



UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE MINERÍA

OPTIMIZACIÓN EN LA ASIGNACIÓN DE ESTACIONAMIENTOS DE LOS
CAMIONES DE EXTRACCIÓN PARA EL CAMBIO DE TURNO MEDIANTE LA
UTILIZACIÓN DE REDES NEURONALES Y ÁRBOLES DE DECISIÓN

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL DE MINAS

DIEGO IGNACIO URRUTIA GOLDSACK

PROFESOR GUÍA:

MANUEL ALEJANDRO SOTO OLMEDO

MIEMBRO DE LA COMISIÓN:

RODRIGO DAVID MONTAÑO FUENZALIDA

CARLOS EDUARDO ALVARADO PEDREROS

SANTIAGO DE CHILE

2017

**RESUMEN DE LA MEMORIA PARA OPTAR
AL TÍTULO DE:** Ingeniero Civil de Minas
POR: Diego Ignacio Urrutia Goldsack
FECHA: 12 de Abril, 2017
PROFESOR GUÍA: Manuel Alejandro Soto Olmedo

OPTIMIZACIÓN EN LA ASIGNACIÓN DE ESTACIONAMIENTOS DE LOS CAMIONES DE EXTRACCIÓN PARA EL CAMBIO DE TURNO MEDIANTE LA UTILIZACIÓN DE REDES NEURONALES Y ÁRBOLES DE DECISIÓN

El desafío de disminuir los costos de las operaciones mineras es un tema de interés dentro de la industria, el cual cobra especial relevancia estos últimos años debido a la constante baja del precio del cobre, junto con la disminución continua de las leyes de los yacimientos cupríferos y la escasa probabilidad de encontrar nuevos yacimientos de dimensiones significativas. En consecuencia, actualmente el trabajo no solo se debe centrar en la disminución de los costos, sino también en optimizar diversos procesos con el fin de lograr una mejor productividad y eficiencia.

El siguiente trabajo se centró en disminuir los tiempos perdidos asociados a las demoras en los cambios de turnos existentes en la minería a cielo abierto, mediante la creación de nuevas herramientas de gestión que apuntan hacia una producción continua. Lo anterior se llevó a cabo a través del uso de redes neuronales y un árbol de optimización, el cual basa su criterio en el rendimiento personal de cada operador. De esta forma, se obtuvieron y cuantificaron los resultados a través de simulaciones computacionales en donde se comparó la productividad actual de una faena minera versus la productividad utilizando el modelo planteado en este trabajo.

Los resultados obtenidos aplicando el modelo de optimización arrojaron que se logró disminuir en un 9.8% las demoras asociadas a los cambios de turno, lo que es equivalente a 398 minutos y a 24 descargas extras por turno, tomando en cuenta que el caso de estudio se realizó para una flota de 130 camiones de extracción. Lo anterior, en términos económicos se traduce en 5.7 MUSD anuales, considerando que solamente se ahorraran los costos asociados al transporte de mineral, suponiendo un costo de transporte igual a 1 dólar por tonelada y una carga efectiva de 330 toneladas por camión.

Al finalizar el trabajo, se observa que la posibilidad de mejorar el proceso de asignación de estacionamientos es completamente factible, al igual que mejorar el modelo planteado incluyendo otras condiciones de borde que puedan ayudar a realizar una simulación más realista, pero que deben ser personalizadas acorde a la manera de operar de cada faena. Por otro lado, se detectó que para las mismas condiciones de trabajo, existen operadores que tienen rendimientos completamente distintos, por lo que se recomienda realizar un análisis de velocidades por trabajador, en donde se comparen sus rendimientos individuales con los rendimientos que debiesen alcanzar las máquinas según su catálogo.

Por último, se concluye que el uso de redes neuronales como herramienta de gestión genera un valor agregado en la toma de decisiones que estén relacionadas a la formación de grupos de trabajo, además de identificar características que ayuden a entender y manejar el comportamiento de ciertas variables tales como la productividad o utilización dentro de una operación minera.

**ABSTRACT OF THE SUBMITTED THESIS TO
OBTAIN THE DEGREE OF: Mining Engineer
BY: Diego Ignacio Urrutia Goldsack
DATE: 12th April, 2017
THESIS ADVISOR: Manuel A. Soto Olmedo**

**OPTIMIZATION IN THE ALLOCATION OF PARKING OF THE EXTRACTION
TRUCKS FOR CHANGE OF SHIFT THROUGH THE USE OF NEURONAL
NETWORKS AND DECISION TREES**

The challenge of decreasing the operational cost of the mining operations it has been always an issue of interest to the industry. This defiance has taken a special importance in the last years because of the constant price decline of the base metals, along with the continuous decrease of the law of the ore deposits and the poor chance of finding new giants deposits for doing mining in a big scale. Because of this, nowadays the work must not center only in taking down the cost of the process, but also to optimize existing processes in order to have a better productivity and efficiency.

The following work was focused on reducing the lost time associated to the delays in shift changes that take place in the open pit mining, in addition to providing new management tools to reach a more continuous production. The above, it was carried out through neural networks and a decision tree which bases its decision –making on the personal performance of each worker, obtaining and quantifying the results through computer simulations, which were carried out with the data facilitated by a mine located at the north of Chile.

The results obtained applying the developed model of optimization showed that it was possible to reduce delays of the shift change by 9%, which is equivalent to 389 minutes and 24 more discharges by shift for the case of a float of 130 mining trucks. This is traduced in 5.7 MUSD per year, considering that the only cost that is saved is the one associated to transport the ore, assuming a cost of transportation of 1 dollar per ton.

One of the conclusion of the memory is that the possibility of improving the whole process of parking assignations to the mining trucks its completely feasible, as well as improving the model proposed through considering more border conditions that can help to reach a more realistic simulation. By the other hand, it was detected that for the same work conditions, it exists some workers that have a notorious better performance that others in terms of the velocity that they reach, so it's believed that it should be done an analysis per worker, comparing the individuals performance of each operator with the performance that the truck should reach according to the factory catalog.

Finally, its concluded that the use of neuronal networks as a tool of management it can be of great utility for supporting the decisions that are related to the formation of the groups that will work together every year, in addition to identifying the characteristics that help to understand and handle the behavior of some variables such as the productivity of a mining work.

DEDICATORIA

La siguiente memoria se la quiero dedicar a mis amigos, familia y polola que me apoyaron durante este largo camino. Ustedes fueron los que rieron conmigo en los buenos momentos, y se quedaron conmigo en los momentos que lo ameritaban, y eso lo agradeceré siempre.

También quisiera dedicarle este trabajo al club de mis amores, el cual tiene los valores de amistad, respeto, esfuerzo, paciencia y perseverancia que vivo día a día, y que junto con mis camaradas, me enseñan a ver la vida de una forma completamente distinta, completamente azul.

Por último, quisiera dedicarles este trabajo a mis dos abuelos que ya no me acompañan físicamente, que sé con toda certeza que estarían infinitamente orgullosos de este logro.

AGRADECIMIENTOS

A mis viejos por darme la oportunidad de estudiar sin complicación ni exigencia alguna, por guiarme cuando estaba perdido y felicitarme cuando sentían que lo estaba haciendo bien. Por apoyarme en todo momento y de la manera más incondicional que me ha tocado vivir. Gracias por bancarme sin cuestionamiento alguno, por los buenos deseos que me mandaban en cada prueba u examen que rendía, y por celebrar mis logros como si fueran suyos (y así lo son).

A mi polola por soportarme en los momentos que me cuestionaba todo, por su increíble incondicionalidad y por su amor verdadero. Definitivamente este recorrido hubiese sido muy distinto sin ti. También quiero agradecer a su familia por celebrar conmigo mis logros y preocuparse siempre de mi buen pasar. Fran, eres un regalo de la vida y te estaré siempre agradecido por acompañarme en esta etapa.

A mis amigos del estadio, que siempre me acompañan en la cancha y comparten conmigo ese mágico momento en donde durante 90 minutos, solo importa una sola cosa. Gracias por acompañarme en mi locura, vivirla y sobre todo por los momentos y viajes que hemos compartido,

A mis amigos franco, rai, morla, nakio y piter, simplemente por estar siempre, son lo más.

A la universidad por brindarme tres grandes amigos como el dolape, coto y el mati, ya que personas así no se encuentran todos los días.

Por último, quiero agradecerle a mi profesor y amigo personal el sr. Manuel Soto Olmedo, por creer en mí desde un principio, brindarme amistad, confianza, transparencia, y por ofrecerme mi primera oportunidad de trabajo como profesional.

Tabla de Contenido

CAPITULO 1 INTRODUCCIÓN	1
1.1 Motivación del trabajo	2
1.2 Objetivos	3
1.2.1 Objetivo General.....	3
1.2.2 Objetivos Específicos	3
1.3 Alcances	3
1.4 Plan de trabajo	4
CAPITULO 2 ANTECEDENTES	5
2.1 Gestión del tiempo	5
2.2 Redes Neuronales	6
2.3 Cambio de turno en operaciones mineras	10
2.4 Asignación de estacionamientos actual	10
2.5 IDW	12
2.6 Simulación de Inundaciones	12
2.7 Largo real vs Largo aparente	13
2.8 Desviación estándar relativa	13
CAPITULO 3 METODOLOGÍA.....	14
3.1 Determinación de variables significativas	14
3.2 Obtención de datos.....	14
3.3 Aplicación del modelo de redes neuronales	15
3.4 Construcción del modelo de optimización.....	15
3.5 Simulación de escenarios.....	16
3.6 Conclusiones y recomendaciones	16
CAPITULO 4 RECOLECCIÓN DE DATOS.....	17
CAPITULO 5 RESULTADOS Y ANÁLISIS	18
5.1 Obtención de los datos significativos	18
5.2 Velocidades de los operadores.....	21
5.3 Redes neuronales	22
5.4 Tiempos perdidos.....	25
CAPITULO 6 CONCLUSIONES Y RECOMENDACIONES	27
6.1 Conclusiones.....	27
6.2 Recomendaciones	28
CAPITULO 7 BIBLIOGRFÍA	29

CAPITULO 8 ANEXOS	30
8.1 Anexo A: Contexto Empresa	30
8.1.1 Visión	30
8.1.2 Misión.....	30
8.1.3 Sectores de negocio.....	31
8.2 Anexo B: Código Memoria.....	34

ÍNDICE DE ECUACIONES

Ecuación 1: Función objetivo K-mean.	9
Ecuación 2: Cálculo de la distancia euclidiana.	9
Ecuación 3: Cálculo centroides, algoritmo k-mean.....	9
Ecuación 4: Método Ponderación Inversa.	12
Ecuación 5: Cálculo <i>SEW</i> y <i>SNS</i>	13
Ecuación 6: Cálculo pendiente por celda.	13
Ecuación 7: Desviación estándar relativa.	13

ÍNDICE DE TABLAS

Tabla 1: Datos entregados por la minera colaboradora.	17
Tabla 2: Velocidades promedio de la flota completa.	20
Tabla 3: Velocidades de cada camión en km/hr.:	20
Tabla 4: Velocidades de cada camión cargado en km/hr.	20
Tabla 5: Velocidades de cada camión descargado en km/hr.	20
Tabla 6: Velocidades promedio de la flota completa diferenciando por estado de carga.	21
Tabla 7: Desviación estándar relativa de las velocidades.....	21
Tabla 8: Centroides grupos redes neuronales.	22
Tabla 9: Demoras cambio de turno aplicando el proceso de optimización.	25
Tabla 10: Demoras cambio de turno sin aplicar proceso de optimización.	25
Tabla 11: Ranking de empresas líderes en tecnología minera.....	33

ÍNDICE DE ILUTRACIONES

Ilustración 1: Impacto del cambio de turno para en la productividad de una faena.	1
Ilustración 2: Productividad mejorada por hora.	2
Ilustración 3: Norma ASARCO.....	5
Ilustración 4: Neurona, unidad base de las redes neuronales.	6
Ilustración 5: Redes neuronales.....	7
Ilustración 6: Error vs Numero de muestras utilizadas.....	8
Ilustración 7: Error vs Numero de Hidden Layers.	8
Ilustración 8: Estacionamientos por pala.....	10
Ilustración 9: Búsqueda radial, IDW.	12
Ilustración 10: Matriz ejemplo.	12
Ilustración 11: Ejemplo simulación por inundación.....	13
Ilustración 12: Metodología.....	14
Ilustración 13: Árbol de decisión modelo de optimización.....	16
Ilustración 14: Interpolación global de la mina.	18
Ilustración 15: Interpolación rajo A.....	19
Ilustración 16: Interpolación rajo B.....	19
Ilustración 17: Gráfico minutos restantes v/s Velocidad Ascenso.	22
Ilustración 18: Gráfico minutos restantes v/s Velocidad plano.....	23
Ilustración 19: Gráfico minutos restantes v/s Velocidad descenso.	23
Ilustración 20: Gráfico minutos restantes v/s Velocidad descenso.	24
Ilustración 21: Histograma demoras cambio de turno sin aplicación del modelo de optimización.	25
Ilustración 22: Histograma demoras cambio de turno con el modelo de optimización.	25
Ilustración 23: Enfoue principal de Hexagon	30
Ilustración 24: Sectores de Negocios – Hexagon (Fuente: Presentación Corporativa).....	31
Ilustración 25: Ventas y márgenes anuales de Hexagon (Fuente: Presentación corporativa). 32	
Ilustración 26: Ingresos por área de negocio - Hexagon (Fuente: Presentación Corporativa). 32	

CAPITULO 1 INTRODUCCIÓN

Hoy en día la gran minería utiliza extensas y variadas flotas de equipos de alta capacidad para aprovechar la economía de escala presente en este negocio. En esta materia, tener un alto número de unidades de acarreo a disposición tiene como desafío operacional y financiero la optimización en el uso de estos recursos por parte de las empresas. Lo anterior, ha generado la implementación de eficientes sistemas de asignación de destinos los cuales centran sus esfuerzos en gestionar oportunamente la ubicación, hora y estado de ocupación en que se encuentre cada equipo, al momento de asignar una nueva ruta.

Un desafío pendiente del modelo es la baja significativa en la productividad de las operaciones mineras al principio y al final de cada jornada lo que se puede explicar en la pérdida de tiempo operacional que actualmente generan estos cambios. A modo de ejemplo, se muestra un gráfico de productividad por hora de una faena tipo, donde se observa claramente la baja de producción en las instancias mencionadas y en color rojo la pérdida de movimiento mina asociada, con el respectivo impacto negativo en la eficiencia operacional y rentabilidad de la empresa.



Ilustración 1: Impacto del cambio de turno para en la productividad de una faena.

Esto se debe principalmente a la administración de los camiones, ya que son estos los que permiten mover tanto el mineral como el estéril desde el interior de la mina hasta el exterior, donde cada detención genera inevitablemente una disminución de la producción. Es por esto que los cambios de turno ofrecen una gran oportunidad de mejora en la utilización de los equipos de transporte, en el rendimiento de los operadores y consecuentemente en la rentabilidad de la empresa.

Actualmente cuando se acerca la hora del cambio de turno, a todos los camiones se les asigna un estacionamiento a través del sistema de despacho, aunque existen casos en que los operadores no respetan esta asignación ante una eventual pérdida del bus que realiza el cambio de operador. Más aun, la asignación proveniente de despacho no optimiza el tonelaje que los equipos transportan. Finalmente, el hecho de que no se respeten los estacionamientos asignados, afecta directamente el rendimiento de los operadores del turno entrante, ya que al no encontrar el camión en el estacionamiento que corresponde, pierden tiempo en el traslado que se requiere para reubicarlo.

1.1 Motivación del trabajo

El objetivo de esta memoria de título es determinar y proponer una solución a la brecha de eficiencia planteada. Lo anterior, definiendo e implementando un sistema que asigne el estacionamiento más apropiado (en términos de productividad) a los equipos basándose en distintas variables, permitiendo una alimentación más continua a la planta de chancado y maximizando el uso de esta. Las variables para la construcción del modelo propuesto, son las velocidades del camión en régimen durante el turno, diferenciando distintos estados del equipo, que junto con su ubicación GPS 40 minutos antes de finalizar el turno permita asignar un estacionamiento a cada camión tal que maximice su tiempo de utilización, optimizando el recorrido productivo sobre el destino final e inicial de cada unidad de acarreo entre turnos.

Los anteriores parámetros serán utilizados en un modelo de análisis basado en redes neuronales, que permiten que el modelo de asignación construido se base en los grupos de operadores críticos dentro de la operación y en las variables que más afectan al sistema, incluyendo también la cantidad de camiones que deben quedar cargados y vacíos según el protocolo de la faena.

Además, la eficiencia del sistema se cuantificará antes de implementarse mediante una comparación entre una simulación de la operación actual de la minera versus una simulación donde se tenga la asignación mediante el sistema propuesto, considerando las brechas operacionales que existen dentro de una operación, tal como el panne de los camiones.

