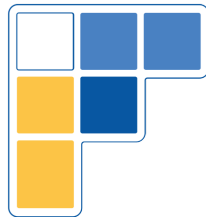


# UNIVERSIDAD DE TARAPACÁ



## FACULTAD DE INGENIERÍA



## DEPARTAMENTO DE INGENIERÍA EN COMPUTACIÓN E INFORMÁTICA



### Proyecto

“Modelo de predicción en términos productivos  
y de seguridad, mejores líderes de operación”

**Alumno(s):** Scarlet Gavia Mondaca

**Asignatura:** Proyecto IV

**Profesor:** Diego Aracena Pizarro

Arica, 2 de Diciembre 2024



## Índice

<b>1. Introducción</b>	<b>5</b>
<b>2. Descripción de la empresa</b>	<b>6</b>
<b>3. Definición del proyecto</b>	<b>7</b>
3.1. Contexto	7
3.2. Problema	7
3.3. Solución	7
<b>4. Objetivos</b>	<b>8</b>
<b>4.1. Objetivo general</b>	<b>8</b>
<b>4.2. Objetivo específicos</b>	<b>8</b>
<b>5. Requisitos del sistema</b>	<b>9</b>
<b>6. Planificación del proyecto</b>	<b>11</b>
6.1. Metodología	11
<b>7. Carta gantt</b>	<b>12</b>
<b>8. Modelo de contexto</b>	<b>13</b>
<b>9. Diagrama de caso de uso</b>	<b>14</b>
<b>10. Herramientas a utilizar en la implementación</b>	<b>15</b>
<b>11. Alcance del producto a desarrollar</b>	<b>17</b>
<b>12. Modelo BPMN</b>	<b>17</b>
<b>13. Implementación del sistema</b>	<b>18</b>
<b>14. Pruebas del sistema</b>	<b>20</b>
<b>Figura 12: Entrenamiento 2</b>	<b>21</b>
<b>15. Análisis de resultado</b>	<b>22</b>
<b>16. Conclusión</b>	<b>23</b>



## Índice de tablas

Tabla 1: Tabla de requisitos de alto nivel.....	9
Tabla 2: Tabla de requisitos funcionales.....	9
Tabla 3: Tabla de requisitos no funcionales.....	10



## Índice de Figuras

Figura 1: Logo Codelco.....	5
Figura 2: Logo Codelco División Ministro Hales.....	5
Figura 3: Proceso de scrum.....	11
Figura 4: Carta gantt.....	12
Figura 5: Modelo de contexto.....	13
Figura 6: Diagrama de caso de uso.....	14
Figura 7: datos nulos de la base de datos.....	15
Figura 8: datos de fatiga y cabeceo.....	16
Figura 9: modelo BPMN.....	17



## 1. *Introducción*

Actualmente en el rubro minero del cobre, el proceso de transporte es el que representa el mayor porcentaje del costo unitario Mina, por esta razón es fundamental optimizar el uso eficiente de los camiones de extracción (CAEX) manteniendo los altos estándares de seguridad que se exige al operador en esta actividad. Para lo anterior, en Codelco División Ministro Hales, se cuenta con tecnología instalada para optimizar el uso de los activos críticos así como para el cuidado en la operación de los trabajadores que tripulan estos equipos mina.

En este proyecto se plantea diseñar un modelo que permita proponer a aquellos operadores de camión de extracción mejor calificado en términos de seguridad y productividad de la mina, según el día de turno laboral. Para esto el cliente, entregará datos sobre fatiga de los trabajadores en la operación de CAEX e información productiva de ellos, en el uso de estos equipos



---

## 2. *Descripción de la empresa*

Codelco división ministro Hales es una de las divisiones de la Corporación Nacional del Cobre de Chile (Codelco), la principal productora de cobre del mundo, esta se encuentra en la Región de Antofagasta, al norte de Chile, cerca de la ciudad de Calama, la división inició sus operaciones el año 2013, siendo una de las más recientes de la compañía, Ministro Hales se caracteriza por la explotación de un yacimiento de cobre y plata mediante minería a rajo abierto, su producción anual alcanza alrededor de 200.000 toneladas de cobre fino, además de subproductos como plata y ácido sulfúrico, generado a partir del tratamiento de los gases producidos por la fundición de minerales sulfurados, esta división es conocida por su enfoque en la innovación y la sostenibilidad, implementando tecnologías modernas para reducir su impacto ambiental y optimizar los procesos productivos, la minería en Ministro Hales es una pieza clave dentro de la estrategia de Codelco para mantener su liderazgo en la industria global del cobre.



