

## **Mantenimiento Predictivo para Sistemas Avanzados de Ósmosis Inversa con Inteligencia Artificial**

**Nivel de Madurez:** TRL 2-3

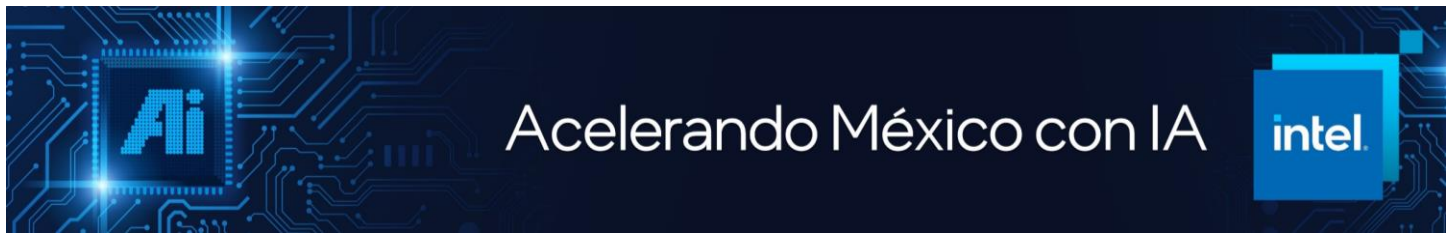
### **1. Generalidades**

#### ***Introducción***

Se estima que nuestro planeta contiene unos 1386 millones de km<sup>3</sup> de agua, cantidad la cual no ha disminuido ni aumentado en los últimos dos mil millones de años. Se calcula que el 97% de esa agua es salada y sólo el 2.5% se considera dulce. Asimismo la población de México ha aumentado de manera considerable a lo largo de los años, y se estima que a principios de 2024 era de 129 millones de habitantes. Esto plantea una problemática significativa en cuanto a la escasez de este recurso. Actualmente han emergido distintas soluciones que prometen revertir esta problemática, una de ellas es la Ósmosis Inversa, el cual es un proceso de purificación de agua que utiliza una membrana semipermeable para eliminar iones, moléculas y partículas más grandes, como sales y contaminantes. La Ósmosis Inversa se utiliza principalmente para desalinizar el agua de mar, transformándola en agua potable. Una de las principales desventajas de este proceso es el costo de las membranas semipermeables y su mantenimiento, problemática que se busca mejorar a través del desarrollo de un modelo de Inteligencia Artificial que permita predecir el desgaste y las irregularidades en el proceso, dando oportunidad para remplazar los componentes antes de que se dañen y así evitar fallas inminentes, tomando datos históricos del proceso como la presión del agua, el flujo de permeado, la temperatura y la salinidad en el agua.

#### ***1.1 Generalidades del Proyecto***

El proyecto tiene como objetivo principal desarrollar un sistema de Inteligencia Artificial para el mantenimiento predictivo de sistemas de Ósmosis Inversa, centrado en optimizar el uso de las membranas



semipermeables y predecir irregularidades en el proceso. El sistema se plantea como un producto orientado a ingenieros y profesionales en el área de tratamiento de agua, particularmente en la industria de desalinización. La escalabilidad del proyecto es significativa, ya que podría aplicarse no solo en plantas industriales, sino también en sistemas domésticos de purificación.

El impacto potencial de este proyecto podría transformar la gestión de los recursos hídricos tanto a nivel nacional como global, fortaleciendo la competitividad de la industria y contribuyendo a una mayor eficiencia en el uso del agua. Al posicionar al país como líder en soluciones sostenibles, el proyecto adquiere una relevancia particular en la gestión de un recurso crítico como el agua, especialmente en el contexto del cambio climático y el aumento de la demanda mundial.

### ***1.2 Oportunidad Detectada***

La necesidad principal que satisface este proyecto es la optimización del mantenimiento de membranas semipermeables, un componente esencial en la Ósmosis Inversa cuyo desgaste afecta tanto los costos como la eficiencia del sistema. Existen evidencias claras de la importancia de esta necesidad, pues el costo de las membranas y su reposición es elevado. El propósito del proyecto es evitar fallas inminentes mediante el monitoreo en tiempo real y la predicción del desgaste basado en datos históricos.