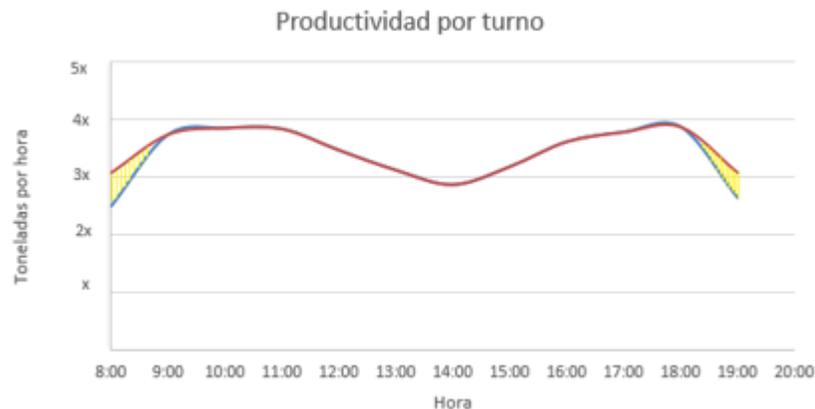


Ilustración 2: Productividad mejorada por hora.

En el análisis y decisiones sobre proyectos, no dejan de ser interesantes los casos de optimización operativa que no requieren inversión de infraestructura, cuya amortización se puede visualizar en el corto plazo mediante ahorro de gastos y aumentos de productividad, incrementando posteriormente y de manera permanente la utilidad neta para la empresa.

1.2 Objetivos

En esta sección se detallan los objetivos del trabajo final a entregar en el curso Trabajo de Título MI6909.

1.2.1 Objetivo General

El objetivo general de este trabajo es crear un modelo de optimización, el cual realice una asignación de estacionamientos para los camiones de extracción en el horario del cambio de turno. El funcionamiento del modelo deberá ser validado mediante algoritmos computacionales que logren simular la operación minera, evaluando el caso donde se designan los estacionamientos por medio del modelo desarrollado a lo largo de este documento, y comparando los resultados con el escenario actual de la operación.

1.2.2 Objetivos Específicos

Los objetivos específicos del trabajo se detallan a continuación:

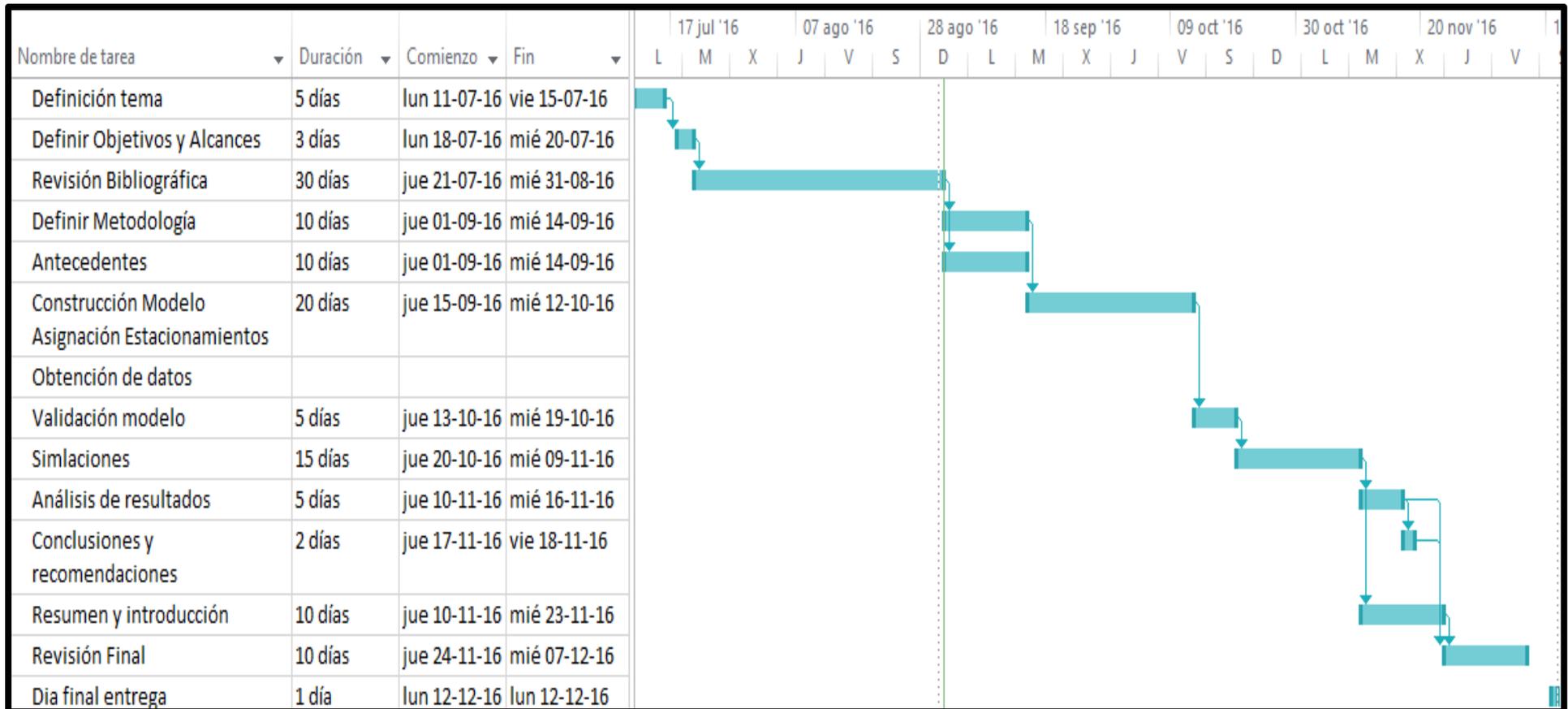
- Desarrollar un modelo de redes neuronales el cual agrupe a los operadores según la velocidad a la que conducen en distintos escenarios, la distancia existente entre su ubicación y el estacionamiento más cercano una hora antes del cambio de turno, y los minutos perdidos al final de cada jornada, con el fin de generar una nueva herramienta de gestión para analizar los distintos grupos de trabajo.
- Desarrollar un árbol de decisión capaz de optimizar el proceso de asignación de estacionamientos de los CAEX.
- Realizar simulaciones computacionales para determinar el impacto de implementar el sistema de asignación desarrollado sin la necesidad de probarlo en terreno.
- Evaluar el impacto del modelo en la disminución de las demoras programadas para el cambio de turno, utilización de los CAEX, y en el rendimiento de sus operadores.

1.3 Alcances

A continuación se detallan los alcances para este proyecto:

- Trabajo desarrollado para la empresa Hexagon | Pixis.
- El trabajo solo considera minería a cielo abierto.
- Para la construcción del modelo se consideran datos de una faena localizada en el norte de Chile.
- Por sistema de transporte solo se consideran los camiones de extracción (CAEX).
- El trabajo se centra en disminuir las demoras programadas del cambio de turno y aumentar la productividad de la operación mediante la aplicación de un modelo de asignación eficiente.
- Los resultados presentados son en base a simulaciones, por lo que para determinar el verdadero impacto del modelo es necesario realizar una prueba en terreno.
- No se evaluarán ni simularán las bajas productivas generadas por el horario de colación, ya que el trabajo se centra en las bajas productivas generadas al final de cada turno.

1.4 Plan de trabajo



CAPITULO 2 ANTECEDENTES

2.1 Gestión del tiempo

Dado que uno de los indicadores con los que se quiere cuantificar el beneficio del nuevo modelo de asignación es el tiempo efectivo, se hace necesario definir como serán clasificados los tiempos dentro de la operación. Para esto, se utilizó la norma ASARCO, la cual es el estándar para la industria minera. Esta norma define los tiempos de la siguiente manera:

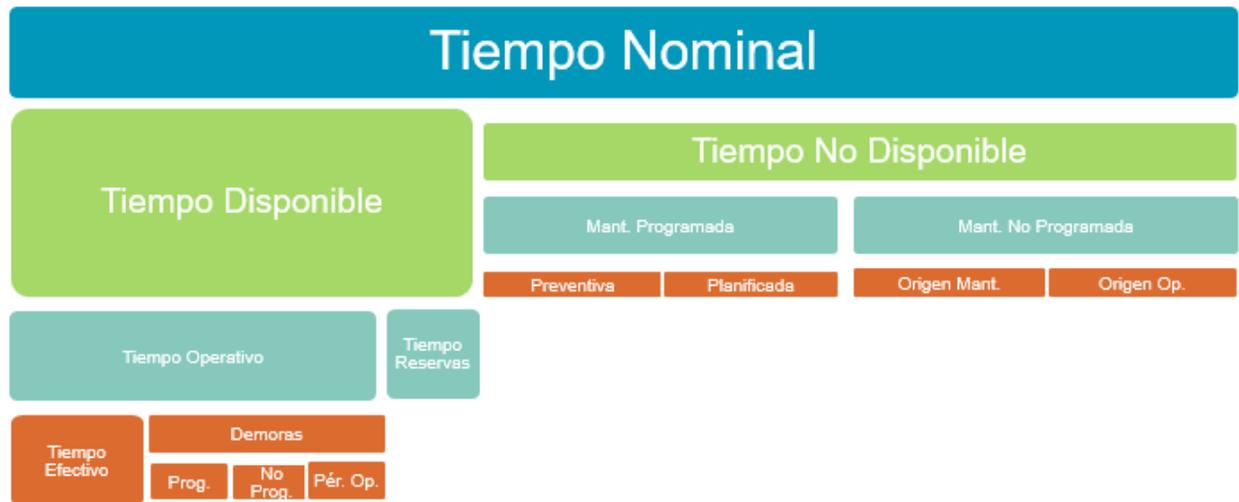


Ilustración 3: Norma ASARCO.

En base a lo anterior, se observa que la utilización efectiva es igual al tiempo efectivo dividido en el tiempo operacional. Es por esto que uno de los objetivos de esta memoria es reducir las demoras con el fin de aumentar el tiempo efectivo de los equipos y así, aumentar la utilización efectiva de los CAEX.

Dentro de las demoras existen 3 tipos, los cuales se detallan a continuación.

1. Demoras programadas
 - a. Colación.
 - b. Cambio de turno.
2. Demoras no programadas
 - a. Espera en petrolera
 - b. Pista obstruida
 - c. Relevos
3. Pérdidas operacionales
 - a. Espera en chancado
 - b. Espera en carguío.

Lo anterior refuerza el objetivo de esta memoria, el cual es reducir las demoras programadas relacionadas exclusivamente al cambio de turno, ya que las colaciones se encuentran fuera del alcance de este trabajo.

2.2 Redes Neuronales

Las redes neuronales son algoritmos para realizar tareas cognitivas, tales como aprender u optimizar procesos, que están basados en conceptos derivados de investigaciones de la naturaleza del cerebro (B. Muller, J. Reinhardt, M.T. Strickland , 1995).

La unidad básica de las redes neuronales son las neuronas, las cuales se definen matemáticamente por los siguientes parámetros:

1. X_i : *Parámetros de entrada*
2. W_i : *Ponderadores*
3. O_i : *Parámetros de salida*
4. $F_{(i)}$: *Función de transferencia*

En donde los parámetros de entrada (inputs), ingresan a la función de transferencia ponderados por distintos factores para entregar un parámetro de salida, tal como se observa en la Ilustración 4.

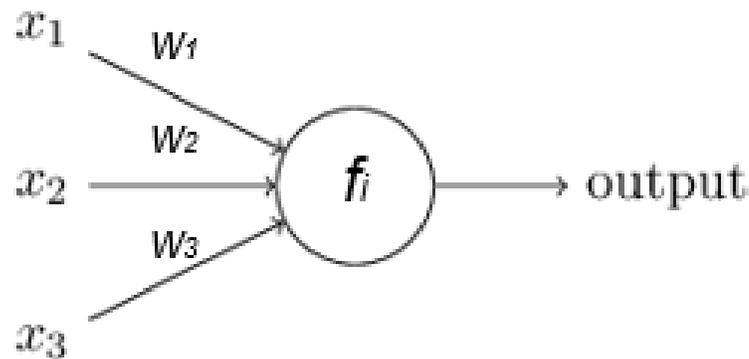


Ilustración 4: Neurona, unidad base de las redes neuronales.

Con esto, se puede construir una red neuronal la cual está compuesta por un conjunto de capas que no son más que neuronas que trabajan en paralelo. Cuando estas capas se encuentran interconectadas entre si forman una red neuronal. Dentro de este conjunto de capas se distinguen 3 tipos:

1. **Input Layer:** Corresponde a la capa que recibe todos los parámetros de entrada de la red neuronal.
2. **Output Layer:** Corresponde a la capa que tiene como salida el output o la data final de la red neuronal.
3. **Hidden Layer:** Corresponden a todas las capas que se sitúan entre la Input y la Output Layer.

Lo anterior se puede apreciar de mejor manera en la Ilustración 5.

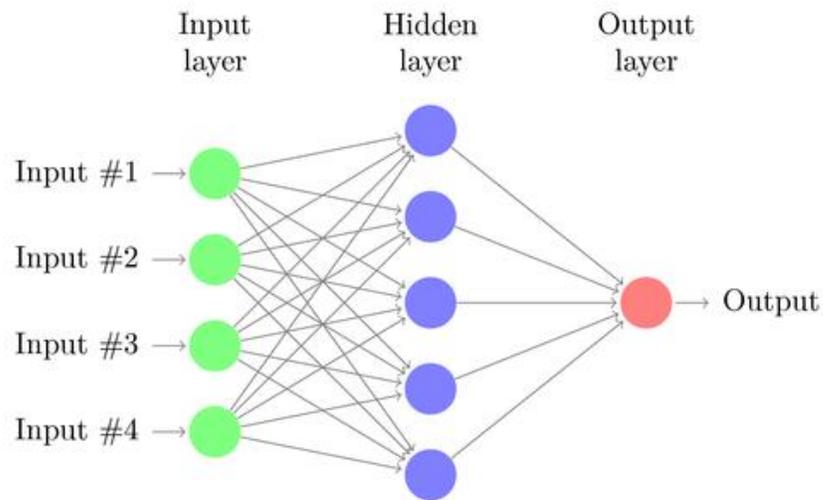


Ilustración 5: Redes neuronales.

A su vez, existen dos tipos de patrones de conexión entre capas, estos son:

1. Feed-forward networks: Este tipo de conexión restringe el flujo de información a una sola dirección, comenzando desde la input hasta la output layer (Dastani, 1991).
2. Recurrent networks: Este tipo de redes permite que los datos fluyan en forma bidireccional, pudiendo devolverse la data procesada en el caso de que lo red lo estime conveniente.

Para que la red funcione y entregue resultados racionales es necesario entrenarla con información coherente y que permita establecer correlaciones. Durante el proceso de entrenamiento, los ponderadores se van ajustando con el fin de minimizar el error y procesar dicha información de manera de minimizar la tasa de error y asegurar la representatividad de la red neuronal. Existen dos tipos de entrenamientos de redes neuronales, estos son:

1. Aprendizaje supervisado: Este tipo de aprendizaje se basa en establecer los outputs que se esperan de la red para los distintos grupos de inputs provistos. De esta manera, la red comienza a variar los ponderadores utilizados para llegar a los resultados establecidos de la forma más precisa posible.
2. Aprendizaje no supervisado: Este tipo de aprendizaje se basa en establecer parámetros que busquen similitudes entre grupos de datos de inputs con el fin de clasificarlos en conjuntos que tengan características similares. Todo esto mediante una un número de iteraciones del proceso de clasificación que se determina antes de empezar el entrenamiento de la red. Cabe destacar que el proceso iterativo se encuentra en la sección de anexos.

Por último, el error se puede manejar mediante dos variables.

1. Numero de muestras utilizadas para entrenar la red.

Mientras mayor sea el volumen de la información utilizada para entrenar la red, esta ajustará los ponderadores de manera más precisa, de tal forma que cuando se procesen nuevos datos el resultado tendrá un menor rango de error asociado. Lo siguiente se observa en la Ilustración 6.

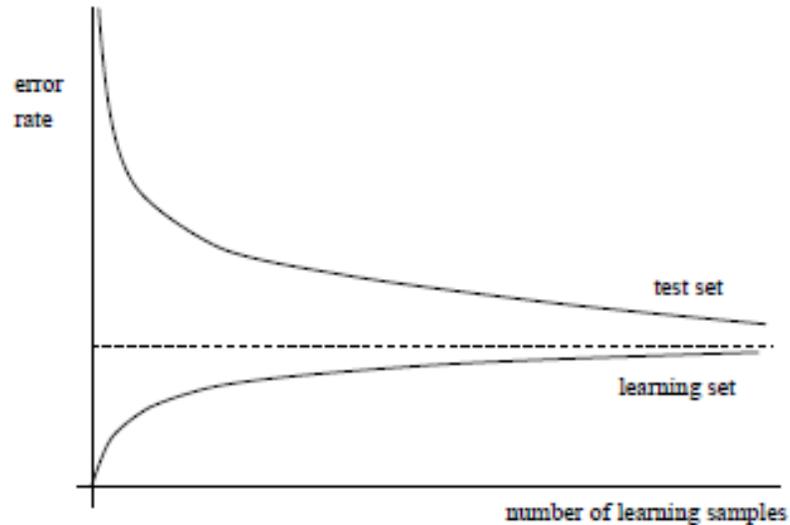


Ilustración 6: Error vs Numero de muestras utilizadas

2. Numero de hidden layers o capas intermedias.

En el caso de las capas intermedias, existe un número óptimo que minimiza el error de la red. Esto, ya que si se utilizan demasiadas hidden layers se sobredimensiona el proceso, haciéndolo no solo más lento si no que menos representativo, afectando su desempeño negativamente. Lo siguiente se muestra en la Ilustración 7.