### **3. Definición del proyecto**

#### **3.1. Contexto**

La minera Ministro Hales opera con dos turnos de trabajadores: uno diurno y otro nocturno, por lo tanto trabaja con cuatro bloques de empleados, dado que algunos trabajadores provienen de otras ciudades, pueden experimentar cansancio después de la colación y algunos de ellos se manejan mejor con ciertas maquinarias, la empresa busca desarrollar un modelo que prediga, en términos de producción y seguridad, cuáles son los candidatos más idóneos para continuar como operadores, el objetivo de este modelo es minimizar el cansancio y proteger la salud de los trabajadores.

La empresa dispone de información sobre los trabajadores que incluye su nivel de cansancio, su desempeño en el manejo de maquinarias y sus horarios de turno.

#### **3.2. Problema**

La División de Codelco Ministro Hales, cuenta con información de producción de los operadores, con esta información, se busca evaluar a sus trabajadores según sus turnos, maquinaria y nivel de cansancio, para realizar el trabajo durante el turno, tomando en cuenta sus horario de colación.

#### **3.3. Solución**

Se sugiere un modelo de predicción el cual muestra la lista de un grupo o un operador seleccionado por el cliente, esta lista nos mostrará el identificador del operador con las posibles alertas que podría generar durante el turno actual de trabajo, este modelo se realizará estudiando la estadística de las alertas generadas, según horarios y operadores, esto con el fin de conocer qué alertas podría generar cada operador o grupo de operadores para los diferentes turnos de la empresa.



---

## 4. *Objetivos*

### 4.1. *Objetivo general*

Realizar un modelo de predicción para la minera Ministro Hales, desde una fuente de datos generada por distintos sistemas de alerta.

### 4.2. *Objetivo específicos*

- Estudio de la fuente de datos.
- Realizar algoritmos para la limpieza de la fuente de datos.
- Agregar datos a la fuente de datos.
- Análisis de datos.
- Crear algoritmos de predicción.
- Realizar pruebas sobre el algoritmo.





## 5. Requisitos del sistema

### 5.1. Requisitos de alto nivel

Identificador	Descripción
RAN1	El sistema debe ser capaz de predecir la calidad del trabajo de los operadores en función de datos.

Tabla 1: Tabla de requisitos de alto nivel

### 5.2. Requisitos funcionales

Identificador	Descripción
RF1	El modelo debe permitir actualizar o agregar información de los trabajadores.
RF2	El modelo debe permitir seleccionar el turno en el que se está produciendo.
RF3	El modelo debe permitir seleccionar la maquinaria en la cual se necesita operador.
RF4	El modelo debe permitir seleccionar el horario en el que se necesita operador.
RF5	El modelo debe mostrar una lista de los trabajadores más calificados para el turno y maquinaria especificados.
RF6	El modelo debe alertar sobre el cansancio de los trabajadores.
RF7	El modelo debe contar con un algoritmo de predicción, el cual utilizará los datos capturados.
RF8	El modelo debe generar una alerta en caso de que el operador ponga en riesgo de bajar su calidad de trabajo.
RF9	El sistema debe integrarse con sistemas existentes.

Tabla 2: Tabla de requisitos funcionales



### 5.3. *Requisitos no funcionales*

Identificador	Descripción
1	El modelo debe tener la medidas necesarias de seguridad para que no se filtren los datos de los trabajadores.
2	El modelo debe estar disponible para su uso en cualquier momento.
3	El modelo debe ser confiable y cumplir con los requisitos del usuario.
4	El modelo debe ser fácil de mantener y actualizar.

Tabla 3: Tabla de requisitos no funcionales.

## 6. Planificación del proyecto

### 6.1. Metodología

Se utilizará una metodología ágil llamada Scrum, la cual consiste en la comunicación constante entre el cliente y el equipo de desarrollo, esto nos permite crear prototipos tempranos y reducir riesgos de malentendidos en los requisitos del cliente. Se implementa Scrum mediante reuniones semanales online donde se revisa las tareas complementadas y pendientes.