### ***Proceso de Ósmosis Inversa***

El proceso de Ósmosis Inversa tradicional se divide en tres etapas principales:

1. **Pretratamiento:** En esta fase, el agua salada pasa a través de filtros de sedimentos y filtros de carbón activado. Estos filtros eliminan partículas más grandes y contaminantes químicos del agua, preparándola para la siguiente etapa.



2. Filtración Principal: A continuación, el agua se dirige al sistema principal, que incluye una membrana semipermeable. Esta membrana actúa como una barrera selectiva, permitiendo el paso de ciertas moléculas y iones mientras restringe el tránsito de otros. La permeabilidad de la membrana se determina por factores como el tamaño molecular, la carga eléctrica y la polaridad de las sustancias.
3. Postratamiento: Finalmente, el agua puede someterse a un postratamiento, que se ajusta según el propósito específico para el que se destinará el agua.

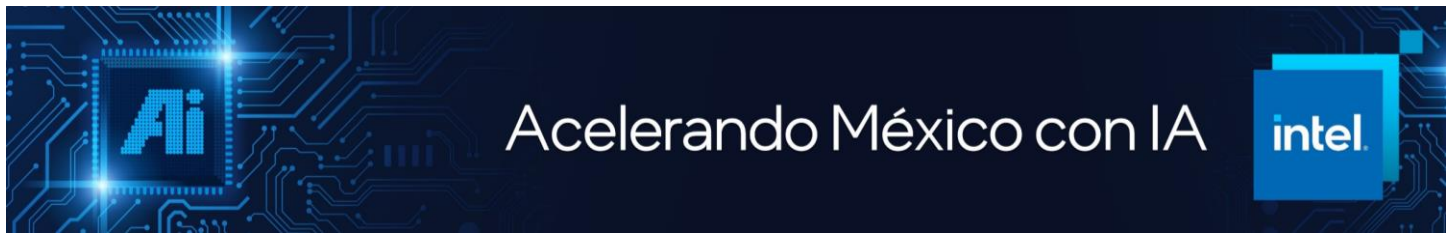
El sistema físico de Ósmosis Inversa está compuesto por diversos elementos, incluidos filtros de sedimentos, filtros de carbón activado, la membrana semipermeable, una bomba de presión, válvulas de control de flujo, tanques de almacenamiento, tubos conectores, medidores de presión y en algunos casos, filtros ultravioleta.

### ***1.3 Generalidades Técnicas***

La solución teórica de este proyecto utiliza una combinación de componentes de hardware y software, incluyendo sensores para medir parámetros como la presión, el flujo y la salinidad, junto a un modelo de Machine Learning desarrollado en *Python* utilizando la biblioteca *Scikit-Learn*. Incluye un sistema basado en IA que recopila datos del sistema de Ósmosis Inversa y los procesa para anticipar fallas. Los procesos utilizados para generar esta PoC incluyen técnicas de aprendizaje supervisado, escalado de datos y balanceo de clases para mejorar la precisión y evitar sesgos.

#### ***Pasos para avanzar al siguiente nivel de TRL***

- Realizar pruebas en entornos más amplios con datos reales del proceso.
- Optimizar el modelo para incluir nuevas variables y mejorar su precisión.
- Desarrollar una interfaz de usuario para facilitar la implementación del sistema en distintas industrias.



### ***1.4 Innovación con IA***

El uso de Inteligencia Artificial es crucial para este proyecto, ya que permite predecir el desgaste de las membranas en función de múltiples variables operativas. Utilizando técnicas de Machine Learning de nivel intermedio, se procesan datos históricos para generar predicciones precisas que optimizan la gestión del sistema de Ósmosis Inversa. En términos éticos, el uso de IA en este proyecto requiere asegurar la transparencia en los resultados. Es crucial prevenir sesgos en los algoritmos que puedan afectar la toma de decisiones, analizando los datos utilizados para asegurar que sean representativos. Además, se debe garantizar la privacidad de los datos de operación, implementando medidas de seguridad adecuadas y cumpliendo con las normativas de protección de datos.