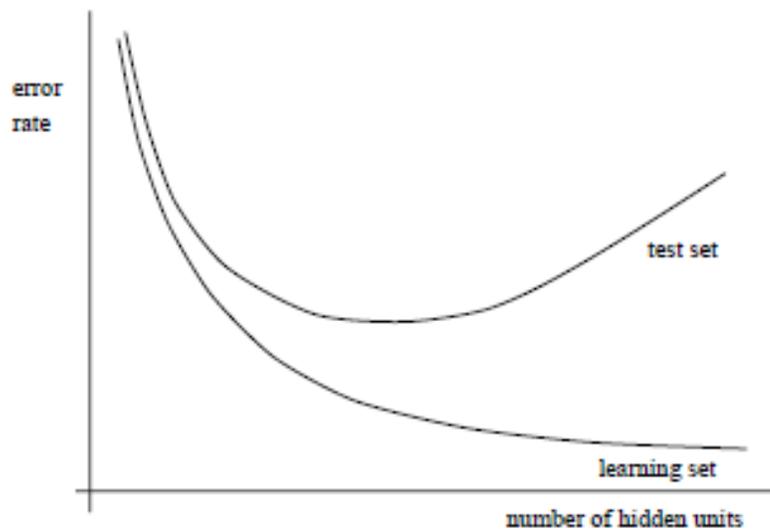


Ilustración 7: Error vs Numero de Hidden Layers.

Existen varios tipos de redes neuronales las cuales son utilizadas según el tipo de tarea que se pretende que la red ejecute. Para este estudio, se utilizó el algoritmo k-mean debido a su simplicidad y a su capacidad de agrupamiento.

2.2.1 Algoritmo k-mean

El método de agrupación k-mean (también conocido como algoritmo de Lloyd's) es uno de los algoritmos iterativos más simples que trabajan bajo la modalidad de aprendizaje no supervisado. El objetivo de este método es minimizar la distancia euclidiana que existe entre cada uno de los datos con el centroide correspondiente al grupo más cercano, lo que se expresa matemáticamente de la siguiente forma.

$$f(x) = \min_c \sum_{i=1}^k \sum_{x \in c_i} d(x, \mu_i)$$

Ecuación 1: Función objetivo K-mean.

Donde

x = Vector del centroide c_i

μ_i = Vector del punto i

$d(x, \mu_i)$ = Distancia euclidiana entre el centroide x y el punto μ_i

Y la distancia euclidiana se calcula como:

$$d = \sqrt{\sum_{j=1}^n (x - \mu_i)^2}$$

Ecuación 2: Cálculo de la distancia euclidiana.

Además, para ejecutar esta metodología se debe tener definido previamente la cantidad k de agrupaciones que se quieren obtener del modelo con sus respectivos centroides ya inicializados, los cuales pueden tomar cualquier valor en una primera instancia. Por último, la iteración k-mean se muestra a continuación.

1. Definir la cantidad de grupos que se quieren obtener del modelo.
2. Inicializar los centroides de todos los grupos.
3. Atribuir cada punto de los datos al centroide más cercano.
4. Redefinir la posición de los centroides mediante la siguiente ecuación:

$$\mu_i = \frac{1}{|c_i|} \sum_{j \in c_i} x_j, \forall i$$

Ecuación 3: Cálculo centroides, algoritmo k-mean.

5. Repetir pasos 3 y 4 hasta alcanzar una convergencia.

2.3 Cambio de turno en operaciones mineras

Dentro de la operación minera existen distintas jornadas de trabajo, las cuales se dividen en dos categorías. La primera se refiere a la cantidad de días que los trabajadores permanecen en el campamento minero, en donde existen variadas modalidades tales como 7x7, 4x3, 6x1 - 6x2 - 6x3, y otros, en donde el primer número indica la cantidad de días que el personal se queda en el campamento minero, y el segundo número indica la cantidad de días de descanso fuera de la faena. La segunda categoría se refiere a la cantidad de horas que se trabajan diariamente. En general este tipo de turnos son de 8 horas, el cual implica dos cambios de turnos en un mismo día, o de 12 horas, en donde se tiene solo 1 cambio de turno diario.

Dicho esto, para efectos de esta memoria se refiere al cambio de turno al cambio de personal que sucede dentro de un mismo día de trabajo, es decir, a la segunda categoría mencionada en el párrafo anterior.

Por último, cabe destacar que la minera que facilitó los datos para llevar a cabo este estudio opera en un turno de 12 horas, por lo que tiene un cambio de turno diario.

2.4 Asignación de estacionamientos actual

Hoy en día la asignación de estacionamientos para los CAEX a la hora del cambio de turno la hace el despachador con apoyo del sistema de despacho. El proceso consiste en asignar los estacionamientos pala a pala, ya que cada una de estas tiene asociada una flota de camiones que están realizando una o múltiples rutas productivas (que inicia en el punto de carguío y termina en el punto de vaciado). Además, cada ruta tiene asociado dos estacionamientos, de los cuales uno se encuentra junto a la pala, y el otro en un punto intermedio de la ruta, existiendo la posibilidad de que el camión quede vacío o cargado. A continuación se ilustra lo anteriormente mencionado.

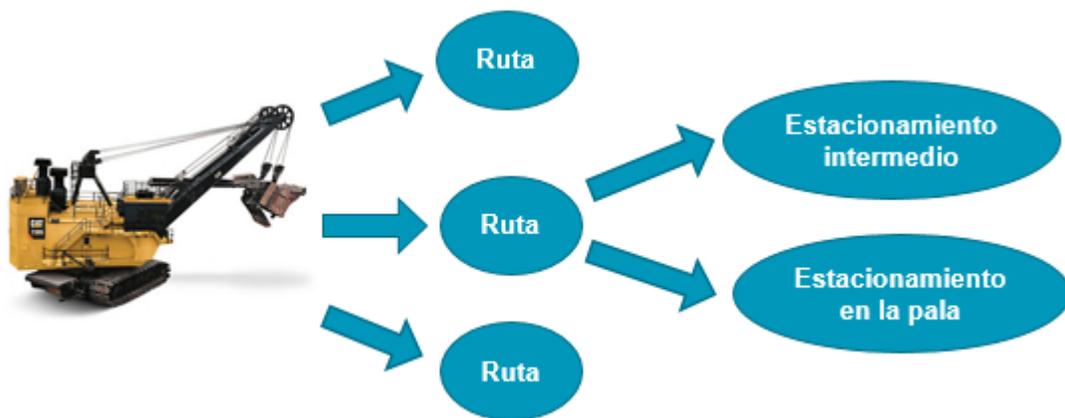


Ilustración 8: Estacionamientos por pala.

En base a esto, una hora antes del cambio de turno el despachador comienza a dar la instrucción a los camiones de quedarse o seguir hacia el próximo estacionamiento en base a los resultados que muestra Dispatch® gracias a los algoritmos que utiliza el sistema. Estos algoritmos son los siguientes:

1. Mejor Ruta (MR – Best Path)
 - i. Se basa en determinar la ruta que tiene el menor tiempo de viaje entre nodos
 - ii. Entrega:
 1. Distancia mínima total.
 2. Estimación del tiempo de viaje.
 3. Puntos intermedios de viaje (nodos).
2. Programación lineal (PL – Lineal Programming)
 - i. Resuelve un conjunto de restricciones de ecuaciones lineales, sujeto a una función de prioridades y exigencias.
 - ii. Minimiza el total de camiones requeridos para cubrir las necesidades de las palas.
3. Programación Dinámica (PD – Dynamic Programming)
 - i. Cambia los destinos (puntos de carguío) de los camiones constantemente con el fin de maximizar el uso de estos.

Los problemas que genera este tipo de asignación son variados y repercuten en una baja considerable en la productividad durante estos horarios. Los problemas detectados se detallan a continuación:

1. El operador decide quedarse en el estacionamiento minutos antes de que termine el turno ya que teme a perder la van que los lleva de regreso al campamento. Cuando esto sucede, existe una gran cantidad de ocasiones en que el despachador le dice al operador que siga hacia el siguiente estacionamiento, pero se encuentra con la respuesta de que el aumentar la velocidad para subir la productividad de la mina atenta contra la seguridad de los trabajadores, por lo que no están dispuestos a forzar su llegada al siguiente lugar para estacionarse, perdiendo minutos valiosos para la operación.
2. Los operadores del turno entrante muchas veces no encuentran el equipo que se les asignó para el turno. Esto se debe a que la información con la que se realizó dicha asignación se basa en que todos los camiones de extracción quedaron estacionados donde el despachador ordenó que quedaran, lo que dista mucho de la realidad debido a lo explicado en el punto 1. Además se identificaron muchos casos en donde el CAEX estaba estacionado donde tenía que estar, pero minutos antes del cambio de turno el taller mecánico llama a los camiones a sus mantenciones, con lo que la unidad de transporte se traslada del estacionamiento asignado, generando demoras para los operadores del turno entrante.
3. Al acercarse la hora del cambio de turno, algunos operadores comienzan a disminuir su velocidad con el fin de llegar justo a tiempo al estacionamiento, y de esta forma no exponerse a que le asignen otro lugar para detenerse, provocando disminuciones significativas en la productividad de la mina en la última hora de cada turno.

2.5 IDW

El método de ponderación de distancia inversa (IDW por sus siglas en inglés) consiste en realizar una búsqueda radial sobre un punto en específico, y luego ponderar cada valor de los puntos encontrados en la búsqueda de manera inversamente proporcional a la distancia con el punto en cuestión. Tal como se muestra a continuación:

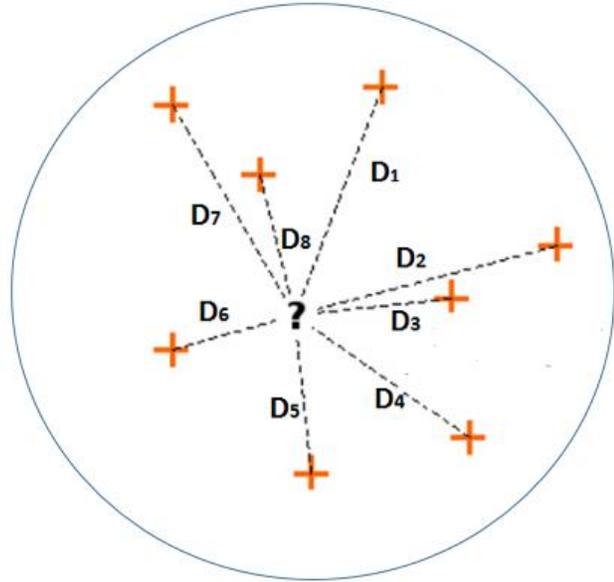


Ilustración 9: Búsqueda radial, IDW.

$$Z_x = \frac{\sum_{i=1}^n Z_i / D_i^2}{\sum_{i=1}^n 1 / D_i^2}$$

Ecuación 4: Método Ponderación Inversa.

2.6 Simulación de Inundaciones

El método de simulación de inundaciones (Slope Neighborhood Algorithm), estima la pendiente (en porcentaje) de una celda perteneciente a una matriz de la siguiente manera:

Z_1	Z_2	Z_3
Z_4	Z_5	Z_6
Z_7	Z_8	Z_9

Ilustración 10: Matriz ejemplo.

Sea Z_5 la celda a la cual se le quiere estimar la pendiente, se calcula un S_{EW} y un S_{NS} como la diferencia de los valores correspondientes a la suma de cada columna en dirección este-oeste y norte-sur respectivamente. Cabe destacar que las celdas que se encuentran aledañas a Z_5 tienen un ponderador por 2 debido a su mayor cercanía a la celda en cuestión.

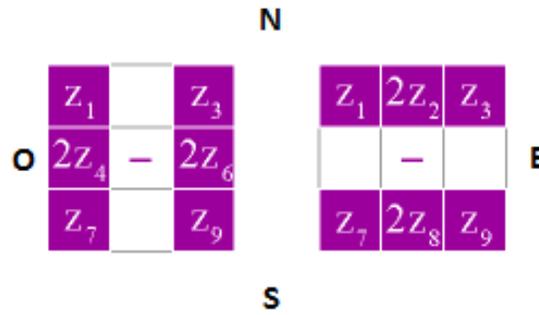


Ilustración 11: Ejemplo simulación por inundación.

Donde:

$$S_{EW} = (Z_1 + 2 * Z_2 + Z_3) - (Z_7 + 2 * Z_8 + Z_9)$$

$$S_{NS} = (Z_1 + 2 * Z_4 + Z_7) - (Z_3 + 2 * Z_6 + Z_9)$$

Ecuación 5: Cálculo S_{EW} y S_{NS}

Finalmente, la pendiente de la celda en cuestión se calcula como:

$$S_5 = \sqrt{S_{EW}^2 + S_{NS}^2}$$

Ecuación 6: Cálculo pendiente por celda.

2.7 Largo real vs Largo aparente

Se define como largo aparente a la distancia que existe entre dos puntos en el plano XY, por lo que si este tramo tiene una pendiente distinta de 0, existirá un largo real el cual difiere del largo aparente.

Por otro lado, se define como largo real a la distancia que existe entre dos puntos considerando un espacio tridimensional. Con esto, existe una clara diferencia con respecto al largo aparente ya que el largo real sí considera las pendientes de los tramos trazados.

2.8 Desviación estándar relativa

La desviación estándar relativa, también conocida como el coeficiente de variación, es la relación entre la desviación estándar de un conjunto de datos y su media, el cual se expresa en porcentaje. Este coeficiente permite comparar dispersiones de distintas distribuciones, siempre y cuando sus medias sean positivas. Con esto, aquella distribución que tenga un coeficiente de variación mayor, corresponderá a aquella que tenga una mayor dispersión en sus datos.

$$DEV\% = \frac{\sigma}{\bar{x}}$$

Ecuación 7: Desviación estándar relativa.

CAPITULO 3 METODOLOGÍA

A continuación, se presentan las 6 etapas en las que se divide el desarrollo de la memoria de título, junto con una descripción de las tareas a desarrollar en cada una de ellas.



Ilustración 12: Metodología.

3.1 Determinación de variables significativas

Dado la gran cantidad de datos que se pueden obtener de una operación minera, es fundamental definir las variables más importantes con las cuales se va a trabajar, ya que de no ser así el modelo de redes neuronales y el modelo de optimización se vuelve demasiado complejo, lo que tiene como consecuencia tiempos de procesamiento excesivamente elevados. Para realizar esto, es de suma importancia contar con una revisión bibliográfica y un estado del arte tal que permita identificar las variables que pueden ser determinantes a la hora de definir un nuevo proceso de asignación de estacionamientos para los CAEX.

3.2 Obtención de datos

Para poder entrenar la red con el fin de que esta logre arrojar los resultados deseados, y para construir un modelo de optimización, se determinó que se utilizaran datos de una faena minera ubicada al norte de Chile interesada en el proyecto. Es importante mencionar que estos datos no solo se utilizaran para el cometido recién mencionado, si no que en una etapa posterior serán aprovechados para simular la operación actual de la mina.

Cabe destacar que dichas variables definidas como significativas no fueron facilitadas directamente, por lo que hubo que calcularlas utilizando los métodos de ponderación de distancia inversa (IDW) y simulación de escenarios mencionados en el capítulo de antecedentes. Los resultados de estos cálculos, junto con una explicación de cómo se obtuvieron se encuentran en el capítulo 5 de este documento.

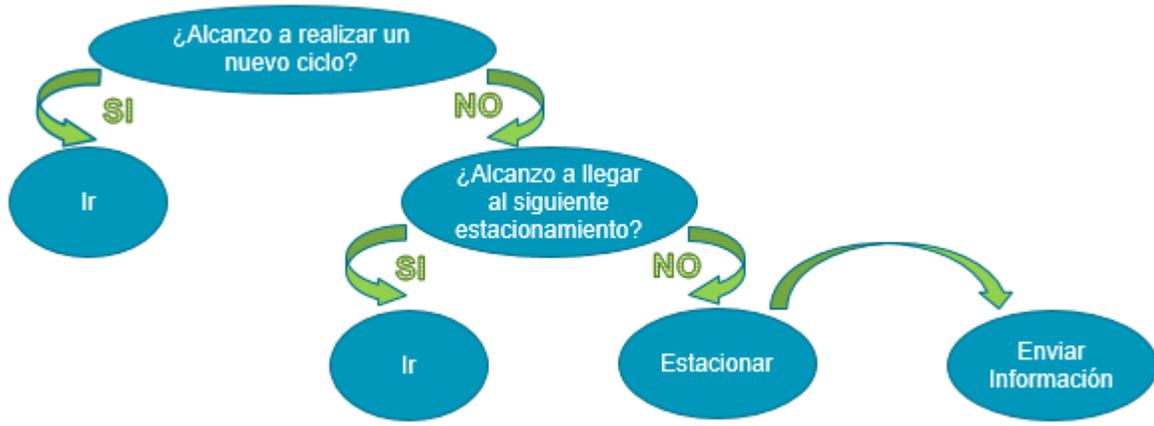
3.3 Aplicación del modelo de redes neuronales

Las redes neuronales se utilizan en este trabajo con el fin de entregar mayor información con la cual se pueda hacer gestión sobre ciertas variables, que afectan tanto de manera positiva como negativa al proceso del cambio de turno y a la productividad de la mina. Para esto, se utilizará una red neuronal que tiene un tipo aprendizaje no supervisado y que trabaja según el método de agrupación k.means. El objetivo de aplicar esta red neuronal es identificar ciertos patrones en los grupos de trabajo los cuales tienden a aumentar o disminuir las demoras en los cambios de turno. Con esto, se pueden determinar las características de los grupos más productivos y detectar con anticipación los operadores que tenderán a perder más tiempo del estimado al final de cada turno (según en el cluster en el que se encuentran) y de esta forma, actuar de manera inmediata corrigiendo ciertos parámetros de operación de cada operador. Además, luego de estudiar a todos los grupos de trabajo, se pueden hacer análisis individuales de cada operador para conformar equipos de características similares y así lograr una extracción más constante a lo largo del tiempo.