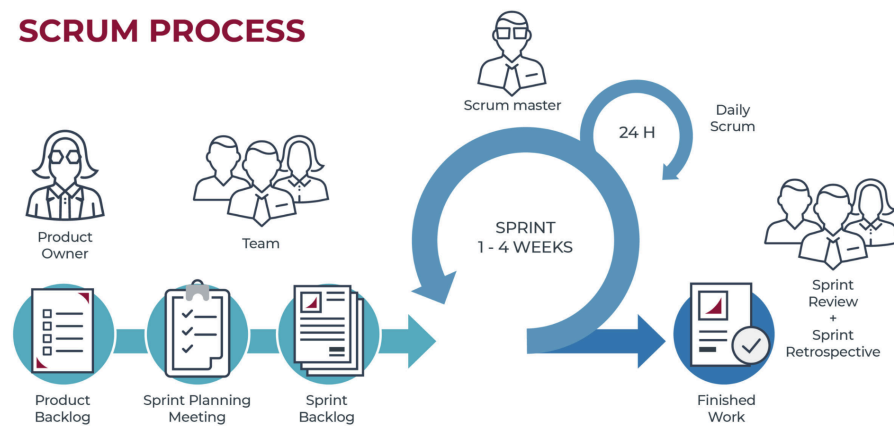


Figura 3: Proceso de scrum



## 7. Carta gantt

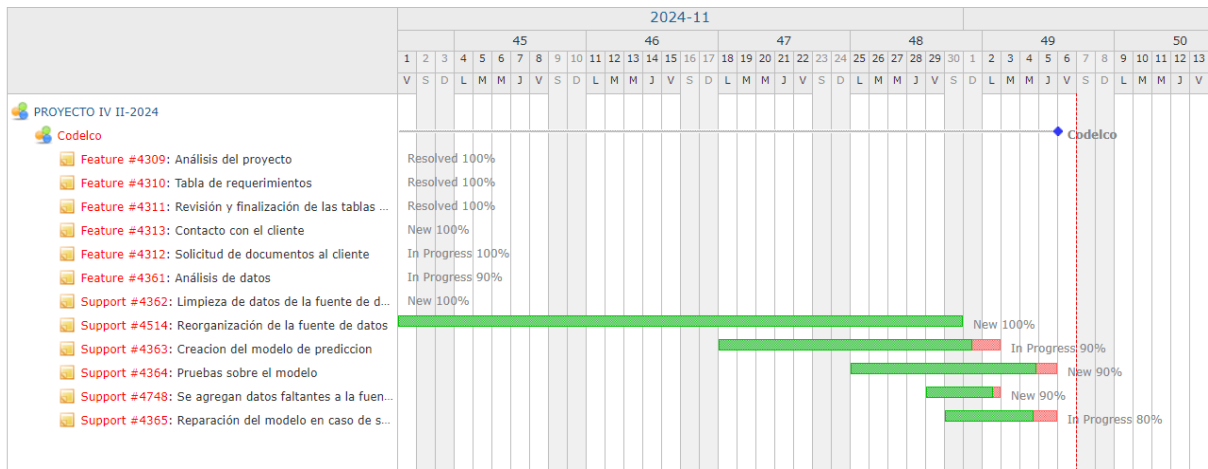


Figura 4: Carta gantt

## 8. Modelo de contexto

En el siguiente diagrama se explicará el funcionamiento del sistema, se obtiene la fuente de datos, se realizan los análisis correspondientes, se pasan los datos por el modelo de predicción y se genera la alerta para el operador.