### ***1.5 Equipo de Desarrollo de la PoC***

El equipo responsable del desarrollo de esta Prueba de Concepto (PoC) abarca diversas disciplinas fundamentales para su implementación exitosa. Mi nombre es Manuel de Jesús Cortez Villa, estudiante de Ingeniería en Software en ITSON, con experiencia en investigación científica y análisis de datos. Mi lenguaje de programación principal es *Python*, lo que me ha permitido abordar de manera eficaz los desafíos técnicos del proyecto.

El interés por el proceso de Ósmosis Inversa surgió a partir de la influencia de uno de mis compañeros, quien se encuentra en el proceso de titulación de Ingeniería Química. Su perspectiva me permitió comprender la importancia de la relación interdisciplinaria entre la Ingeniería Química y el Software en la optimización de procesos de purificación de agua.

La combinación de conocimientos en software y la comprensión de principios químicos ha sido fundamental para abordar los desafíos técnicos y analíticos del proyecto. Mi experiencia en análisis de datos y conocimiento en programación ha facilitado la interpretación de los resultados y la optimización de los algoritmos utilizados.

## *Recolección de Datos*

Para el desarrollo del modelo predictivo, se implementó un enfoque de aprendizaje supervisado, que utiliza un conjunto de 13 variables numéricas con el objetivo de predecir tanto el resultado final de la salinización del agua como el estado físico de las membranas semipermeables. Para ello, se generó un archivo de texto en formato CSV que contiene un total de 1,000 filas de datos, las cuales se derivaron de suposiciones fundamentadas sobre los resultados esperados.

El modelo se alimenta de datos operacionales del sistema, los cuales son esenciales para el análisis del desempeño y la eficiencia del proceso. A continuación, se detallan estos datos en la Tabla 1.

***Tabla 1. Datos Operacionales del Sistema***

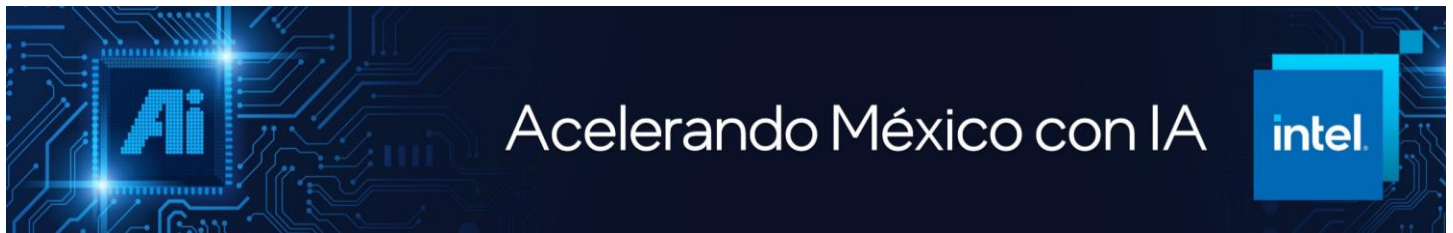
<b><i>Dato</i></b>	<b><i>Descripción</i></b>	<b><i>Unidad de Medida</i></b>
Presión de Entrada (Feed Pressure)	La presión a la que el agua se fuerza a pasar a través de la membrana.	Pascal (Pa)
Flujo de Permeado (Permeate Flow)	La cantidad de agua que pasa a través de la membrana.	Litros por hora (L/h)
Flujo de Rechazo o Concentrado (Concentrate Flow)	La cantidad de agua que no pasa por la membrana y se desecha.	Litros por hora (L/h)
Presión Diferencial (Differential Pressure)	La diferencia de presión entre la entrada y la salida del sistema de ósmosis.	Bar

Temperatura del Agua de Alimentación (Feed Water Temperature)	Afecta la eficiencia de la membrana.	Grados Celsius (°C)
---	--------------------------------------	---------------------

Para predecir de manera precisa las variables objetivo, el modelo también incorpora datos sobre las propiedades del agua de entrada y salida, incluyendo las Unidades de Turbidez Nefelométrica (NTU), las cual son fundamental para evaluar la calidad del tratamiento. Estos datos se presentan en la Tabla 1.2.