3.4 Construcción del modelo de optimización

Para la construcción del modelo de optimización del proceso, se ideó un árbol de decisión capaz de analizar distintas variables con el fin de disminuir los tiempos perdidos al final de cada jornada. Para esto, gracias a los datos obtenidos del GPS de cada camión y a los resultados derivados de la recolección de datos, se logró identificar las velocidades individuales de cada operador en distintos tramos de las rutas y diferenciando su estado de carga. Con esto, una hora antes del cambio de turno cada vez que un CAEX pase por un lugar considerado como estacionamiento, mediante el árbol de decisión el modelo se preguntara si alcanza a realizar otro ciclo completo antes de que termine su jornada laboral. Esto, considerando las características de la ruta que debe recorrer el camión, según los largos de cada tramo en pendiente y en plano, y cuántos de estos tramos debe realizarlos con el camión cargado o vacío. Si el cálculo realizado con las velocidades del operador en cuestión arroja como resultado que el operador si alcanza a dar un ciclo más, entonces la unidad de transporte deberá completar otro ciclo completo antes de cuestionarse su lugar de aparcamiento. Si el cálculo a la pregunta planteada tiene una respuesta negativa, entonces el modelo realizara una segunda pregunta, ¿Alcanzará el camión a llegar al siguiente estacionamiento disponible? Si según las velocidades promedio del operador, y las condiciones del camino hacia el siguiente estacionamiento dan como resultado que es factible realizar el traslado cuestionado antes de la hora del cambio de turno, entonces el camión debe dirigirse al próximo estacionamiento. Si la respuesta es que no alcanza a llegar a tiempo, entonces el operador debe quedarse en estacionamiento por donde se encuentra transitando en ese momento.

Por último, cada vez que a un CAEX se le asigna un estacionamiento, esto se lleva a un registro en donde minutos antes del cambio de turno se tendrá dónde estará estacionado cada camión, facilitando de esta manera la gestión para la asignación del turno entrante. Lo siguiente se muestra en la **¡Error! No se encuentra el origen de la referencia.**, donde se observa que el odelo está en función de las velocidades de cada operador, su ubicación, el largo del circuito y



$$f(V_{i\text{ subida}}, V_{i\text{ bajada}}, V_{i\text{ plano}}, GPS_i, L_{\text{circuito}}, Caracteristica_{\text{circuito}}) = \text{Min}(\text{Demoras cambio de turno})$$

Ilustración 13: Árbol de decisión modelo de optimización.

sus características.

3.5 Simulación de escenarios

Una vez que la red se encuentra entrenada y el modelo de optimización este entregando resultados, se procederá con la realización de una comparación entre una simulación de escenarios. Los escenarios a evaluar son aquel que se encuentra con la asignación de estacionamientos convencional versus aquel se encuentra utilizando el modelo de asignación de estacionamientos para los CAEX desarrollado a lo largo de esta memoria. Cabe destacar que si bien en una primera instancia se podría simular solo el segundo de los casos recién mencionados, para poder concluir sobre los resultados obtenidos se debe hacer una comparación entre escenarios simulados y no entre uno real versus uno simulado.

3.6 Conclusiones y recomendaciones

La última etapa consiste en concluir sobre los resultados obtenidos en la simulación y analizar si se alcanzaron los objetivos planteados al inicio de esta memoria. Además, se entregará una serie de recomendaciones sobre los siguientes pasos a seguir si se quiere probar el modelo y sobre recomendaciones para mejorar el proceso detectadas durante el periodo de tiempo que se realizó este trabajo.

CAPITULO 4 RECOLECCIÓN DE DATOS

Con el fin de construir una red neuronal capaz de abordar los objetivos planteados en esta memoria, se determinaron las variables más significativas en el proceso de asignación de estacionamientos para los CAEX a la hora del cambio de turno, las cuales se muestran a continuación.

- Velocidad de cada camión
 - Con Carga
 - En caminos con pendiente (bajada y subida).
 - En caminos horizontales.
 - Sin Carga
 - En caminos con pendiente (bajada y subida).
 - En caminos horizontales.
- Topografía
 - Rutas utilizadas por los camiones.
 - Por tramos en pendiente y horizontales.
- Ubicación GPS
 - Palas
 - Camiones
 - Zonas de Descarga
 - Estacionamientos habilitados para camiones

Una vez definidas las variables significativas, se procedió a solicitar la información a una faena ubicada al norte de Chile, datos que pudieron ser obtenidos gracias a la buena disposición por parte de la minera y al programa modular, el cual interpreta la información entregada por Dispatch®. Los datos solicitados se recibieron de la siguiente manera.

Tabla 1: Datos entregados por la minera colaboradora.

Dia	Turno	Hora	Equipo	Flota	Este	Norte	Elev	Vel	Cargado	Material	Origen	Destino	Operador	Grupo
16-10-16	D	9:49	TKD0227	C 797 B	16991	106810	28130	12	Si	Marginal1	2755/E04/002	P3F25S	AROS A. JAIME	A
16-10-16	D	9:50	TKD0227	C 797 B	16950	106736	28210	10	Si	Marginal1	2755/E04/002	P3F25S	AROS A. JAIME	A
16-10-16	D	9:50	TKD0227	C 797 B	16937	106680	28280	7	Si	Marginal1	2755/E04/002	P3F25S	AROS A. JAIME	A
16-10-16	D	9:51	TKD0227	C 797 B	17053	106826	28320	22	Si	Marginal1	2755/E04/002	P3F25S	AROS A. JAIME	A
16-10-16	D	9:51	TKD0227	C 797 B	17112	106898	28420	11	Si	Marginal1	2755/E04/002	P3F25S	AROS A. JAIME	A
16-10-16	D	9:52	TKD0227	C 797 B	17175	106984	28530	13	Si	Marginal1	2755/E04/002	P3F25S	AROS A. JAIME	A
16-10-16	D	9:52	TKD0227	C 797 B	17232	107059	28630	11	Si	Marginal1	2755/E04/002	P3F25S	AROS A. JAIME	A
16-10-16	D	9:53	TKD0227	C 797 B	17291	107143	28720	12	Si	Marginal1	2755/E04/002	P3F25S	AROS A. JAIME	A

Con esta información, se logró obtener todas las variables definidas como necesarias para el desarrollo de esta memoria.

CAPITULO 5 RESULTADOS Y ANÁLISIS

5.1 Obtención de los datos significativos

Para poder obtener todas las variables significativas fue necesario hacer una interpolación utilizando las coordenadas de cada: pata, cresta y borde de la topografía facilitada por la minera. Para esto, se cuadrículó dicha topografía en baldosas de $4 \times 4 \text{ m}^2$ y luego se le aplicó el método IDW a cada “celda” utilizando un radio de 800 metros (definido arbitrariamente) para calcular su altura. De esta manera, se pasó de tener un plano a un modelo de elevación digital, donde cada baldosa tiene su propia a coordenada de elevación, con lo que es posible calcular el largo aparente entre distintos puntos.

Para calcular el largo real de los caminos y rampas es necesario conocer sus pendientes, para lo que se utilizó el método de simulación por inundación. De esta forma, se asignaron colores a las celdas según el siguiente criterio:

- Si $S_i < 5^\circ$, se considera un tramo horizontal y se asigna un color verde a la celda.
- Si $5^\circ \leq S_i \leq 15^\circ$, se considera una rampa y se le asigna un color amarillo a la celda.
- Si $15^\circ < S_i$, se considera que es un sector no transitable y se le asigna un color rojo a la celda.

Finalmente se logró visualizar la topografía diferenciando las rutas de los camiones de extracción por tramos horizontales e inclinados, lo que se ilustra a continuación.

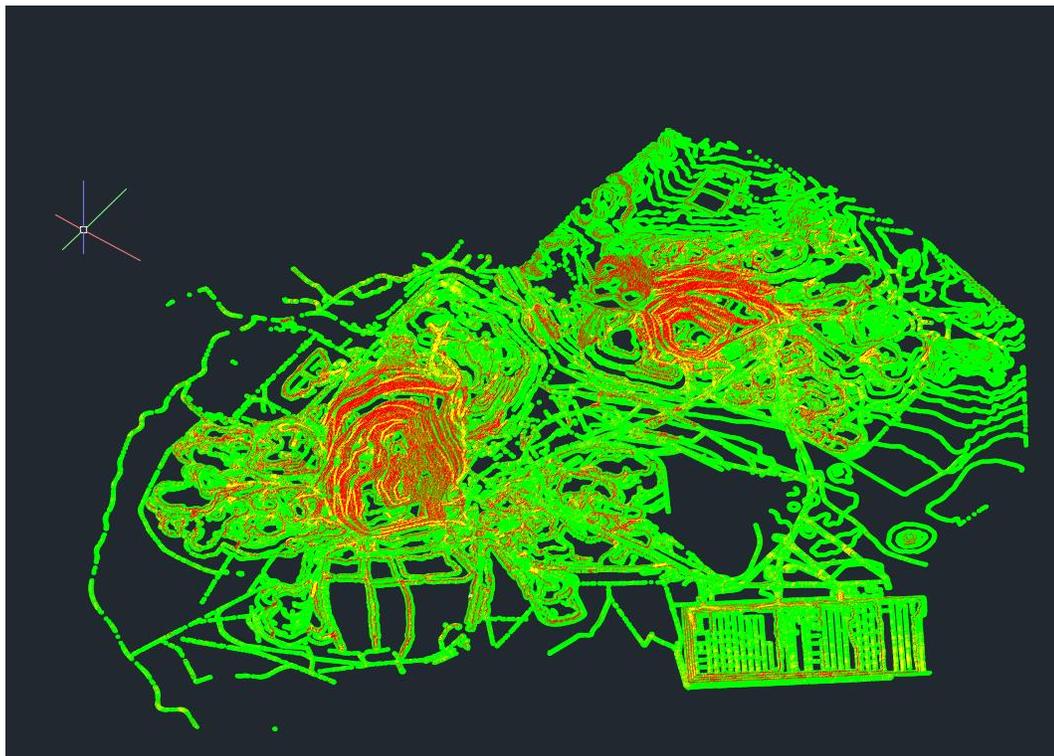


Ilustración 14: Interpolación global de la mina.

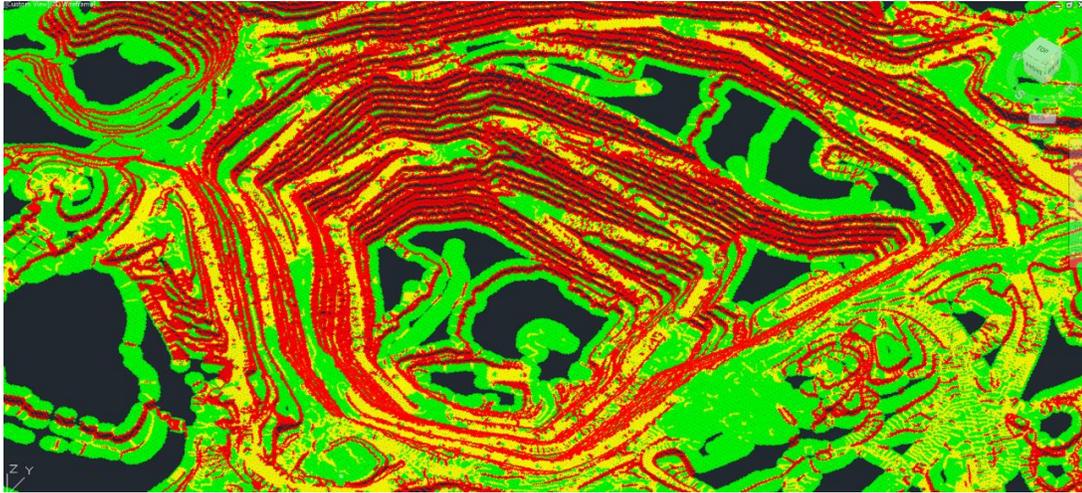


Ilustración 15: Interpolación rajo A

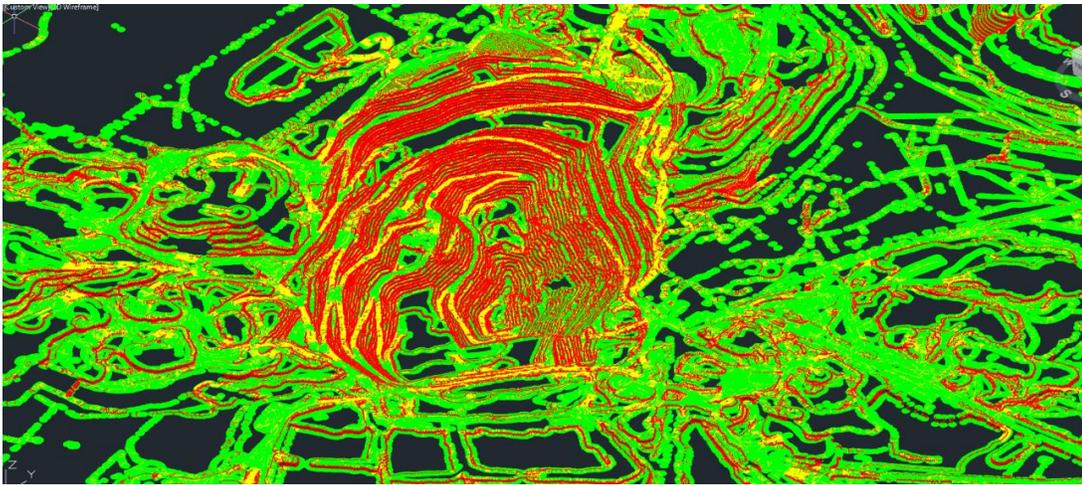


Ilustración 16: Interpolación rajo B.

Una vez finalizada la interpolación y el cálculo de pendientes del terreno, se procedió a sobreponer los datos GPS de los CAEX entregados por la minera y calcular las velocidades de los camiones para distintos tramos. Para esto, se tomaron todos los pares de datos que tuviesen como máximo un periodo de 30 segundos de diferencia entre que fueron emitidos (esto para cada camión), y se calculó la velocidad real de los camiones como la distancia real entre ambos puntos dividido en el tiempo que transcurrió entre que se emitió la primera y segunda señal GPS. Además, se diferenciaron estas velocidades según los tramos en donde fueron calculadas, existiendo velocidades en rampas de subida, de bajada y en tramos horizontales, además de considerar el estado del camión (cargado o vacío).

Cabe destacar que no se consideraron los pares de puntos que tienen una diferencia de transmisión mayor a 30 segundos ya que los camiones debiesen emitir una señal cada 15 segundos. Además, luego de 30 segundos si la unidad de extracción realizó una curva o algún movimiento no convencional, este no sería considerado en la distancia utilizada para calcular las velocidades, provocando errores en los resultados. Cabe mencionar que en los resultados se obtuvieron las velocidades promedios y medianas, en subida, bajada y en plano para cada camión, tal como se observa de manera ejemplar en la tabla

Tabla 3: Velocidades de cada camión en km/hr.:

Equipo	Prom Subida	Media Subida	Prom Bajada	Media Bajada	Prom Plano	Media Plano
TKD0227	11	10	32	32	33	35
TKD0229	11	10	28	31	28	26
TKD0230	14	10	33	29	30	31
TKD0232	10	10	33	36	26	25
TKD0234	10	10	24	27	20	16
TKD0235	10	10	34	37	28	28
TKD0236	11	10	27	32	28	27

Tabla 2: Velocidades promedio de la flota completa.

Subida		Bajada		Plano	
Promedio	Media	Promedio	Media	Promedio	Media
11	11	32	34	26	24

Luego, se realizó el mismo ejercicio pero diferenciando si el camión va cargado o vacío, esto con el fin de poder analizar si es que el estado de carga de la unidad de extracción influye de manera determinante en las velocidades de estos o no. Los resultados obtenidos se muestran a continuación:

Tabla 4: Velocidades de cada camión cargado en km/hr.

Equipo	Cargado					
	Prom Subida	Media Subida	Prom Bajada	Media Bajada	Prom Plano	Media Plano
TKD0227	11	10	20	16	29	29
TKD0229	11	10	11	9	23	20
TKD0230	11	10	22	17	30	30
TKD0232	10	10	9	10	23	19
TKD0234	10	10	12	12	18	13
TKD0235	10	10	19	20	25	23
TKD0236	11	10	14	15	26	26
TKD0238	11	10	17	18	21	16

Tabla 5: Velocidades de cada camión descargado en km/hr.