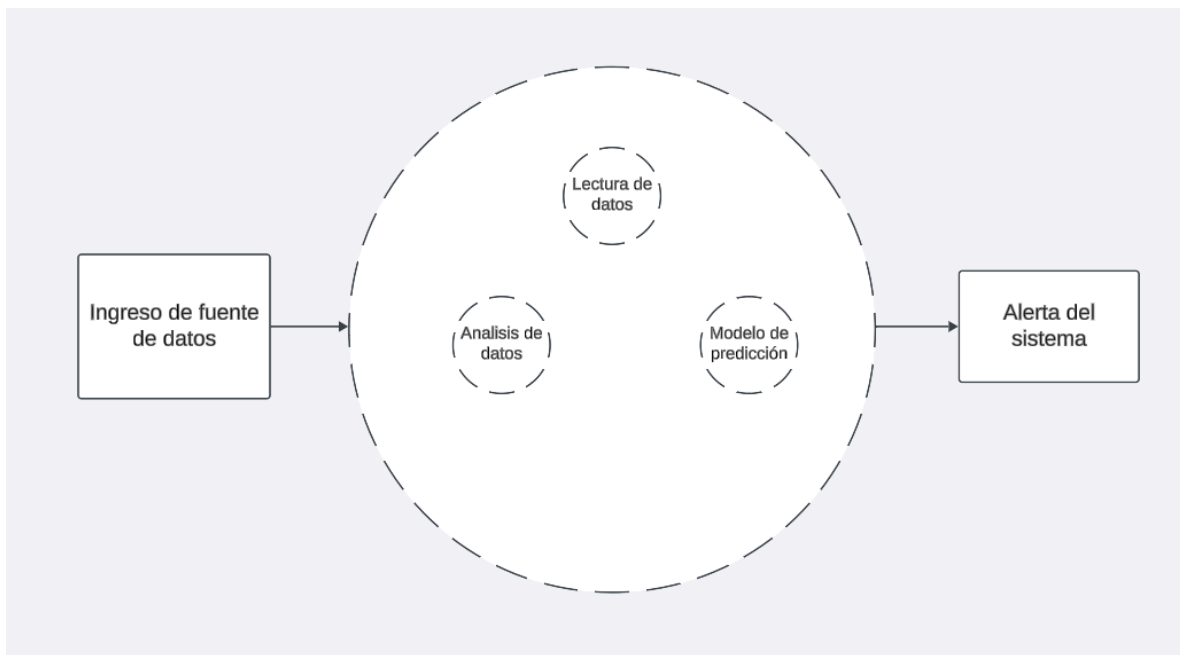


Figura 5: Modelo de contexto

## 9. Diagrama de caso de uso

El siguiente diagrama, presenta el funcionamiento del modelo, se obtienen datos desde una fuente de datos, los cuales serán leídos y analizados por el sistema, además tenemos al jefe de turno, quien es el que ingresa los datos necesarios para generar al operador mejor calificado para el turno.

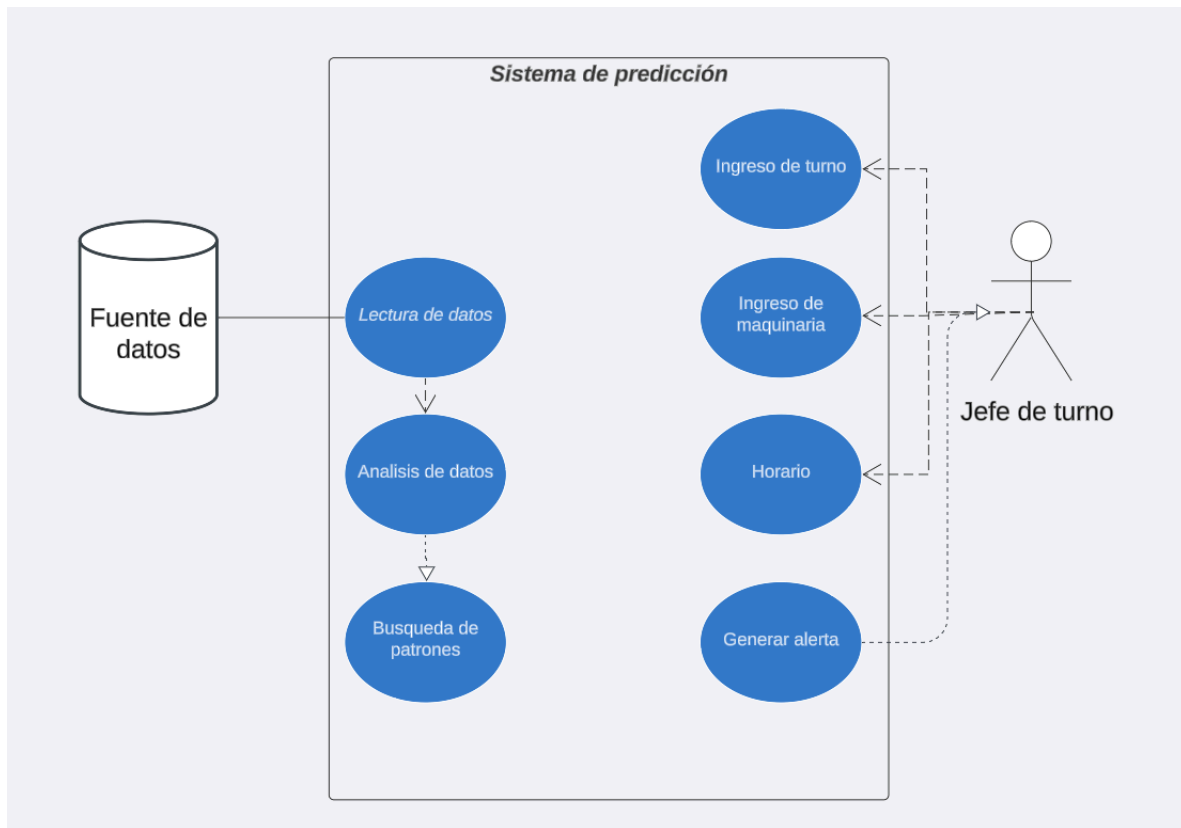


Figura 6: Diagrama de caso de uso.