*Tabla 1.2 Propiedades del Agua Tratada*

<i>Dato</i>	<i>Descripción</i>	<i>Unidad de Medida</i>
Concentración de Sólidos Disueltos Totales (Total Dissolved Solids)	Para medir la salinidad.	Partes por millón (ppm)
Conductividad del Agua (Water Conductivity)	Relacionada con la salinidad y la cantidad de iones en el agua	Microsiemens por centímetro (µS/cm)
Concentración de Cloro (Chlorine Concentration)	Puede causar daño en la membrana si está presente.	Miligramos por litro (mg/L)
pH del Agua de Entrada (Feed Water pH)	Afecta el rendimiento y durabilidad de la membrana.	pH



Finalmente, se recopilan datos sobre las condiciones físicas de las membranas a utilizar, los cuales son cruciales para evaluar su estado y efectividad. Estos datos se describen en la Tabla 1.3.

*Tabla 1.3 Condiciones Físicas de las Membranas*

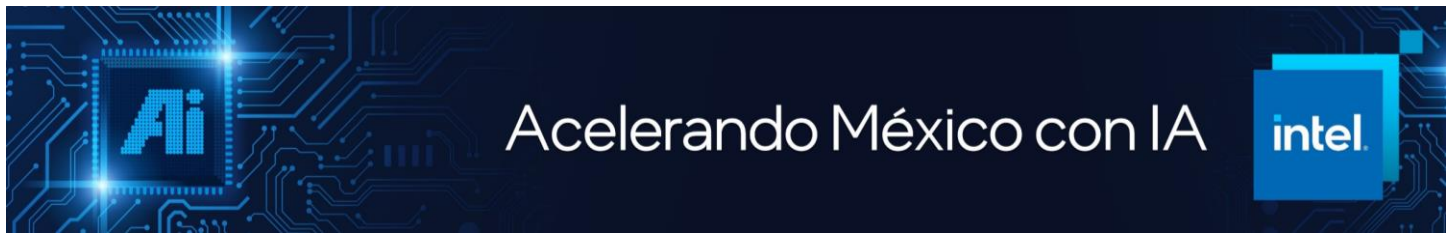
<i>Dato</i>	<i>Descripción</i>	<i>Unidad de Medida</i>
Horas de operación acumuladas (Total Operating Hours)	A mayor uso, mayor desgaste.	Horas (h)
Tasa de rechazo de sal (Salt Rejection Rate)	Eficiencia de la membrana para rechazar salinidad.	Porcentaje (%)
Ensuciamiento (Fouling Index)	Índice que mide la acumulación de material orgánico o inorgánico en la membrana.	Índice de ensuciamiento específico (SDI)
Tamaño de poros de la membrana (Membrane Pore Size)	Relacionado con el tipo de filtración que realiza la membrana.	Nanómetros (nm)

*Preprocesamiento de Datos*

Para el procesamiento de los datos, se siguieron varios pasos estructurados utilizando la biblioteca *Pandas* y herramientas de *Scikit-Learn*. En primer lugar, se importó el archivo CSV que contenía el conjunto de datos sobre Ósmosis Inversa.

```
# Importar Datos Osmosis Inversa
data = pd.read_csv("data/reverse_osmosis_data.csv")
```





A continuación, se definieron las variables predictoras y las variables objetivo:

1. Variables Predictoras (X): Se seleccionaron trece variables numéricas, incluyendo la presión de entrada, el flujo de permeado y el flujo de concentrado, que son fundamentales para modelar el comportamiento del sistema de Ósmosis Inversa.

```
# Variables Predictoras
X = data[["Feed Pressure (Pa)", "Permeate Flow (L/h)", "Concentrate Flow (L/h)" ,
         "Differential Pressure (bar)", "Feed Water Temperature (°C)",
         "Total Dissolved Solids (ppm)", "Water Conductivity (µS/cm)",
         "Chlorine Concentration (mg/L)", "Total Operating Hours (h)",
         "Salt Rejection Rate (%)", "Fouling Index (SDI)", "Feed Water pH",
         "Membrane Pore Size (nm)"]]
```

2. Variables Objetivo (Y): Se establecieron dos variables objetivo: *Y\_membrane\_condition*, que representa el estado de la membrana semipermeable, y *Y\_final\_tds*, que indica la concentración final de sólidos disueltos totales (TDS).