Equipo	Descargado					
	Prom Subida	Media Subida	Prom Bajada	Media Bajada	Prom Plano	Media Plano
TKD0227	11	11	33	34	38	41
TKD0229	10	9	29	31	33	38
TKD0230	26	10	36	32	30	32
TKD0232	8	6	35	36	29	34
TKD0234	10	10	24	27	24	25
TKD0235	11	10	35	37	32	34
TKD0236	14	12	29	33	30	30
TKD0238	11	10	38	38	27	29

Finalmente, se obtuvieron las velocidades promedios de la flota diferenciado su estado de carga.

Tabla 6: Velocidades promedio de la flota completa diferenciando por estado de carga.

Estado	Subida		Bajada		Plano	
	Promedio	Media	Promedio	Media	Promedio	Media
Cargado	11	11	19	19	24	22
Descargado	13	12	33	35	28	29

Con estos resultados, se observa que existe una gran diferencia entre las velocidades de los camiones dependiendo de su estado de carga, por lo que a la hora de calcular los tiempos estimados que se demorará cada camión a los distintos estacionamientos, la tarea no solo se debe realizar con las velocidades promedios o medias del operador en cuestión, si no que diferenciando el estado de carga de las unidades de transporte a la hora de calcular su tiempo de viaje a los distintos estacionamientos.

5.2 Velocidades de los operadores

Si bien es importante mencionar que las velocidades de los camiones no sufren variaciones debido a que el modelo no propone aumentar la rapidez con la que se desplazan los CAEX, si no minimizar el tiempo que están detenidos, es importante señalar que se identificó una gran diferencia entre distintos operadores en términos de velocidades para las mismas condiciones. Lo siguiente se observa en la Tabla 7, donde se tiene una desviación estándar relativa (DER%) importante para ambos estados de carga del camión.

Tabla 7: Desviación estándar relativa de las velocidades.

Descargado			Cargado		
Subida	Bajada	Plano	Subida	Bajada	Plano
36.8	15.1	28.7	9.9	39.6	22.0

De estos resultados, se tiene que la mayor dispersión en las velocidades de los camiones se encuentra en los tramos de subida cuando los CAEX están descargados, y en los tramos de bajada cuando están cargados. Si bien la desviación estándar relativa es un parámetro que sirve para realizar comparaciones entre datos más que ver la magnitud de la dispersión de estos, resulta llamativo las diferencias que se observan en los rendimientos individuales de los operadores, existiendo casos en que un operador tiene velocidades promedios que doblan a otros (bajo las mismas condiciones). Debido a lo anterior, se recomienda realizar un análisis exhaustivo de los rendimientos individuales de los trabajadores, lo que sin duda facilitará la formación de grupos de trabajo, o la asignación de operadores a los equipos al inicio de cada jornada.

5.3 Redes neuronales

Para la aplicación de redes neuronales se utilizó el algoritmo k-mean, en donde se definió un mínimo de 2 y un máximo de 5 agrupaciones a formar. El resultado al aplicar dicho algoritmo

Tabla 8: Centroides grupos redes neuronales.

Variables	Cluster 1	Cluster 2	Cluster 3
Distancia [m]	10505	8585	7574
Minutos restantes [min]	29	28	32
Estacionó en Pala	0	1	0
Velocidad Ascenso [km/hr]	13	11.6	9.4
Velocidad Descenso [km/hr]	30.5	26.5	20.5
Velocidad Plano [km/hr]	24.7	28	26.3

fue que las agrupaciones más consistentes y claras se obtuvieron con 3 grupos, en donde los centroides correspondientes se muestran a continuación.

En donde:

Distancia: Metros a recorrer hasta el próximo estacionamiento.

Minutos restantes: Tiempo restante entre que el camión estacionó y la hora del cambio de turno.

Estaciono en pala: 0 si es que el camión no estaciono en pala, 1 si es que sí estaciono en pala.

Los gráficos obtenidos son los siguientes.

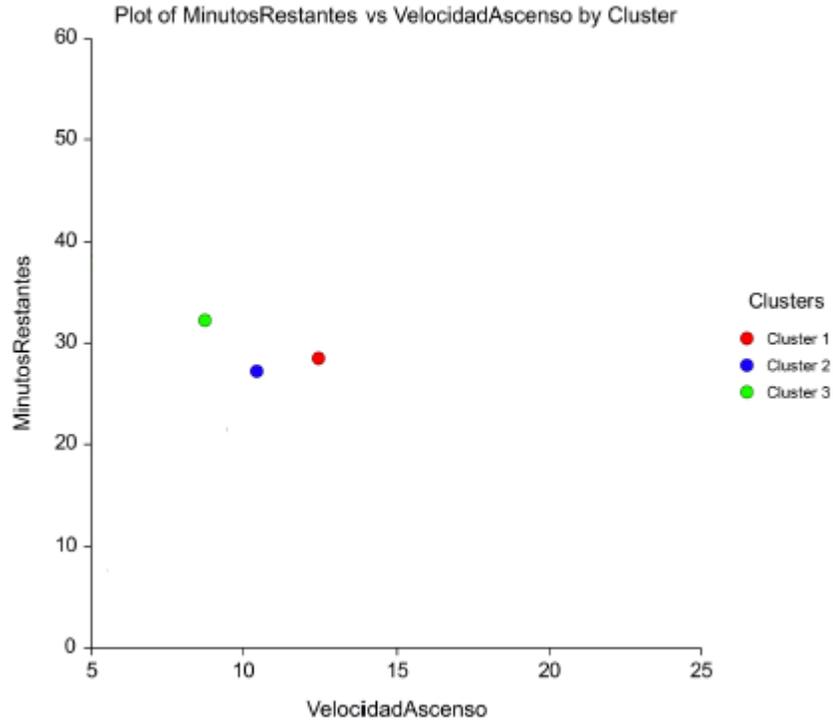


Ilustración 17: Gráfico minutos restantes v/s Velocidad Ascenso.

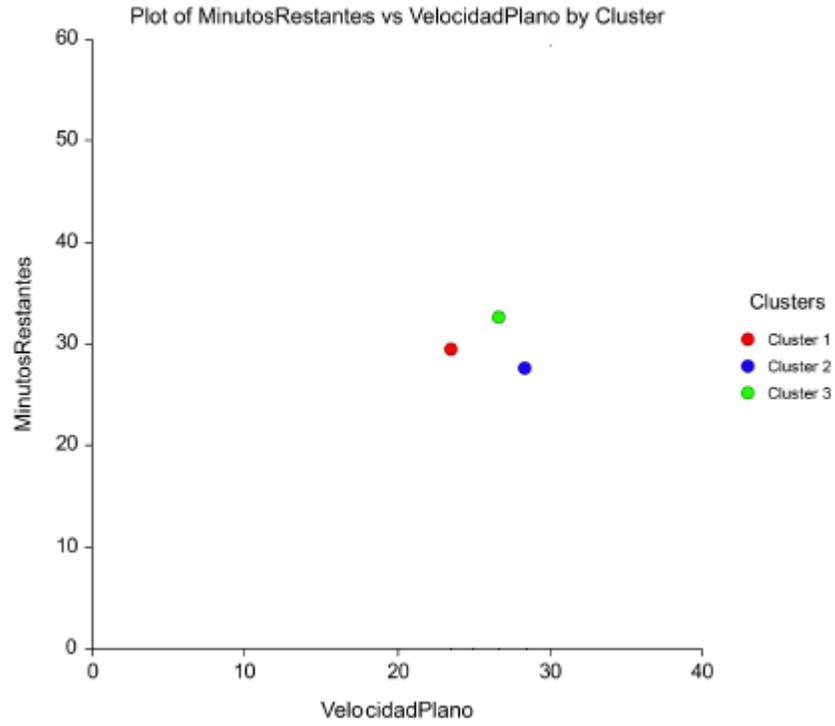


Ilustración 18: Gráfico minutos restantes v/s Velocidad plano.

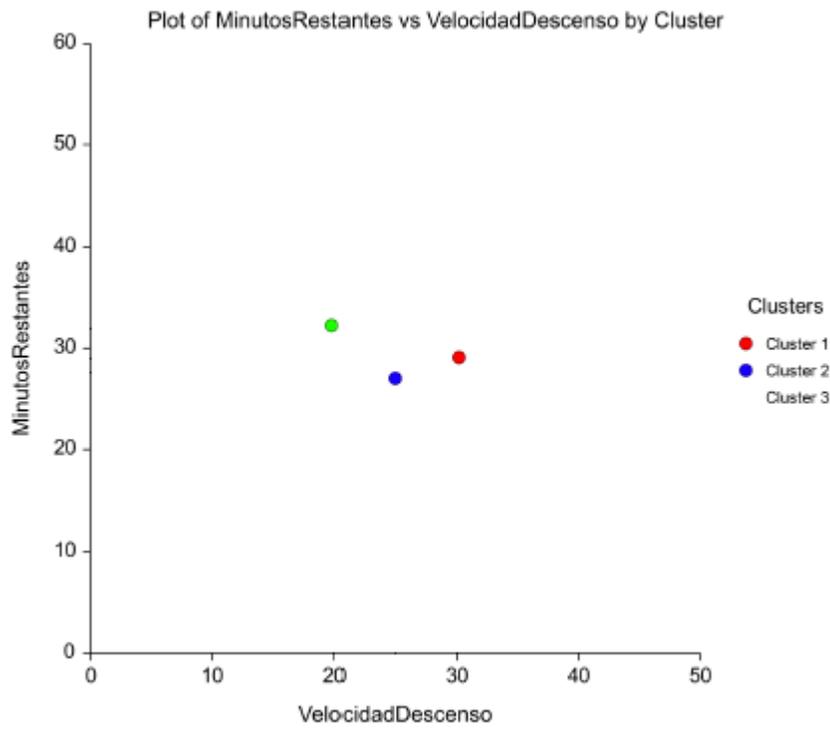


Ilustración 19: Gráfico minutos restantes v/s Velocidad descenso.

De los gráficos anteriores, se observa claramente que el grupo 3 es el que tiende a perder la mayor cantidad de minutos a la hora del cambio de turno, además de ser en 2 de los 3 casos, el grupo que trabaja a velocidades más bajas. Debido a estos resultados, se recomienda que a la hora de armar los grupos de trabajo se intente mezclar a los operadores que van lento con aquellos que circulan más rápido, con el objetivo de aumentar la productividad al final de los turnos y tener una producción más constante.

Por otro lado, se analizó el gráfico obtenido de distancia v/s minutos restantes, en donde se observa que el grupo que se encuentra más cerca de los estacionamientos 1 hora antes del cambio de turno, tiende a perder más tiempo al final de la jornada ya que se queda en el estacionamiento más cercano y espera el fin del turno en ese lugar.

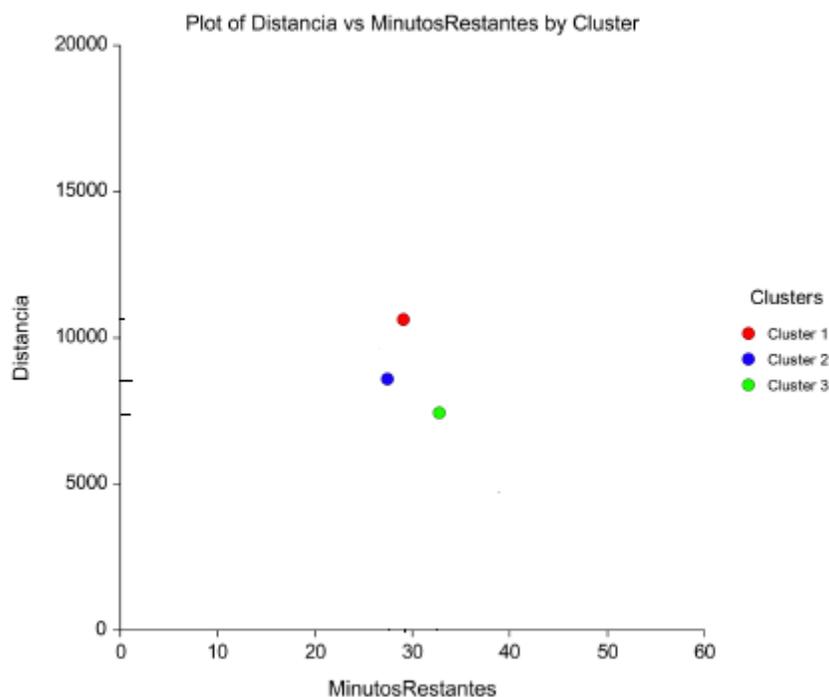


Ilustración 20: Gráfico minutos restantes v/s Velocidad descenso.

Por último, cabe destacar que se obtuvieron más gráficos de los cuales se puede seguir concluyendo distintos hitos, pero el objetivo de esta memoria en cuanto a redes neuronales se refiere, es demostrar que dichas redes se pueden utilizar como una herramienta importante para realizar inteligencia y gestión sobre los datos generados por los distintos dispositivos y programas que utilizan las compañías mineras diariamente.

5.4 Tiempos perdidos

A continuación se muestran los resultados obtenidos para las demoras en los cambios de turno en el caso donde no se aplicó el modelo, y en el caso en donde se utilizó la optimización de asignación de estacionamientos para una flota de 130 CAEX.

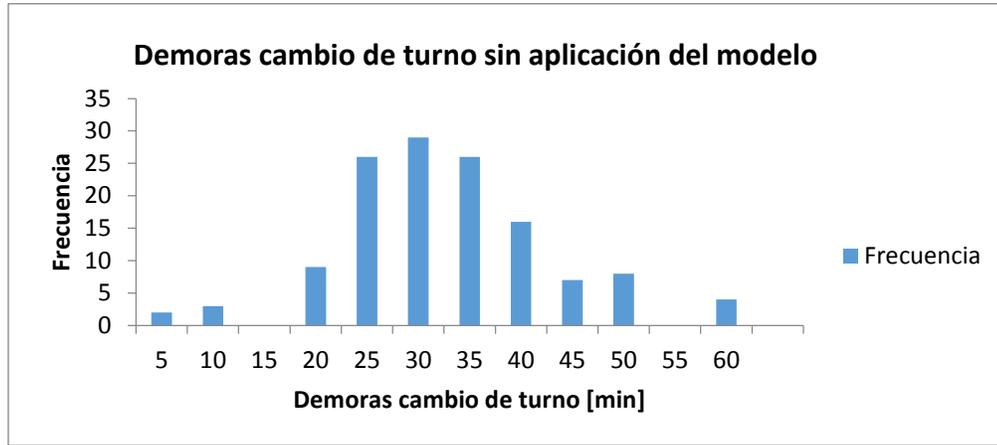


Ilustración 21: Histograma demoras cambio de turno sin aplicación del modelo de optimización.

Tabla 10: Demoras cambio de turno sin aplicar proceso de optimización.

Demoras cambio de turno				
Minutos	Horas	Días	Mediana	Promedio
3984	66	2.8	28	30

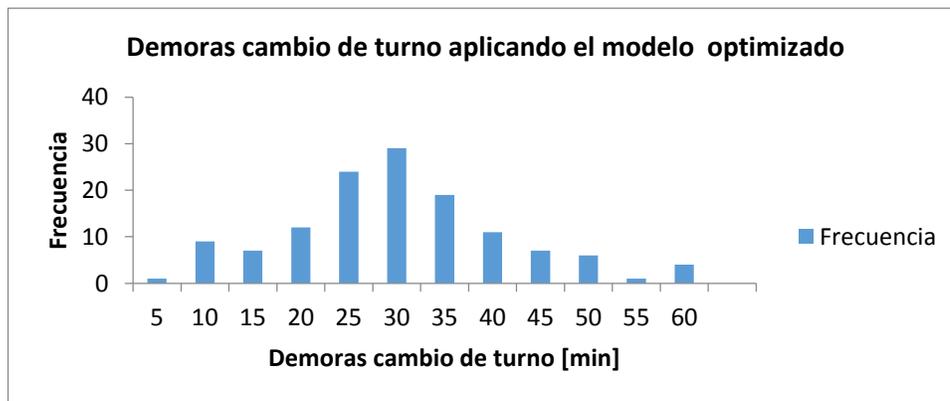


Ilustración 22: Histograma demoras cambio de turno con el modelo de optimización.

Tabla 9: Demoras cambio de turno aplicando el proceso de optimización.

Demora cambio de turno proceso optimizado				
Minutos	Horas	Días	Mediana	Promedio
3586	60	2.5	26	28

Se puede observar que para un cambio de turno estándar, se disminuyeron 398 minutos en las demoras asociadas a este proceso, reduciendo el 9,8% del total de minutos perdidos, lo que es equivalente a 6:38 horas. Además, dentro de la simulación se contabilizaron 24 descargas adicionales cuando se asignaron los estacionamientos según el modelo planteado. Suponiendo una carga promedio de 330 toneladas por camión, esto equivale a 7,900 toneladas extras por turno.