## 10. Herramientas a utilizar en la implementación

Para la limpieza de datos se utilizará Pandas, con esta herramienta también realizaremos el análisis de los datos, mediante tablas estadísticas y podremos eliminar los datos nulos dentro de la base de datos entregada por el cliente, con esta misma herramienta uniremos los datos de las distintas fuentes de datos.

En la siguiente imagen se puede ver la cantidad de operadores nulos que posee la base de datos, estos datos corresponden a los identificadores de los trabajadores.

```
[ ] #Imprimir valores nulos
print(julio.isnull().sum())
```

```
↔ created_at      0
   equipment_name  0
   eve            0
   operator_id    25390
   hutc          0
   futc          0
   dtype: int64
```

```
[ ] import pandas as pd

filas_con_nulos = julio[julio['operator_id'].isnull()]

# Esto te devuelve las filas donde la columna 'columna' tiene valores nulos
print(filas_con_nulos)
```

```
↔
```

	created_at	equipment_name	eve	operator_id	hutc	futc
12	01-07-2024 8:35	C332	NPD	NaN	123551	10724
30	01-07-2024 8:36	C332	NPD	NaN	123655	10724
50	01-07-2024 8:37	C332	NPD	NaN	123759	10724
71	01-07-2024 8:39	C332	NPD	NaN	123903	10724
86	01-07-2024 8:39	C332	OFF	NaN	123949	10724
...	...	...	...	...	...	...
751828	17-07-2024 9:45	C349	NPD	NaN	134302	170724
751829	17-07-2024 9:45	C349	NPD	NaN	134406	170724
751830	17-07-2024 9:45	C349	NPD	NaN	134510	170724
751851	17-07-2024 9:46	C349	NPD	NaN	134614	170724
751869	17-07-2024 9:47	C349	NPD	NaN	134718	170724

```
[25390 rows x 6 columns]
```

Figura 7: datos nulos de la base de datos



Se analizan también los datos de fatiga, cabeceo y alerta de ambos casos que se produce en los trabajadores, dentro de la fuente de datos, esto se puede ver en la figura n° 8.

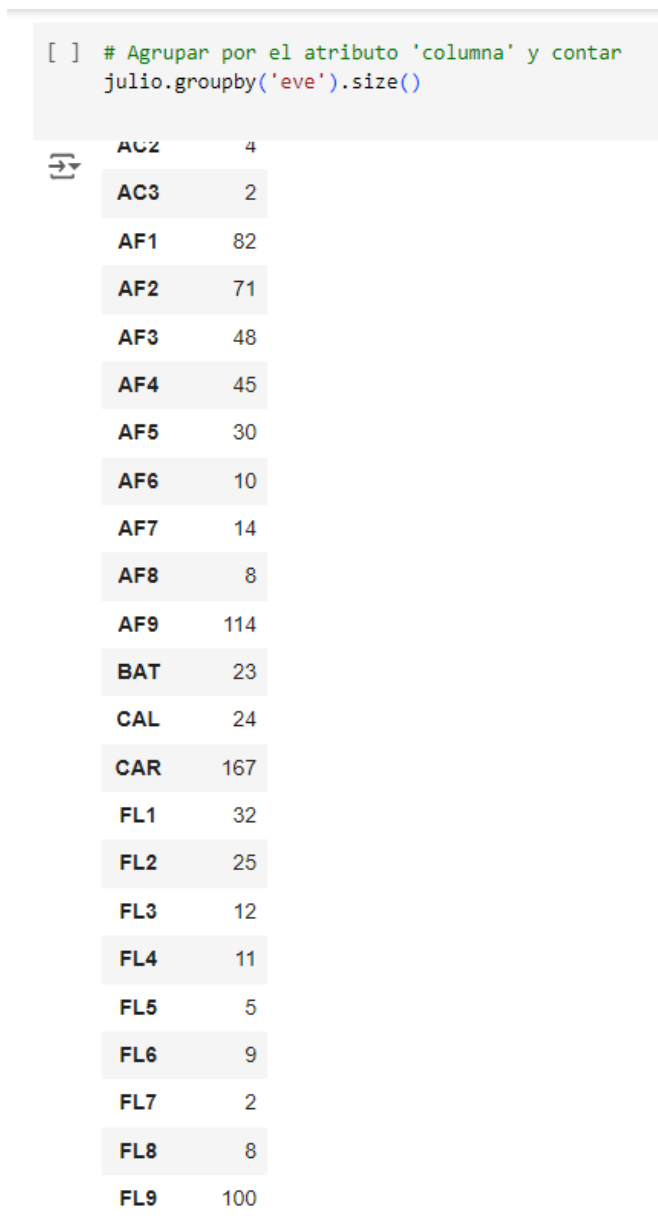


Figura 8: datos de fatiga y cabeceo.