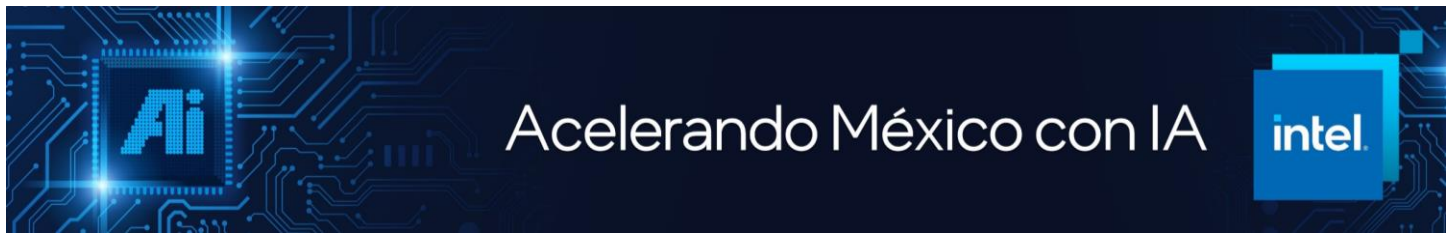
```
# Variables Objetivo
Y_membrane_condition = data["Membrane Condition"]
Y_final_tds = data["Final TDS (ppm)"]
```

### *Particiones*

Para procesar los datos, se crearon varias particiones utilizando la función *train\_test\_split* de la biblioteca *Scikit-Learn*. Estas particiones se distribuyeron de la siguiente manera: 60% de los datos se destinaron al entrenamiento del modelo, 20% a la validación y 20% a las pruebas. Este enfoque se aplicó a todas las variables que se desean predecir.

Al generar las particiones, se utilizó muestreo aleatorio para las variables numéricas y muestreo estratificado para la variable categórica que corresponde al estado de la membrana semipermeable. La elección de tratar esta variable como categórica permite que el modelo sea flexible y capaz de predecir diversos tipos de variables, enriqueciendo así su capacidad predictiva.





### ***Escalado de Datos***

Así mismo, para mejorar la precisión del modelo, los datos fueron escalados para que las variables mantuvieran un rango uniforme entre sus valores mínimos y máximos, utilizando la función *MinMaxScaler*, que ajusta los datos a un rango uniforme entre 0 y 1. Los datos de entrenamiento fueron escalados usando *scaler.fit\_transform()*, mientras que los conjuntos de validación y prueba fueron transformados con *scaler.transform()*, con el objetivo de evitar la filtración de datos (Data Leakage) durante el proceso de entrenamiento del modelo.

```
# Escalar Datos (Entrenamiento, Validacion y Prueba)
scaler = MinMaxScaler()
X_train_s = scaler.fit_transform(X_train)
X_val_s = scaler.transform(X_val)
X_test_s = scaler.transform(X_test)
```

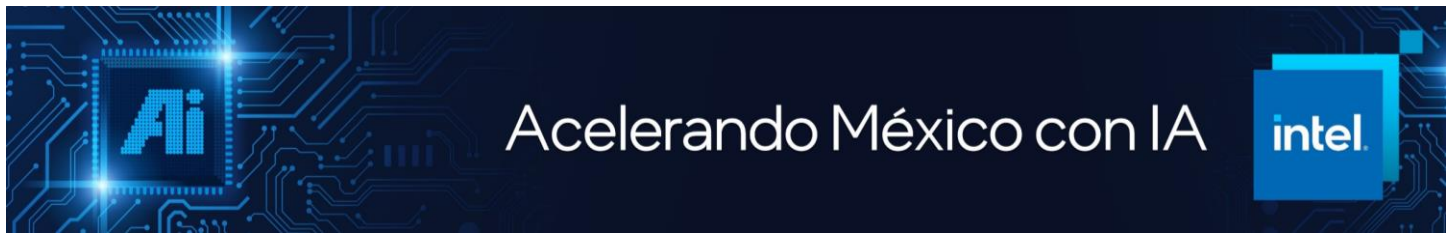
### ***Balanceo de la Variable Categórica***

Es importante resaltar que al predecir el estado de la membrana semipermeable, se empleó un modelo de clasificación donde algunas clases pueden estar sobrerrepresentadas en comparación con otras. Si no se equilibra, el modelo podría volverse sesgado, lo que afectaría negativamente su rendimiento en la predicción de las clases menos frecuentes.