Para el mismo caso, si llevamos los números a años, se ganarían 386,560 minutos (o 4,776 horas), lo que se vio reflejado en 17280 descargas, y asumiendo la misma carga promedio por camión, esto resulta en 5,702,400 toneladas de material acarreado de manera adicional. Para cuantificar este resultado, se asumieron los siguientes supuestos:

1. El valor se dedujo solamente del ahorro en el costo por transporte.
2. El costo por transporte es de 1 dólar la tonelada.
3. No se asumieron otros beneficios tales como la ganancia en producción.
4. Se realizó una variación porcentual del 20% para establecer un rango del impacto que generara el modelo de optimización de asignación de estacionamientos dentro de un año

El resultado obtenido fue que se tiene un ahorro anual entre 4.5 y 6.8 [MUSD] para la flota estudiada.

CAPITULO 6 CONCLUSIONES Y RECOMENDACIONES

6.1 Conclusiones

Luego de finalizar el trabajo, se concluye que existe una clara oportunidad de mejora en lo que se refiere a la asignación de estacionamientos de los camiones de extracción para el cambio de turno. Esto se debe a que al automatizar un proceso industrial mediante un árbol de decisión que procesa la información y que toma sus decisiones de manera personalizada para cada operador, se logra disminuir las demoras asociadas al cambio de turno de manera significativa (9.8%) sin la necesidad de realizar una intervención invasiva a la operación. Además, esta disminución en las demoras afecta directamente a la producción, ya que se refleja en una mayor cantidad de descargas al final del turno.

Por otro lado, las decisiones que se toman aplicando el modelo tienen una justificación matemática y personificada, lo que la diferencia del método utilizado actualmente. De esta manera, existe un respaldo en el cual se basa la decisión y sirve para justificar la asignación a aquellas personas que cuestionen su estacionamiento designado. Esto se concluye ya que al estudiar el proceso actual, se descubrió que en muchos casos a los operadores se les determinaba un estacionamiento, y este no era respetado con la justificación de que aquellos que tenían que trasladar el equipo al nuevo estacionamiento, no tenían la seguridad de si alcanzarían a llegar a la hora para no perder el transporte que los lleva al exterior de la mina, aumentando las demoras asociadas al cambio de turno.

Con respecto a la robustez del modelo, se concluye que este aspecto puede ser mejorado incluyendo un mayor número de restricciones, tales como la capacidad de todos los estacionamientos, o la cantidad de camiones que deben quedar cargados antes de terminar cada jornada. La razón por la cual no se incluyeron estas condiciones de borde se debe a que no se dispuso de toda la información solicitada a la faena, ya que el traspaso de información hacia empresas externas (Hexagon) es limitado por términos de confidencialidad. Lo anterior, se traduce en una excelente oportunidad para disminuir la incertidumbre generada por la simulación, ya que se pueden incorporar al modelo tantas condiciones como sean posibles antes de realizar una eventual prueba en terreno.

En cuanto a la velocidad a la que trabajan los operadores, se detectó una gran diferencia en el rendimiento del personal encargado de manejar los CAEX, existiendo casos en que para las mismas condiciones de carga y de ruta (en pendiente o plano), había operadores que transitaban un 50% más rápido que otros. Con esto, se concluye que el rendimiento de los operadores no debiese evaluarse solo por la cantidad de ciclos realizados o tonelaje que logran al final del turno, sino también comparando la velocidad a la que transitó cada operario con la velocidad a la que el equipo debiese operar según su ficha técnica.

Por último, se concluye que las redes neuronales pueden ser bastante útiles a la hora de realizar gestión e inteligencia en cuanto a la formación de grupos se refiere, además de detectar ciertas características que tienen algunos grupos de trabajo que los hacen ser menos productivos al final de cada jornada. Además, existe la posibilidad de seguir trabajando con las redes neuronales al punto de definir una cierta cantidad de grupos, y realizar un árbol de decisión para cada uno de ellos. De esta manera, se podría seguir optimizando y personalizando el proceso de asignación de estacionamiento para los camiones de extracción y de esta forma seguir mejorando en la productividad de los operadores y de la mina.

6.2 Recomendaciones

A medida que se fue profundizando en el tema del cambio de turno se detectaron varias oportunidades de mejora, que si bien quedaron fuera del alcance de esta memoria, es importante mencionarlas ya que se cree que son una gran oportunidad para seguir mejorando el proceso del cambio de turno.

La primera oportunidad detectada se encuentra en mejorar el protocolo de los talleres de mantención a la hora de solicitar un equipo. Lo anterior se debe a que existen muchos casos en que los talleres piden que le entreguen un equipo para realizarle la mantención respectiva dentro de los últimos minutos de cada turno. El problema se encuentra en que a esa altura del turno, el camión ya tiene asignado un estacionamiento y por ende, alguien del turno entrante ya se dirige a dicho estacionamiento para operar la unidad durante su jornada. Lo anterior, tiene como consecuencia que al llegar el operador al lugar, éste no encuentra su camión asignado, teniendo que ser reasignado a otro equipo, perdiendo valiosos minutos para la operación por una simple descoordinación. Es por esto que se recomienda reevaluar el protocolo de los talleres de mantención, impidiendo que los mantenedores soliciten equipos después de que los estacionamientos de los camiones hayan sido asignados.

Por otro lado, se detectó una segunda oportunidad de mejora en el proceso de entrada del personal a su jornada. Lo anterior se debe a que actualmente los vehículos de transporte que tienen como objetivo llevar a los operarios a su lugar de trabajo, trasladan grupos que contienen a 2 o hasta 3 operadores de pala, junto con el personal necesario para operar los camiones que se encuentran estacionados al lado del equipo de excavación. Esto, tiene como consecuencia que el último equipo¹ debe ir a dejar a 1 o 2 equipos antes de comenzar a operar, perdiendo minutos valiosos para la operación. En consecuencia, se recomienda realizar una evaluación económica sobre la adquisición de una van por pala, en donde se cuantifique la cantidad de minutos que se podría ahorrar en toda la operación gracias a este cambio, y se concluya si es o no rentable la adquisición de dichos equipos.

En cuanto a los datos con los que se trabajó a lo largo de este informe, se tuvieron bastantes dificultades para su procesamiento debido a la poca precisión que tienen los sistemas GPS ubicados en los equipos mineros. Para resolver dicha problemática, fue necesario incorporar a los algoritmos programados un código capaz de realizar una limpieza de datos, transformándolos a un sistema de alta precisión. Debido a que con la información recibida por los GPS de los equipos se pueden realizar mejoras importantes gracias al “Big Data Analysis”, se recomienda que los sistemas GPS de los camiones y palas sean de alta precisión, ya que de no ser así, se abre la posibilidad de que algunos análisis y decisiones mineras que se basen en la data proporcionada por los equipos sean equívocas o poco acertadas.

¹ Equipo: Un operador de pala junto con los operadores de los camiones que se encuentran estacionados al lado de dicha pala.

CAPITULO 7 BIBLIOGRFÍA

- [1]. Yarmuch, J. (2014). Apuntes del curso MI5071 Sistemas mineros, Universidad de Chile.
- [2]. Bonzi, J. 2007. Propuesta de mejora de la utilización efectiva en base a disponibilidad de la flota de carguío y transporte en minera los pelambres. Memoria ingeniería civil de minas. Santiago, Chile: Facultad de Ciencias Físicas y Matemáticas. 15h.
- [3]. Stuart Russell, Peter Norvig (2009). Artificial Intelligence: A Modern Approach.
- [4]. Ben Krose & Patrick van der Smagt (1996). *An introduction to neural networks*.
- [5]. B. Muller, J. Reinhardt, M.T. Strickland (1995). *Neural Networks. An Introduction*.
- [6]. M. Jordan, J. Kleinberg, B. Scholkopf (2006). *Pattern Recognition and Machine Learning*.
- [7]. *Hacker's guide to Neural Networks (2016)*. [en línea] <<http://karpathy.github.io/neuralnets/>>.
- [8]. Michael A. Nielsen (2015). *Neural Networks and Deep Learning*. Determination Press.
- [9]. Gonzalez, R. 2016. Optimización de las horas operativas de los CAEX en los procesos de descarga de chancado y abastecimiento de combustible, mediante la utilización de modelamientos compuestos y redes neuronales. Memoria ingeniería civil de minas. Santiago, Chile: Facultad de Ciencias Físicas y Matemáticas.
- [10]. Minsky. M. Kohonen's Selg Organizing Feature Maps. [en línea] < <http://www.ai-junkie.com/ann/som/som1.html>> [consulta: 03 noviembre 2016].

CAPITULO 8 ANEXOS

8.1 Anexo A: Contexto Empresa

Hexagon es una empresa enfocada en tecnologías de medición industrial, la cual se ha expandido a soluciones geoespaciales y software que generan sinergias, para de esta forma establecerse como líder mundial en soluciones con tecnologías de información. La cual está presente en 46 países y cuenta con 16.000 empleados.

Además, es una empresa enfocada en la innovación y desarrollo, invirtiendo el 11% de sus ventas en esta área y contando con más de 3.400 empleados y 3.200 patentes.



Ilustración 23: Enfoue principal de Hexagon .

8.1.1 Visión

En Hexagon se aspira a desempeñar un rol de líderes en el esfuerzo para resolver los desafíos que enfrenta nuestro mundo, proporcionando tecnologías de la información innovadoras que hacen un impacto positivo y duradero.

8.1.2 Misión

Hexagon está dedicada a ofrecer información útil a través de tecnologías que permiten a los clientes cambiar de forma inteligente a través de los diversos pasajes de la industria.

8.1.3 Sectores de negocio

Hexagon se enfoca en dos grandes áreas de negocio, las soluciones geoespaciales y para empresas industriales, llegando a los distintos mercados por uno u otro aspecto. El resumen de industrias en las cuales se encuentran soluciones de esta empresa, se encuentran en la ilustración 40.

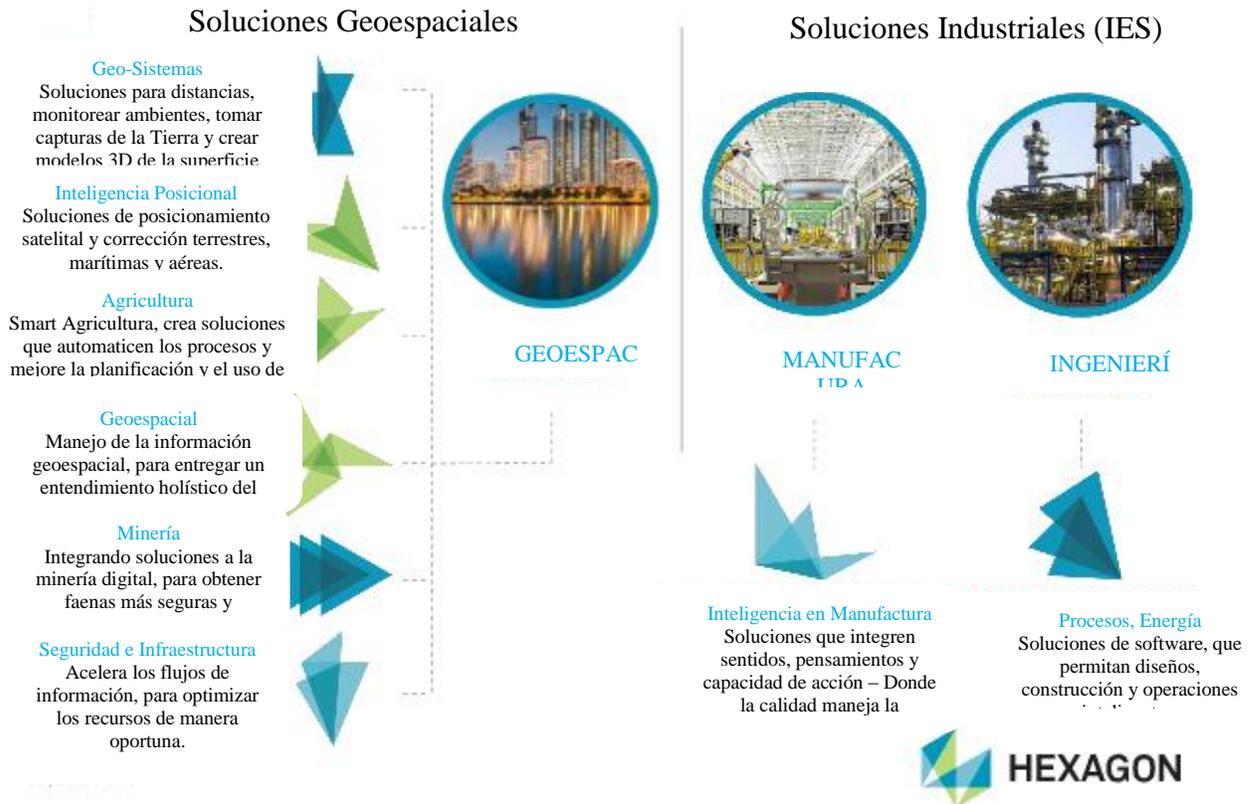


Ilustración 24: Sectores de Negocios – Hexagon (Fuente: Presentación Corporativa).

En la siguiente ilustración se encuentran los ingresos por ventas que ha tenido la empresa desde el año 2000, en el cual se comenzó con la estrategia de negocios actual, llegando a su record de ventas el año 2015 alcanzado los 3.000 millones de Euros y una utilidad de un 22,8% aproximadamente. Además, en el mismo grafico se puede ver la transición al nuevo sector de negocios de Hexagon y como fueron aportando al ingreso neto anual.

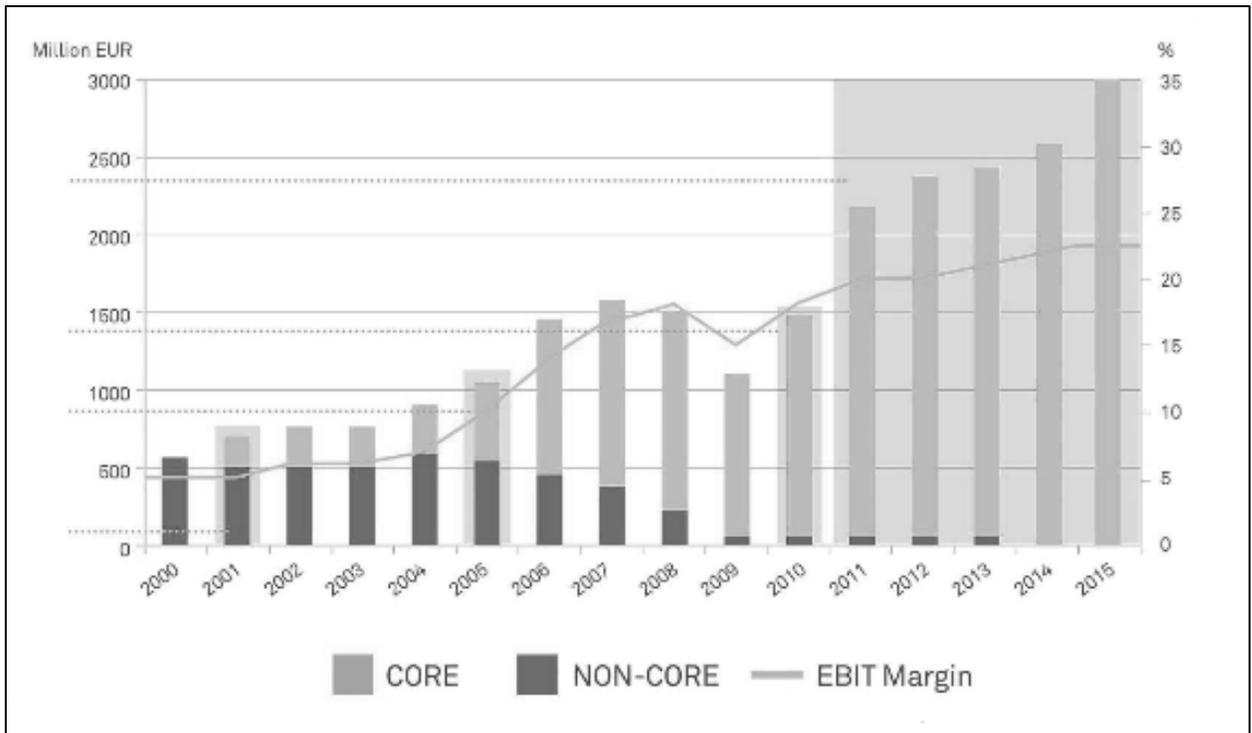


Ilustración 25: Ventas y márgenes anuales de Hexagon (Fuente: Presentación corporativa)

Ahora, también es relevante observar como contribuye cada una de las grandes áreas de negocios en la empresa, esto se puede observar en la ilustración a continuación:

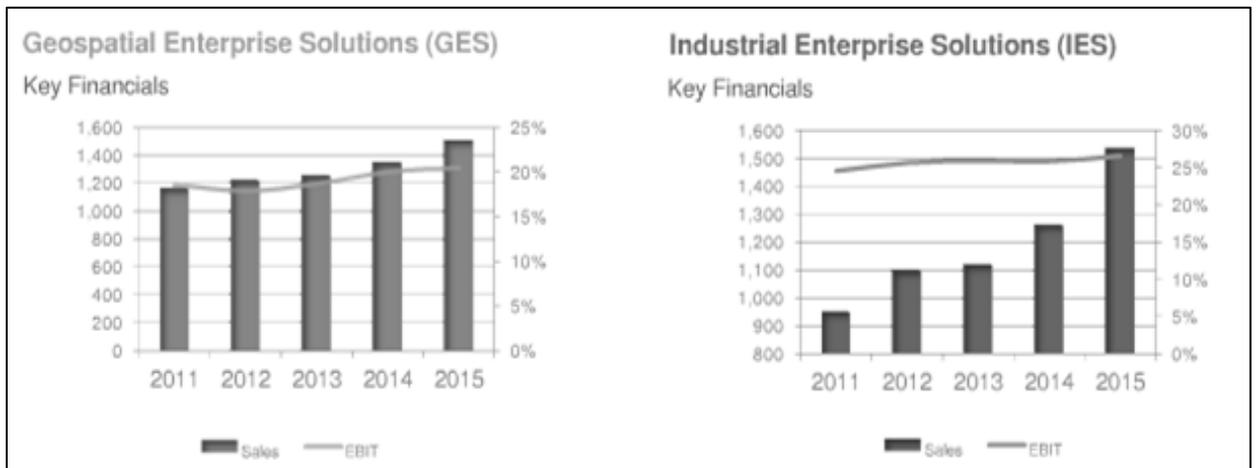


Ilustración 26: Ingresos por área de negocio - Hexagon (Fuente: Presentación Corporativa).