## 11. Alcance del producto a desarrollar

El alcance del proyecto se inicia con la predicción de los meses de junio, julio y agosto, para luego ser utilizado con los datos de los siguientes meses, con el fin de ser utilizado con los datos necesarios del mes correspondiente, lo ideal para el alcance del producto es iniciar en la división de Codelco Ministro Hales, pero podría llegar a las demás mineras para cuidar a los trabajadores.

## 12. Modelo BPMN

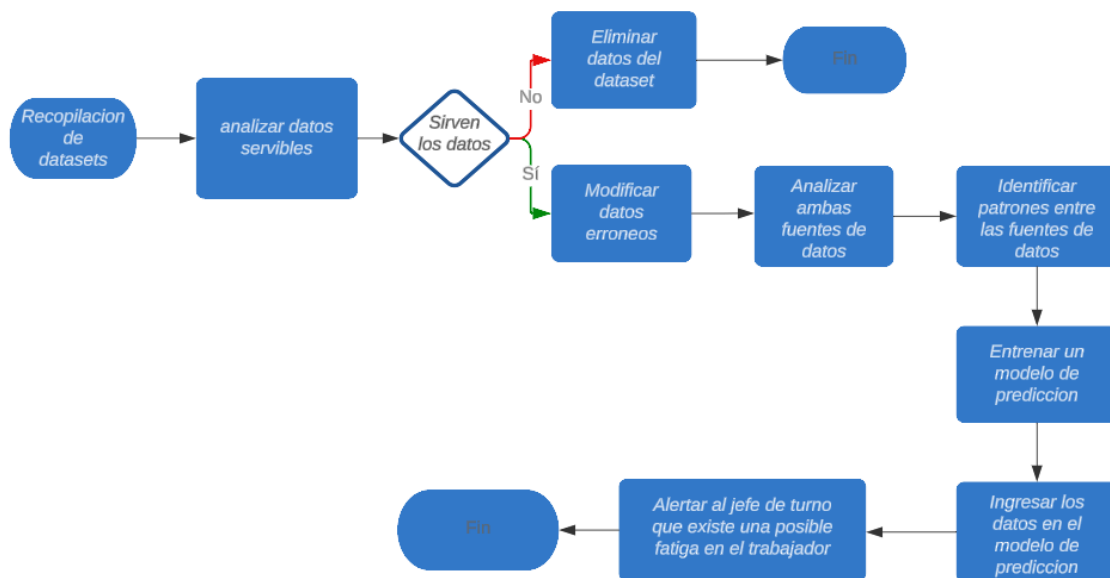


Figura 9: modelo BPMN



### 13. *Implementación del sistema*

Se crea un modelo de predicción, este modelo se entrena cada vez que el cliente requiere cambiar los datos de entrenamiento, para esto el cliente debe tener la fuente de datos de la cual desea realizar la predicción, el modelo leera la fuente de datos y realiza la predicción, luego se mostrará una interfaz básica que permita al cliente elegir si desea la predicción de un operador en específico o si desea la predicción de un grupo completo, para realizar esta predicción, el modelo trabaja con la hora actual, es decir el modelo obtiene la hora en el momento de hacer la predicción.

Para realizar la predicción el modelo necesitará que la fuente de datos contenga las comunas:

- operator\_id:  
Columna que posee de forma numérica el identificador del trabajador.
- hora:  
Columna que posee la hora en formato hh:mm, esta es la hora en la que se genera la alerta en el sistema.
- Grupo:  
Columna en la cual se guardará el grupo de cada trabajador, esta se utilizará en formato MH-x, donde la x es el valor correspondiente al grupo.
- hutc:  
En esta columna se obtienen coordenadas de ubicación.
- futc:  
En esta columna se obtienen coordenadas de ubicación.
- eve:  
Esta columna corresponde a la alerta generada por el sistema.
- turno:  
Columna que posee el turno en el que se generó la alerta del trabajador.

