage(file="USM.png")
lbl_img = Label(window, image = img).place(x= 870, y = 10)

miFrame = Frame(window, width = 550 , height= 450)
miFrame.config(bg= "White")
miFrame.pack(expand=50)

fig = Figure(figsize = (8, 5), dpi = 90)
plot1 = fig.add_subplot()
canvas = FigureCanvasTkAgg(fig, master = miFrame)
```

```
botonModelo = Button(master = window, height = 4, width = 10, command = BotonModelo, text = "Modelo").
place(x= 50 , y= 100 )
boton3D = Button(master = window, height = 4, width = 10, text = "3D", command = Boton3D). place(x= 50 ,
y= 200 )
botonPiso = Button(master = window, height = 4, width = 10, command = BotonPiso, text = "Piso óptimo").
place(x= 50 , y= 300 )
boton1 = Button(master = window,
                command = partial(Boton1, plot1),
                height = 2,
                width = 10,
                text = "Gráfico").place(x= 870 , y= 200 )
boton2 = Button(master = window,
                command = partial(Boton2, plot1),
                height = 2,
                width = 10,
                text = "Gráfico 1").place(x= 870 , y= 300 )
boton3 = Button(master = window,
                command = partial(Boton3, plot1),
                height = 2,
                width = 10,
                text = "Gráfico 2"). place(x= 870 , y= 350 )
window.mainloop()
```

Anexo 11: Cálculos del Modelo 1 en Excel

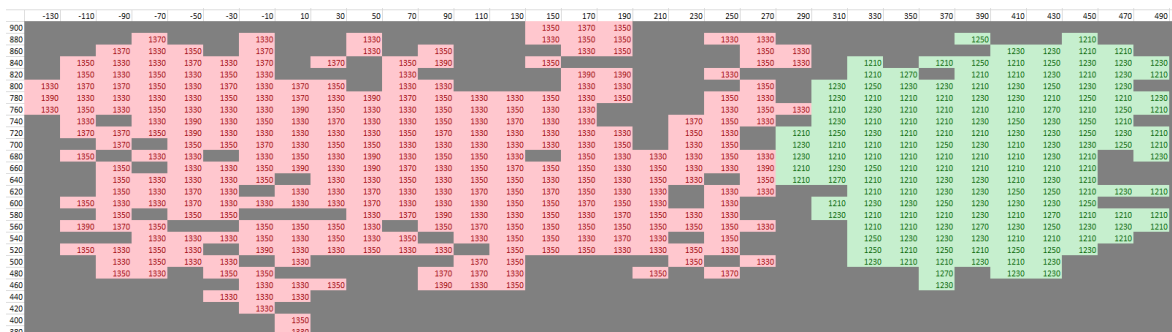