En la Ilustración 26, la situación de la izquierda se explica por un fuerte desarrollo en Europa Occidental y América del Norte, sumado a las nuevas iniciativas y productos geoespaciales de Leica. Por su parte el gráfico de la derecha se debe a un desarrollo en la inteligencia en las industrias de automovilismo, aeroespacial y electrónica, principalmente.

En el ámbito minero, Hexagon se ha posicionado como una de las grandes empresas colaboradoras de tecnologías a la industria, llegando al segundo lugar del ranking realizado por Mining Journal, en donde se evaluaron más de 100 parámetros y proveedores desde la exploración hasta el procesamiento de mineral. La lista completa se puede observar en la Tabla 11.

Tabla 11: Ranking de empresas líderes en tecnología minera.

Mining technology supply leaders 2015				
Major software/hardware suppliers to miners: SAP, IBM, Microsoft, Oracle, Autodesk, Pitney Bowes				
Mining/mineral processing equipment manufacturers also supplying technology: Atlas Copco, Caterpillar, Sandvik, Outotec, Joy Global, Boart Longyear, Komatsu*, Hitachi**				
Leading dedicated technology suppliers to the mining/metals industry (including exploration)				
	Company	Headquartered	Principal mining technology business/s	Public/private
1	ABB/ABB Ventyx	Switzerland	Monitoring, control and automation systems; software	Public
2	Hexagon/Hexagon Mining	Sweden	Software; monitoring and control systems	Public
3	Glencore Technology	Australia	Develops, markets and supports technologies for mineral processing/metals smelting and refining	Public
4	Dassault Systemes	France	Software	Private
5	Siemens	Germany	Industrial process control technologies	Public
6	Rajant Corporation	USA	Communications technology/software	Private
7	Trimble Navigation	USA	Surveying/communication technologies; software	Public
8	Honeywell	USA	Process control/automation technologies	Public
9	Schneider Electric	France	Plant monitoring/automation technologies	Public
10	Bentley Systems	USA	Software	Private
11	Becker Mining Systems	Germany	Communications technology/software	Private
12	Modular Mining Systems	USA	Software/monitoring, control and automation systems	Public (part of Komatsu)
13	Endress+Hauser	Germany	Instrumentation/monitoring and control technologies	Private
14	Maptek	Australia	Software; surveying technology	Private
15	Mine Site Technologies	Australia	Communications technology/software	Private
16	Krohne Group	Germany	Industrial process control technologies	Private (KROHNE Messtechnik is public-listed entity)
17	Thermo Fisher Scientific	USA	Elemental analysis technologies	Public
18	Immersive Technologies	Australia	Training simulation technologies	Private
19	GroundProbe	Australia	Surveying/communication technologies; software	Private
20	Rockwell Automation	USA	Plant monitoring/automation technologies	Public
21	Remote Control Technologies	Australia	Remote control, machine guidance, machine monitoring technologies	Private
22	Downer Group	Australia	Software (Otraco, Snowden)	Public
23	Wenco	Canada	Software/monitoring, control and automation systems	Public (part of Hitachi)
24	Tiefenbach Control Systems	Germany	Automation/control technologies	Private
25	TOMRA Sorting Solutions	Norway	Mineral sorting technologies	Public
26	Micromine	Australia	Software	Private
27	Davey Bickford	France	Mine blast initiation technologies	Private
28	Cavotec	Switzerland	Remote control, machine guidance, machine monitoring technologies	Private
29	ARANZ Geo	New Zealand	Software	Private
30	acQuire	Australia	Software	Private

*Komatsu includes Modular Mining Systems sales. ** Hitachi includes Wenco Mining Systems sales

8.2 Anexo B: Código Memoria

```
namespace MemoriaDiego
{
    public class Program
    {
        static int delta = 800;
        static double p = 16;
        static int res = 4;
        static void Main(string[] args)
        {
            List<Muestra> data = GetData(@"C:\Users\Marinela-
pc\Desktop\datadiego\data.csv");
            List<Muestra> subdata = GetSubData(data);
            Dictionary<string, List<Muestra>> muestrasByEquipo =
GetMuestrasByEquipo(subdata);
            Dictionary<string, Detencion> detenciones = Detenciones(muestrasByEquipo);
            Dictionary<string, List<double[]>> circuitos = AnalyzeCircuits(muestrasByEquipo);
            Dictionary<string, double[]> velocidades = LoadSpeeds(@"C:\Users\Marinela-
pc\Desktop\datadiego\velocidades.csv");
            List<EquipoDetencion> equiposDetenciones = OutputConsolidado(detenciones,
circuitos,
velocidades,
@"C:\Users\Marinela-
pc\Desktop\datadiego\consolidado.csv");
            Dictionary<string, double[]> circuitosExtra = new Dictionary<string, double[]>();
            foreach (EquipoDetencion ed in equiposDetenciones)
            {
                double restantes = ed.MinutosRestantes;
                bool enPala = ed.EnPala;

                while (true)
```

```

    {
        double vp = GetVelocidad(ed);
        double tiempo = GetDistancia(ed, enPala, vp);
        // no alcanzo a un nuevo circuito
        if (tiempo > restantes)
            break;
        Console.WriteLine("Alcanzo");
        double t2 = (ed.Distancia * 0.5) / vp;
        if (t2 > restantes - tiempo)
            break;
        Console.WriteLine("Alcanzo nuevo ciclo");
        if (!circuitosExtra.ContainsKey(ed.Equipo))
            circuitosExtra[ed.Equipo] = new double[] { 0, 0 };
        circuitosExtra[ed.Equipo][0]++;
        circuitosExtra[ed.Equipo][1] += tiempo;
        restantes -= (tiempo + t2);
    }
}

OutputOptimizado(circuitosExtra, @"C:\Users\Marinela-
pc\Desktop\datadiego\optimizado.csv");

OutputDetencionesOptimizado(detenciones, circuitosExtra, @"C:\Users\Marinela-
pc\Desktop\datadiego\detencionesOptimizado.csv");

Console.WriteLine();
}

private static double GetDistancia(EquipoDetencion ed, bool enPala, double vp)
{
    return (ed.Distancia * (enPala ? 1 : 1.5)) / vp;
}

private static double GetVelocidad(EquipoDetencion ed)
{
    double vp = (ed.VelocidadAscenso + ed.VelocidadDescenso + ed.VelocidadPlano) / 3;

```

```

        vp = vp * 1000 / 60;
        return vp;
    }

    private static Dictionary<string, List<double[]>> AnalyzeCircuits(Dictionary<string,
List<Muestra>> muestrasByEquipo)
    {
        Dictionary<string, List<double[]>> circuitos = new Dictionary<string,
List<double[]>>();

        Dictionary<string, List<Muestra[]>> muestrasCircuito = new Dictionary<string,
List<Muestra[]>>();

        foreach (string equipo in muestrasByEquipo.Keys)
        {
            List<Muestra> muestras = muestrasByEquipo[equipo].OrderBy(d =>
d.Fecha).ToList();

            bool comenzarAnalisis = false;
            double distancia = 0;
            string origen = "";
            string destino = "";
            double minutes = 0;
            for (int i = 1; i < muestras.Count; i++)
            {
                Muestra m1 = muestras[i - 1];
                Muestra m2 = muestras[i];
                if (m1.Cargado == false && m2.Cargado == true)
                {
                    comenzarAnalisis = true;
                    distancia = 0;
                    minutes = 0;
                    origen = m2.Origen;
                    destino = m2.Destino;
                }
            }
        }
    }

```

```

if (m1.Cargado == true && m2.Cargado == false)
{
    comenzarAnalisis = false;
    string key = $"{origen}_{destino}";
    if (!circuitos.ContainsKey(key))
        circuitos[key] = new List<double[]>();
    circuitos[key].Add(new double[] { distancia, minutos });
}
if (comenzarAnalisis)
{
    double seconds = (m2.Fecha - m1.Fecha).TotalSeconds;
    if (seconds <= 30)
    {
        double hip = m1.Distancia(m2);
        if (hip <= 1000)
            distancia += hip;
        minutos += seconds;
    }
}
}

OutputCircuitos(circuitos,                                     @"C:\Users\Marinela-
pc\Desktop\datadiego\circuitos.csv");

return circuitos;
}

private static Dictionary<string, Detencion> Detenciones(Dictionary<string,
List<Muestra>> muestrasByEquipo)
{
    Dictionary<string, Detencion> detencionesEquipos = new Dictionary<string,
Detencion>();

    foreach (string equipo in muestrasByEquipo.Keys)

```

```

{
    Muestra det = muestrasByEquipo[equipo].Where(m => m.Fecha.Hour >= 18 &
m.Fecha.Hour <= 19).OrderByDescending(m => m.Fecha).FirstOrDefault();
    if (det != null)
    {
        DateTime f = det.Fecha;
        DateTime fecha = new DateTime(f.Year, f.Month, f.Day, f.Hour + 1, 0, 0);
        detencionesEquipos[equipo] = new Detencion()
        {
            DistanciaUltimoCircuito = 0,
            Equipo = equipo,
            HoraDetencion = det.Fecha,
            MinutosRestantes = (fecha - f).TotalMinutes,
            LastUpdate = det,
            Origen = det.Origen,
            Destino = det.Destino
        };
    }
}
List<ICoord> polyNorte = GetPolyRajoNorte();
List<ICoord> polySur = GetPolyRajoSur();
DxfDocument dxf = new DxfDocument();
List<ICoord[]> estacionamientos = new List<ICoord[]>();
double delta = 500;
foreach (string equipo in detencionesEquipos.Keys)
{
    Detencion d = detencionesEquipos[equipo];
    Text texto = new Text(equipo, new Vector2(d.LastUpdate.X, d.LastUpdate.Y), 10);
    texto.Color = AciColor.Green;
    dxf.AddEntity(texto);
}

```

```

// Generaremos un nuevo boundingbox
ICoord detencion = d.LastUpdate;
if (!detencion.IsInPolygons(estacionamientos))
    estacionamientos.Add(detencion.GetArea(delta));
    d.EnPala = detencion.IsInPolygon(polyNorte.ToArray()) ||
detencion.IsInPolygon(polySur.ToArray());
}
foreach (Punto[] estacionamiento in estacionamientos)
{
    List<PolylineVertex> vertices = new List<PolylineVertex>();
    foreach (Punto p in estacionamiento)
    {
        vertices.Add(new PolylineVertex(p.X, p.Y, p.Z));
    }
    Polyline poly = new Polyline(vertices, true);
    poly.Color = AciColor.Red;
    dxf.AddEntity(poly);
}
dxf.Save(@"C:\Users\Marinela-pc\Desktop\datadiego\detenciones.dxf");

OutputDetenciones(detencionesEquipos, @"C:\Users\Marinela-
pc\Desktop\datadiego\detenciones.csv");
return detencionesEquipos;
}
private static double[] DoGetMeanAndAverage(List<double> data)
{
    if(data.Count > 0)
    {
        int numberCount = data.Count();
        int halfIndex = data.Count() / 2;
        int hi = halfIndex - 1;

```

```

var sortedNumbers = data.OrderBy(n => n);
double median;
if ((numberCount % 2) == 0)
{
    median = ((sortedNumbers.ElementAt(halfIndex)
sortedNumbers.ElementAt(hi)) / 2);
}
else
{
    median = sortedNumbers.ElementAt(halfIndex);
}
double average = data.Average();
return new double[] { average, median };
}
return new double[] { 0, 0 };
}
#region "Init - Inicializa data"
private static List<Muestra> GetData(string filename)
{
    StreamReader sr = new StreamReader(filename);
    List<Muestra> muestras = new List<Muestra>();
    while (!sr.EndOfStream)
    {
        string linea = sr.ReadLine();
        string[] comps = linea.Split(';');
        Muestra m = new Muestra()
        {
            Fecha = DateTime.Parse($"{comps[0]} {comps[2]}"),
            Turno = comps[1],
            Equipo = comps[3],

```

```

        Flota = comps[4],
        X = double.Parse(comps[5]),
        Y = double.Parse(comps[6]),
        Elev = double.Parse(comps[7]),
        Vel = double.Parse(comps[8]),
        VelCalc = double.Parse(comps[9]),
        Cargado = (comps[10].Equals("Si")) ? true : false,
        Material = comps[11],
        Origen = comps[12],
        Destino = comps[13],
        Operador = comps[14],
        Grupo = comps[15]
    };
    muestras.Add(m);
}
return muestras;
}
private static List<Muestra> GetSubData(List<Muestra> data)
{
    return data.Where(d => d.Fecha >= new DateTime(2016, 10, 17, 8, 0, 0) &&
        d.Fecha <= new DateTime(2016, 10, 17, 20, 0, 0)).ToList();
}
private static Dictionary<string, List<Muestra>> GetMuestrasByEquipo(List<Muestra>
subdata)
{
    return subdata.GroupBy(d => d.Equipo).ToDictionary(d => d.Key, d => d.ToList());
}
private static Dictionary<string, double[]> LoadSpeeds(string filename)
{
    Dictionary<string, double[]> velocidades = new Dictionary<string, double[]>();

```

```

StreamReader sr = new StreamReader(filename);
sr.ReadLine();
while (!sr.EndOfStream)
{
    string l = sr.ReadLine();
    string[] c = l.Split(';');
    string equipo = c[0];
    double amAsc = double.Parse(c[2]);
    double amDesc = double.Parse(c[4]);
    double amPlain = double.Parse(c[6]);
    double amAscDesc = double.Parse(c[8]);
    double amDescDesc = double.Parse(c[10]);
    double amPlainDesc = double.Parse(c[12]);
    double asc = (amAsc + amAscDesc) / 2;
    double desc = (amDesc + amDescDesc) / 2;
    double plain = (amPlain + amPlainDesc) / 2;
    velocidades[equipo] = new double[] { asc, desc, plain };
}
return velocidades;
}
private static List<ICoord> GetPolyRajoSur()
{
    return new List<ICoord>()
    {
        new Punto() { X = 14142, Y = 109789},
        new Punto() { X = 14230, Y = 108257},
        new Punto() { X = 14985, Y = 107177},
        new Punto() { X = 15858, Y = 106212},
        new Punto() { X = 16792, Y = 105861},
        new Punto() { X = 17558, Y = 106806},
    }
}

```

```

        new Punto() { X = 17636, Y = 107615},
        new Punto() { X = 17368, Y = 108118},
        new Punto() { X = 16658, Y = 108830},
        new Punto() { X = 16337, Y = 109379},
        new Punto() { X = 15232, Y = 109833},
    };
}
private static List<ICoord> GetPolyRajoNorte()
{
    return new List<ICoord>()
    {
        new Punto() { X = 17932, Y = 114337 },
        new Punto() { X = 18469, Y = 114667 },
        new Punto() { X = 19197, Y = 114739 },
        new Punto() { X = 19784, Y = 114880 },
        new Punto() { X = 19759, Y = 114576 },
        new Punto() { X = 18491, Y = 113186 },
        new Punto() { X = 17675, Y = 113487 },
    };
}
private static List<ICoord> GetTerrainData(string filename)
{
    List<ICoord> retorno = new List<ICoord>();
    DxfDocument dxf = DxfDocument.Load(filename);
    foreach (Polyline p in dxf.Polylines)
        foreach (PolylineVertex pv in p.Vertexes)
            retorno.Add(pv.Location.ToICoord());
    return retorno;
}
#endregion