El sistema permite que se suba una fuente de datos para el entrenamiento del modelo, una vez que el modelo reconozca la fuente de datos, se entrena el modelo, y guardará el modelo de predicción para ser utilizado con distintas consultas.

```
# Carga de La fuente de datos en formato CSV
@st.cache_data
def cargar_datos(ruta_csv):
    try:
        df = pd.read_csv(ruta_csv)
        df['created_at'] = pd.to_datetime(df['created_at'])
        df['hora'] = df['created_at'].dt.hour
        return df
    except Exception as e:
        st.error(f"Error al cargar el archivo CSV: {e}")
        return None
```

Figura 10: Carga de archivo CSV.

```
# Entrenar el modelo con todos los datos disponibles
@st.cache_resource
def entrenar_modelo(df):
    X = df[['hora', 'turno', 'hutc', 'futc', 'operator_id', 'Grupo']]
    X = pd.get_dummies(X)
    y = df['eve']
    modelo = RandomForestClassifier(n_estimators=100, random_state=42, class_weight='balanced')
    modelo.fit(X, y)
    return modelo, X.columns
```

Figura 11: Modelo entrenado.



Luego se realizan las vistas para 2 consultas diferentes, siendo estas consultas por el grupo o el identificador del operador.

```
menu = st.sidebar.selectbox("Menú", ["Inicio", "Predicción por ID", "Predicción por Grupo"])

if menu == "Inicio":
    st.write("Bienvenido al sistema de predicción.")
    st.write("Ingrese una fuente de datos para realizar el entrenamiento.")

elif menu == "Predicción por ID":
    st.header("Predicción por ID")
    operador_id = st.number_input("Ingrese el ID del operador", min_value=0, step=1)
    if st.button("Realizar Predicción"):
        prediccion, error = predecir_por_id(modelo, columnas, df, operador_id)
        if error:
            st.error(error)
        else:
            st.success(f"La predicción para el operador {operador_id} es: {prediccion}")

elif menu == "Predicción por Grupo":
    st.header("Predicción por Grupo")
    grupo = st.selectbox("Seleccione un grupo", ["MH-1", "MH-2", "MH-3", "MH-4"])
    if st.button("Realizar Predicción para el Grupo"):
        resultados, error = predecir_por_grupo(modelo, columnas, df, grupo)
        if error:
            st.error(error)
        else:
            st.success(f"Predicciones para el grupo {grupo}:")
            st.table(resultados)
```

Figura 12: Vista para la selección de predicción.

Una vez ingresados los valores para realizar la solicitud, el modelo utilizará la hora actual para calcular el turno del trabajador y utilizará estos datos para mostrarnos el resultado obtenido en la predicción.

```
# Realizar La predicción por grupo
def predecir_por_grupo(modelo, columnas, df, grupo):
    try:
        datos_grupo = df[df['Grupo'] == grupo]
        if datos_grupo.empty:
            return None, "Grupo no encontrado."

        hora_actual = datetime.now().hour # Se solicita La hora actual para realizar La predicción en el horario actual
        turno_codificado = {'Día': 1, 'Noche': 0} if 8 <= hora_actual <= 19 else {'Día': 0, 'Noche': 1}
        #entrada de datos por cada trabajador que pertenece al grupo
        entradas_grupo = []
        for operador_id in datos_grupo['operator_id'].unique():
            entrada = [hora_actual, turno_codificado['Día'], 0, 0, operador_id, grupo]
            entradas_grupo.append(entrada)

        entradas_grupo_df = pd.DataFrame(entradas_grupo, columns=['hora', 'turno', 'hutc', 'futc', 'operator_id', 'Grupo'])
        entradas_grupo_df = pd.get_dummies(entradas_grupo_df).reindex(columns=columnas, fill_value=0)

        #se buscan Las predicciones de cada trabajador perteneciente al grupo en el horario actual
        predicciones = modelo.predict(entradas_grupo_df)
        operadores = datos_grupo['operator_id'].unique()
        return list(zip(operadores, predicciones)), None
    except Exception as e:
        return None, str(e)
```

Figura 13: Recopilación de datos para resultado de grupo.

```
# Realizar La predicción por identificador del operador
def predecir_por_id(modelo, columnas, df, operador_id):
    try:
        datos_operador = df[df['operator_id'] == operador_id]
        if datos_operador.empty:
            return None, "Operador no encontrado."

        hora_actual = datetime.now().hour
        turno_codificado = {'Día': 1, 'Noche': 0} if 8 <= hora_actual <= 19 else {'Día': 0, 'Noche': 1}
        grupo = datos_operador['Grupo'].iloc[0]

        entrada = pd.DataFrame([[hora_actual, turno_codificado['Día'], 0, 0, operador_id, grupo]],
                               columns=['hora', 'turno', 'hutc', 'futc', 'operator_id', 'Grupo'])
        entrada = pd.get_dummies(entrada).reindex(columns=columnas, fill_value=0)
        #Se buscan Las predicciones del operador seleccionado.
        prediccion = modelo.predict(entrada)
        return prediccion[0], None
    except Exception as e:
        return None, f"Error en la predicción: {e}"
```

Figura 14: Recopilación de datos para resultados de operador.