Figura 36: Piso óptimo obtenido a través de Excel.

Tabla 6: Resultados obtenidos al calcular piso óptimo a través de Excel.

Cota	Beneficio [US\$]
1170	1,014,136
1190	1,051,845
1210	1,089,779
1230	1,113,005
1250	1,128,125
1270	1,124,119
1290	1,053,037
1310	990,427
1330	941,411
1350	891,627
1370	854,388
1390	794,465
1410	607,576
1430	429,098
1450	258,529
1470	113,759
1490	27,596
1510	612
1530	0
1550	0
1570	0
1590	0
1610	0
1630	0
1650	0

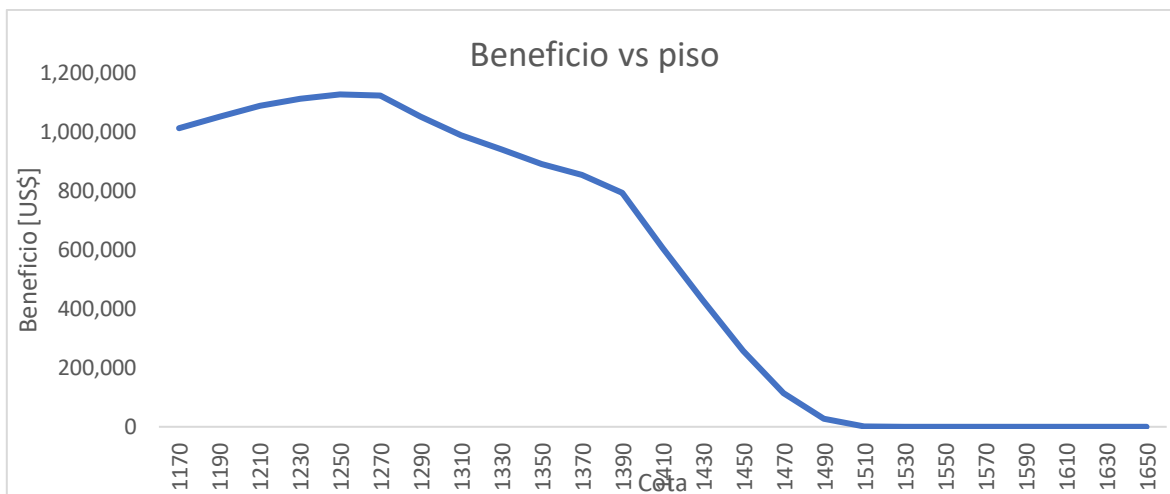


Gráfico 1: Beneficio vs piso del Modelo 1 en un nivel único.

Tabla 7: Resultados obtenidos al calcular el piso óptimo a través de Excel.

Cota	Beneficio [US\$]	Cota	Beneficio [US\$]
1170	677,583	1170	336,553
1190	702,279	1190	349,566
1210	727,150	1210	362,629
1230	752,096	1230	360,909
1250	777,077	1250	351,049
1270	802,102	1270	322,018
1290	827,131	1290	225,906
1310	852,233	1310	138,194
1330	877,335	1330	64,076
1350	874,869	1350	16,757
1370	853,748	1370	640
1390	794,465	1390	0
1410	607,576	1410	0
1430	429,098	1430	0
1450	258,529	1450	0
1470	113,759	1470	0
1490	27,596	1490	0
1510	612	1510	0
1530	0	1530	0
1550	0	1550	0
1570	0	1570	0
1590	0	1590	0
1610	0	1610	0
1630	0	1630	0
1650	0	1650	0

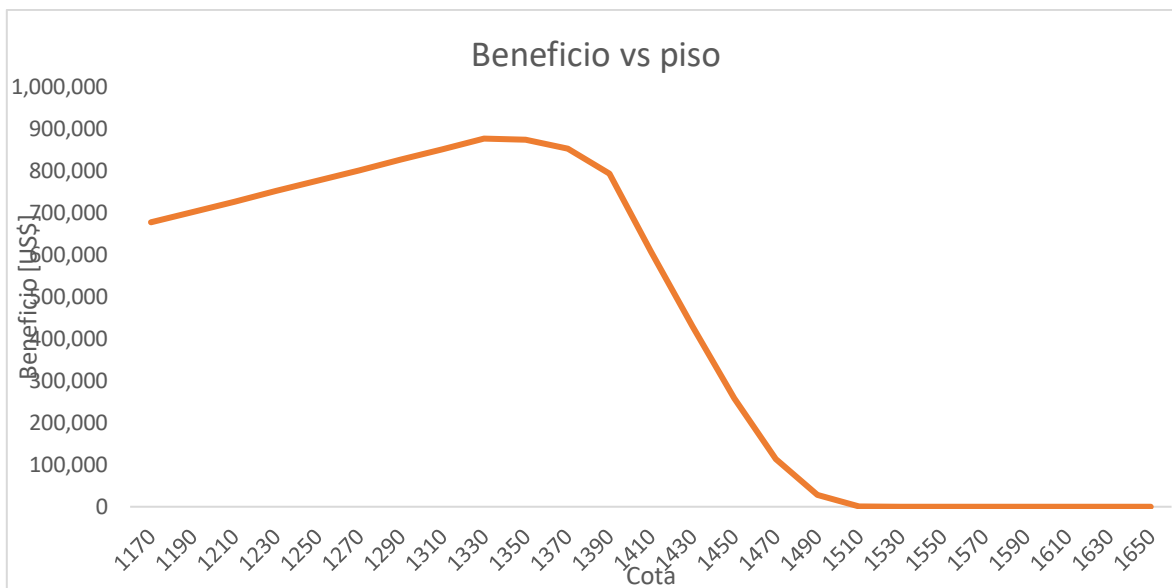


Gráfico 2: Beneficio vs piso del Modelo 1 en un Multinivel.

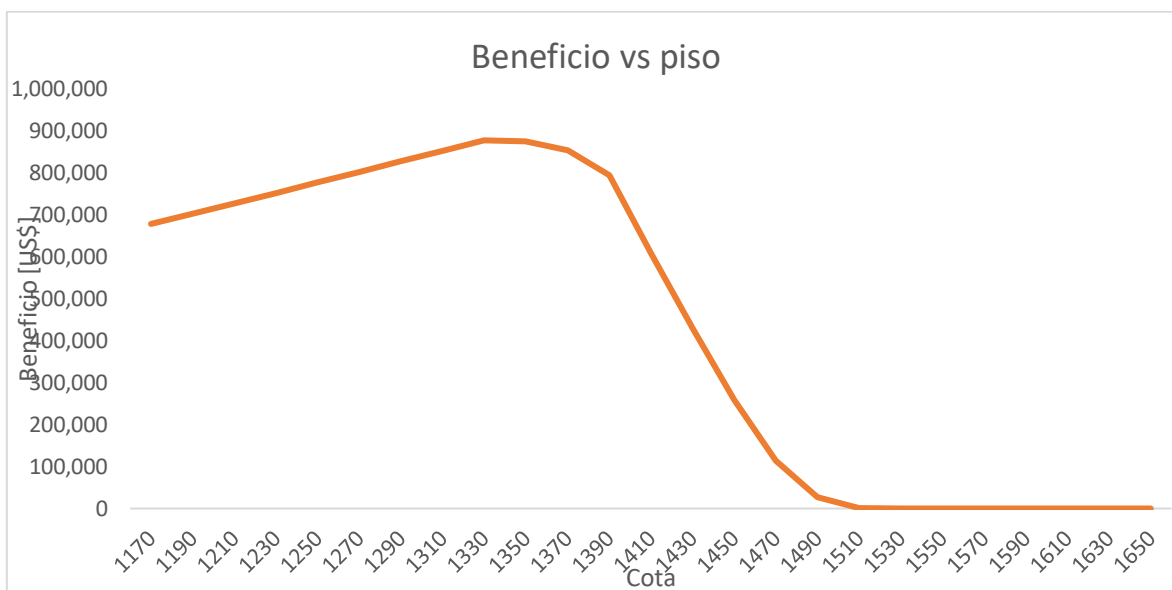


Gráfico 3: Beneficio vs piso del Modelo 1 en un Multinivel.