```

```

#region "Ouput - Generación archivos de salida"
private static void OutputDetenciones(Dictionary<string, Detencion>
detencionesEquipos,
    string filename)
{
    StreamWriter sw = new StreamWriter(filename);
    sw.WriteLine("Equipo;MinutesLeft;FechaHora;Origen;Destino;EnPala");
    foreach (string equipo in detencionesEquipos.Keys)
    {
        Detencion d = detencionesEquipos[equipo];

sw.WriteLine($"{d.Equipo};{d.MinutosRestantes};{d.HoraDetencion};{d.Origen};{d.Destin
o};{d.EnPala}");
    }

    sw.Close();
}
private static void OutputCircuitos(Dictionary<string, List<double[]>> circuitos,
    string filename)
{
    StreamWriter sw = new StreamWriter(filename);
    sw.WriteLine("Circuito;DistAvg;DistMean");
    foreach (string circuito in circuitos.Keys)
    {
        double[] dist = DoGetMeanAndAverage(circuitos[circuito].Select(d =>
d[0]).ToList());
        sw.WriteLine($"{circuito};{dist[0]};{dist[1]}");
    }
    sw.Close();
}
private static List<EquipoDetencion> OutputConsolidado(Dictionary<string, Detencion>
detenciones,

```

```

        Dictionary<string, List<double[]>> circuitos,
        Dictionary<string, double[]> velocidades,
        string filename)
    {
        StreamWriter sw = new StreamWriter(filename);

        sw.WriteLine("Equipo;Origen;Destino;Distancia;MinutosRestantes;VelocidadAscenso;VelocidadDescenso;VelocidadPlano;EstacionoPala");

        List<EquipoDetencion> equiposDetenciones = new List<EquipoDetencion>();
        foreach (string equipo in velocidades.Keys)
        {
            if (detenciones.ContainsKey(equipo))
            {
                Detencion d = detenciones[equipo];
                string key = $"{d.Origen}_{d.Destino}";
                if (circuitos.ContainsKey(key))
                {
                    List<double[]> circuito = circuitos[key];
                    double[] velocidad = velocidades[equipo];
                    double[] dist = DoGetMeanAndAverage(circuito.Select(c => c[0]).ToList());

                    sw.WriteLine($"{equipo};{d.Origen};{d.Destino};{dist[1]};{d.MinutosRestantes};{velocidad[0]};{velocidad[1]};{velocidad[2]};{d.EnPala}");

                    equiposDetenciones.Add(new EquipoDetencion(equipo, d, dist[1], velocidad));
                }
            }
        }
        sw.Close();
        return equiposDetenciones;
    }

    private static void OutputDetencionesOptimizado(Dictionary<string, Detencion> detenciones,

```

```

        Dictionary<string, double[]> circuitosExtra,
        string filename)
    {
        StreamWriter sw3 = new StreamWriter(filename);
        sw3.WriteLine("Equipo;MinutesLeft;FechaHora;Origen;Destino;EnPala");

        foreach (string equipo in detenciones.Keys)
        {
            Detencion d = detenciones[equipo];
            if (circuitosExtra.ContainsKey(equipo))
                d.MinutosRestantes -= circuitosExtra[equipo][1];

sw3.WriteLine($"{d.Equipo};{d.MinutosRestantes};{d.HoraDetencion};{d.Origen};{d.Desti
no};{d.EnPala}");
        }
        sw3.Close();
    }
private static void OutputOptimizado(Dictionary<string, double[]> circuitosExtra,
        string filename)
    {
        StreamWriter sw2 = new StreamWriter(filename);
        sw2.WriteLine("Equipo;Ciclos;Tiempo");
        foreach (string equipo in circuitosExtra.Keys)
        {
            double[] e = circuitosExtra[equipo];
            sw2.WriteLine($"{equipo};{e[0]};{e[1]}");
        }
        sw2.Close();
    }
#endregion

```

```

#region "Obsoleto - No hay llamadas en el código actual"
private static Terrain DoInterpolate()
{
    Terrain t = new Terrain();
    List<ICoord> puntos = GetTerrainData(@"C:\Users\Marinela-
pc\Desktop\datadiego\terrain.dxf");
    t.MinX = puntos.Min(pt => pt.X);
    t.MinY = puntos.Min(pt => pt.Y);
    t.MaxX = puntos.Max(pt => pt.X);
    t.MaxY = puntos.Max(pt => pt.Y);
    double[][] refPoints = BuildRefPoints(puntos);
    List<Muestra> muestras = GetData(@"C:\Users\Marinela-
pc\Desktop\datadiego\data.csv");
    KDTree<Punto> arbol = new KDTree<Punto>(2);
    foreach (Punto pto in puntos)
    {
        arbol.AddPoint(new double[] { pto.X, pto.Y }, pto);
    }
    int rows = (int)((t.MaxX - t.MinX) / res);
    int cols = (int)((t.MaxY - t.MinY) / res);
    TerrainCell[,] terrain = new TerrainCell[rows, cols];
    Console.WriteLine("Starting Terrain Interpolation");
    for (int i = 0; i < rows; i++)
    {
        for (int j = 0; j < cols; j++)
        {
            terrain[i, j] = new TerrainCell() { X = t.MinX + i * res, Y = t.MinY + j * res, Slope
= 0 };
        }
    }
    // Terrain Interpolation

```

```

Parallel.For(0, rows, new ParallelOptions() { MaxDegreeOfParallelism =
Environment.ProcessorCount }, (i, state) =>
{
    for (int j = 0; j < cols; j++)
    {
        double x = t.MinX + i * res;
        double y = t.MinY + j * res;
        double[] center = new double[] { x, y };
        double[] uppk = new double[] { y + delta, y + delta };
        try
        {
            NearestNeighbour<Punto> search = arbol.NearestNeighbors(center, 200,
delta);

            double altura = CalculateHeight(x, y, search, p);
            terrain[i, j].Z = altura;
            terrain[i, j].Row = i;
            terrain[i, j].Col = j;
        }
        catch (Exception e)
        {
            Console.WriteLine("Oops");
        }
    }
});

// Slope Calculation

Parallel.For(1, rows - 1, new ParallelOptions() { MaxDegreeOfParallelism =
Environment.ProcessorCount }, (i, state) =>
{
    for (int j = 1; j < cols - 1; j++)
    {
        if (terrain[i - 1, j - 1].Z > 0 && terrain[i - 1, j].Z > 0 && terrain[i - 1, j + 1].Z > 0

```

```

        && terrain[i, j - 1].Z > 0 && terrain[i, j + 1].Z > 0 && terrain[i + 1, j - 1].Z > 0
        && terrain[i + 1, j].Z > 0 && terrain[i + 1, j + 1].Z > 0)
    {
        double sEast = terrain[i - 1, j - 1].Z + 2 * terrain[i, j - 1].Z + terrain[i + 1, j -
1].Z;

        double sWest = terrain[i - 1, j + 1].Z + 2 * terrain[i, j + 1].Z + terrain[i + 1, j +
1].Z;

        double sew = (sEast - sWest) / (8 * res);

        double sNorth = terrain[i - 1, j - 1].Z + 2 * terrain[i - 1, j].Z + terrain[i - 1, j +
1].Z;

        double sSouth = terrain[i + 1, j - 1].Z + 2 * terrain[i + 1, j].Z + terrain[i + 1, j +
1].Z;

        double sns = (sNorth - sSouth) / (8 * res);

        double slope = Math.Sqrt(Math.Pow(sew, 2) + Math.Pow(sns, 2)) * 100;
        terrain[i, j].Slope = slope;
    }
}
});
t.Celdas = terrain;
Console.WriteLine("End of Interpolation");
Console.WriteLine("Saving Terrain");
// Terrain Save
using (StreamWriter sw = new
StreamWriter(@"C:\Users\DarkAlien\Desktop\datadiego\terrain.csv"))
{
    foreach (TerrainCell tc in terrain)
    {
        sw.WriteLine($"{tc.X};{tc.Y};{tc.Z};{tc.Slope};{tc.Row};{tc.Col}");
    }
    sw.Close();
}
Console.WriteLine("Terrain Saved");

```

```

    return t;
}
private static double[][] BuildRefPoints(List<ICoord> puntos)
{
    double[][] retorno = new double[puntos.Count][];
    for (int i = 0; i < puntos.Count; i++)
    {
        ICoord p = puntos[i];
        retorno[i] = new double[] { p.X, p.Y };
    }
    return retorno;
}
private static double CalculateHeight(double x, double y, NearestNeighbour<Punto>
puntos, double p)
{
    double retorno = 0;
    double pTotal = 0;
    foreach (Punto pto in puntos)
    {
        double distancia = Math.Sqrt(Math.Pow(pto.X - x, 2) + Math.Pow(pto.Y - y, 2));
        double ponderado = Math.Pow(distancia, -16);
        pTotal += ponderado;
        retorno += pto.Z * ponderado;
    }
    retorno = retorno * (1 / pTotal);
    if (double.IsNaN(retorno))
        retorno = 0;
    else
        retorno = (double)Math.Round(retorno * 100) / 100;
    return retorno;
}

```

```

    }
    private static Terrain DoLoad()
    {
        Terrain t = new Terrain();
        StreamReader sr = new StreamReader(@"C:\Users\Marinela-
pc\Desktop\datadiego\terrain.csv");
        List<TerrainCell> cells = new List<TerrainCell>();
        double minX = double.MaxValue;
        double maxX = double.MinValue;
        double minY = double.MaxValue;
        double maxY = double.MinValue;
        int cols = int.MinValue;
        int rows = int.MinValue;
        while (!sr.EndOfStream)
        {
            string l = sr.ReadLine();
            string[] p = l.Split(';');
            double x = double.Parse(p[0]);
            double y = double.Parse(p[1]);
            if (x < minX)
                minX = x;
            if (x > maxX)
                maxX = x;
            if (y < minY)
                minY = y;
            if (y > maxY)
                maxY = y;
            int row = int.Parse(p[4]);
            int col = int.Parse(p[5]);
            if (row > rows)

```

```

        rows = row;
    if (col > cols)
        cols = col;
}
sr.Close();
TerrainCell[,] terrain = new TerrainCell[rows + 1, cols + 1];
sr = new StreamReader(@"C:\Users\Marinela-pc\Desktop\datadiego\terrain.csv");
while (!sr.EndOfStream)
{
    string l = sr.ReadLine();
    string[] p = l.Split(';');
    double x = double.Parse(p[0]);
    double y = double.Parse(p[1]);
    double z = double.Parse(p[2]);
    double slope = double.Parse(p[3]);
    int row = int.Parse(p[4]);
    int col = int.Parse(p[5]);
    TerrainCell cell = new TerrainCell()
    {
        Row = row,
        Col = col,
        X = x,
        Y = y,
        Z = z,
        Slope = slope
    };
    terrain[row, col] = cell;
}
sr.Close();
t.Celdas = terrain;

```

```

    t.MinX = minX;
    t.MinY = minY;
    t.MaxX = maxX;
    t.MaxY = maxY;
    return t;
}

private static void AnalizeSpeeds(Terrain t, Dictionary<string, List<Muestra>>
muestrasByEquipo)
{
    Dictionary<string, EquipoVelocidad> velocidadesEquipos = new Dictionary<string,
EquipoVelocidad>();
    foreach (string equipo in muestrasByEquipo.Keys)
    {
        List<Muestra> muestras = muestrasByEquipo[equipo].OrderBy(d =>
d.Fecha).ToList();
        for (int i = 1; i < muestras.Count; i++)
        {
            Muestra m1 = muestras[i - 1];
            Muestra m2 = muestras[i];
            bool cargado = false;
            bool descargado = false;
            if (m1.Cargado && m2.Cargado)
                cargado = true;
            else if (!m1.Cargado && !m2.Cargado)
                descargado = true;
            double seconds = (m2.Fecha - m1.Fecha).TotalSeconds;
            // Comparamos el tiempo que transcurrio entre dos muestras.
            if (seconds <= 30 && (cargado || descargado))
            {
                if (m1.X >= t.MinX && m1.X <= t.MaxX && m1.Y >= t.MinY && m1.Y <=
t.MaxY &&

```

```

t.MaxY)          m2.X >= t.MinX && m2.X <= t.MaxX && m2.Y >= t.MinY && m2.Y <=
{
    int r1 = ((int)m1.X - (int)t.MinX) / res;
    int c1 = ((int)m1.Y - (int)t.MinY) / res;
    int r2 = ((int)m2.X - (int)t.MinX) / res;
    int c2 = ((int)m2.Y - (int)t.MinY) / res;
    TerrainCell b1 = t.Celdas[r1, c1];
    TerrainCell b2 = t.Celdas[r2, c2];

    // Debemos revisar que ambas celdas tengan componente Z para realizar el
    calculo de la velocidad
    if (b1.Z > 0 && b2.Z > 0)
    {
        // Revisamos adicionalmente que el punto pertenezca a una region valida
        dentro de la mina.
        if (b1.Slope <= 15 && b2.Slope <= 15)
        {
            double dx = Math.Pow(b1.X - b2.X, 2);
            double dy = Math.Pow(b1.Y - b2.Y, 2);
            double dz = Math.Pow(b1.Z - b2.Z, 2);
            double hours = seconds / 3600;
            double hip = Math.Sqrt(dx + dy + dz);
            double dp = Math.Sqrt(dx + dy);
            double alpha = Math.Acos(dp / hip) * (180.0 / Math.PI);

            double velocidad = (hip / 1000) / hours;
            // Nos estamos desplazando en el plano.
            if (alpha < 5)
            {
                if (!velocidadesEquipos.Keys.Contains(equipo))
                {

```

```

        velocidadesEquipos[equipo] = new EquipoVelocidad(equipo);
    }
    if (cargado)
        velocidadesEquipos[equipo].VelocidadPlano.Add(velocidad);
    else
        velocidadesEquipos[equipo].VelocidadPlanoDescargado.Add(velocidad);
    }
    else if (alpha >= 5 && alpha < 15 && b1.Z < b2.Z)
    {
        if (!velocidadesEquipos.Keys.Contains(equipo))
        {
            velocidadesEquipos[equipo] = new EquipoVelocidad(equipo);
        }
        if (cargado)
            velocidadesEquipos[equipo].VelocidadAscenso.Add(velocidad);
        else
            velocidadesEquipos[equipo].VelocidadAscensoDescargado.Add(velocidad);
        }
    else if (alpha >= 5 && alpha < 15 && b1.Z > b2.Z)
    {
        if (!velocidadesEquipos.Keys.Contains(equipo))
        {
            velocidadesEquipos[equipo] = new EquipoVelocidad(equipo);
        }
        if (cargado)
            velocidadesEquipos[equipo].VelocidadDescenso.Add(velocidad);
        else
            velocidadesEquipos[equipo].VelocidadDescensoDescargado.Add(velocidad);
        }
    }
}

```

```

    }
    }
    }
    }
}

StreamWriter sw = new StreamWriter(@"C:\Users\Marinela-
pc\Desktop\datadiego\velocidades.csv");

```

```

sw.WriteLine("Equipo;AscAvg;AscMean;DescAvg;DescMean;PlainAvg;PlainMean;AscDes
cAvg;AscDescMean;DescDescAvg;DescDescMean;PlainDescAvg;PlainDescMean");

```

```

    foreach (string equipo in velocidadesEquipos.Keys)
    {
        double[] amAsc =
DoGetMeanAndAverage(velocidadesEquipos[equipo].VelocidadAscenso);

        double[] amDesc =
DoGetMeanAndAverage(velocidadesEquipos[equipo].VelocidadDescenso);

        double[] amPlain =
DoGetMeanAndAverage(velocidadesEquipos[equipo].VelocidadPlano);

        double[] amAscDesc =
DoGetMeanAndAverage(velocidadesEquipos[equipo].VelocidadAscensoDescargado);

        double[] amDescDesc =
DoGetMeanAndAverage(velocidadesEquipos[equipo].VelocidadDescensoDescargado);

        double[] amPlainDesc =
DoGetMeanAndAverage(velocidadesEquipos[equipo].VelocidadPlanoDescargado);

```

```

sw.WriteLine($"{ equipo};{ amAsc[0]};{ amAsc[1]};{ amDesc[0]};{ amDesc[1]};{ amPlain[0] }
;{ amPlain[1]};{ amAscDesc[0]};{ amAscDesc[1]};{ amDescDesc[0]};{ amDescDesc[1]};{ am
PlainDesc[0]};{ amPlainDesc[1]}");

```

```

    }
    sw.Close();
}

private static void DoChartTerrain(Terrain t)
{
    DxfDocument dxf = new DxfDocument();

```

```

foreach (TerrainCell c in t.Celdas)
{
    if (c.Z != 0)
    {
        Vector3 v1 = new Vector3(c.X, c.Y, c.Z);
        Vector3 v2 = new Vector3(c.X + res, c.Y, c.Z);
        Vector3 v3 = new Vector3(c.X + res, c.Y + res, c.Z);
        Vector3 v4 = new Vector3(c.X, c.Y + res, c.Z);
        Face3d f = new Face3d(v1, v2, v3, v4);
        if (c.Slope < 5)
            f.Color = AciColor.Green;
        if (c.Slope >= 5 && c.Slope < 15)
            f.Color = AciColor.Yellow;
        if (c.Slope >= 15)
            f.Color = AciColor.Red;
        dxf.AddEntity(f);
    }
}
dxf.Save(@"C:\Users\DarkAlien\Desktop\datadiego\t3d.dxf");
}
public static Func<double[], double[], double> L2Norm_Squared_Double = (x, y) =>
{
    double dist = Math.Sqrt(Math.Pow(x[0] - y[0], 2) + Math.Pow(x[1] - y[1], 2));
    return dist;
};
#endregion
}
}

```