## 14. Pruebas del sistema

Para visualizar el modelo creado, se realizó una interfaz simple, en la cual se cargará la fuente que se utilizará para entrenar el modelo (figura 15), una vez subido el documento, se crearán 2 opciones de predicción, ver la fatiga por el identificador y ver las predicciones de alertas por grupo en esta última el sistema mostrará una lista de todos los operadores pertenecientes al grupo con sus respectivas alertas.

### Sistema de Predicción de Eventos

Sube un archivo CSV

Drag and drop file here  
Limit 200MB per file • CSV

Browse files

Por favor, sube un archivo CSV.

Figura 15: Vista para subir la fuente de datos.

Para realizar las pruebas de este sistema se tomaron 2 fuentes de datos procesadas, una de estas fuentes de datos, contiene más datos que la otra, esto debido a que en la columna evento se tomaron 5 eventos, los cuales son:

ACx, AFx, FLx, SQx y ON\_, debido a que la predicción con esta fuente entregaba mayoritariamente el valor ON\_.



## Predicción por Grupo

Seleccione un grupo

MH-3

Realizar Predicción para el Grupo

Predicciones para el grupo MH-3:

	0	1
0	414	ON_
1	625	ON_
2	486	ON_
3	84	ON_
4	761	ON_
5	555	ON_
6	706	ON_
7	375	ON_
8	982	ON_
9	8933	ON_
10	768	ON_
11	227	ON_

Figura 16: Entrenamiento 1

Se utiliza la otra fuente de datos, la cual contiene los eventos ACx, AFx, FLx, y SQx, se realiza el entrenamiento del modelo con esta fuente de datos, ya se puede obtener los valores de alerta de fatiga.



## Predicción por Grupo

Seleccione un grupo

MH-3



Realizar Predicción para el Grupo

Predicciones para el grupo MH-3:

	0	1
0	486	AC1
1	555	AC1
2	375	SQ2
3	706	AC1
4	768	AC1
5	273	SQ2
6	47	SQ1
7	813	AC1
8	397	AC1
9	720	AC1
10	327	SQ2

Figura 17: Entrenamiento 2





## 15. *Análisis de resultado*

Al construir el modelo de predicción utilizando los datos considerados importantes, se observó que los resultados arrojados eran predominantemente el evento “ON\_”, Este valor es significativo porque indica que el operador está utilizando el dispositivo de monitoreo de fatiga de manera adecuada, esta alerta generaba un ruido constante en el modelo debido a su alta frecuencia de aparición, para mejorar la claridad de los resultados, se decidió excluir el evento “ON\_” de la columna “eve”. Este ajuste permitió obtener predicciones más precisas sobre las alertas relacionadas con los operadores.

Sin embargo, al concluir el proyecto, se detectaron algunas limitaciones en el modelo, en particular, se notó que no se había considerado el porcentaje de probabilidad de ocurrencia de cada alerta en los operadores, este aspecto es fundamental, ya que permite priorizar la atención a los operadores en función de su nivel de riesgo y garantizar un monitoreo más efectivo durante sus turnos, la inclusión de esta métrica podría mejorar significativamente la utilidad del modelo, reforzando su capacidad para proteger la seguridad y bienestar de los operadores en su jornada laboral.



---

## 16. Conclusión

El presente informe nos da a conocer sobre la creación de un modelo de predicción para la minera ministro Hales, tomando en cuenta los análisis de datos entregados, los cuales pertenecen a distintas fuentes de datos sobre los operadores de la empresa, sus turnos y grupos respectivos, estos datos fueron analizados y reorganizados con la herramienta pandas de python, se utilizó la librería matplotlib para realizar diagramas y entender de mejor manera los valores dentro de la fuente de datos, se utiliza de igual manera la librería "streamlit" para crear una interfaz simple para ver los resultados de la predicción.

En este proyecto se logra concluir con la idea básica que tiene el cliente, debido a que las fuentes de datos contenían pocos datos y por la carga académica del alumno.