A fin de abordar este problema, se aplicó el método *SMOTE* (Synthetic Minority Over-sampling Technique) a la variable *Y\_membrane\_condition*, para generar muestras sintéticas y evitar problemas de desigualdad en las clases del conjunto de datos, garantizando que el modelo no favorezca a las clases mayoritarias.

```
# Balanceo Clases Membrane Condition
smote = SMOTE(random_state = 42)
X_train_res, Y_train_membrane_res = smote.fit_resample(X_train_s, Y_train_membrane)

modelMembraneCondition(X_train_res, Y_train_membrane_res, X_val_s,
                        X_test_s, Y_val_membrane, Y_test_membrane)
ModelFinalTds(X_train_s, Y_train_tds, X_val_s, X_test_s, Y_val_tds, Y_test_tds)
```



### *Modelos*

Para predecir la variable categórica  $Y_{membrane\_condition}$ , se empleó un modelo de Clasificación basado en un Bosque Aleatorio, el cual demostró ser el más preciso después de realizar varias pruebas.

Los parámetros de entrada para este modelo son:

1. Número de Árboles en el Bosque Aleatorio: 50, 100 o 200.
2. Profundidad Máxima de Cada Árbol: 10, 20 o 30.
3. Mínimo de Muestras Requeridas para Dividir un Nodo: 2, 5 o 10.

Se estableció un parámetro adicional para asegurar la reproducibilidad de los resultados.

```
# Parametros Modelo Clasificación
parameters = {
    'n_estimators': [50, 100, 200],
    'max_depth': [None, 10, 20, 30],
    'min_samples_split': [2, 5, 10]
}
```

Para encontrar la mejor combinación de hiperparámetros, se utilizó la técnica de validación cruzada con cinco pliegues. Esto implica que el conjunto de datos se divide en cinco partes, y el modelo se entrena y valida cinco veces, utilizando una parte diferente como conjunto de validación y el resto como conjunto de entrenamiento en cada iteración. La evaluación del rendimiento del modelo se realizó utilizando la puntuación F1-Macro.

```
parameters_search = GridSearchCV(RandomForestClassifier(random_state = 42),
                                  parameters, cv = 5, scoring = "f1_macro")
parameters_search.fit(X_train_res, Y_train_membrane_res)
```

Para predecir la variable numérica  $Y_{final\_tds}$ , se utilizó un modelo de Regresión basado en un Bosque Aleatorio, configurado con 100 árboles y el mismo parámetro de reproducibilidad que el modelo anterior. El rendimiento de este modelo se evaluó mediante el cálculo del Error Cuadrático Medio (MSE) y el Coeficiente de Determinación ( $R^2$ ).

2. Escenario de Pruebas Según TRL

Tabla 2. Prueba de Concepto: Proceso de Validación a Nivel de Laboratorio TRL 2-3

<i>Nombre de la Prueba</i>	<i>Tipo de Prueba</i>	<i>Resultados/Conclusiones</i>
Precisión del Modelo de Bosque Aleatorio de Clasificación	Matriz de Confusión	La matriz de confusión para la validación mostró 54 verdaderos positivos, 2 falsos negativos, y 0 falsos positivos, resultando en una precisión del 98% (56% para la categoría Desgaste Leve, 100% para Desgaste Moderado y Severo, 93% para Excelente Estado y 96% para Mal Estado).
Evaluación del Rendimiento del Modelo de Bosque Aleatorio de Clasificación	Informe de Clasificación	El informe de clasificación indicó una precisión promedio de 98% y un F1-Score de 0.98 en general, validando la robustez del modelo, con un alto desempeño en todas las categorías.
Evaluación de Error del Modelo de Bosque Aleatorio de Regresión	Error Cuadrático Medio	En la validación, el Error Cuadrático Medio (MSE) fue de 4.50, indicando que el modelo tuvo un rendimiento preciso en la predicción de TDS. En la prueba, el MSE fue de 4.03, mostrando un desempeño sólido.
Evaluación de la Proporción de Varianza	Coefficiente de Determinación	En la validación, el Coeficiente de Determinación ( $R^2$ ) fue de 0.9957, indicando

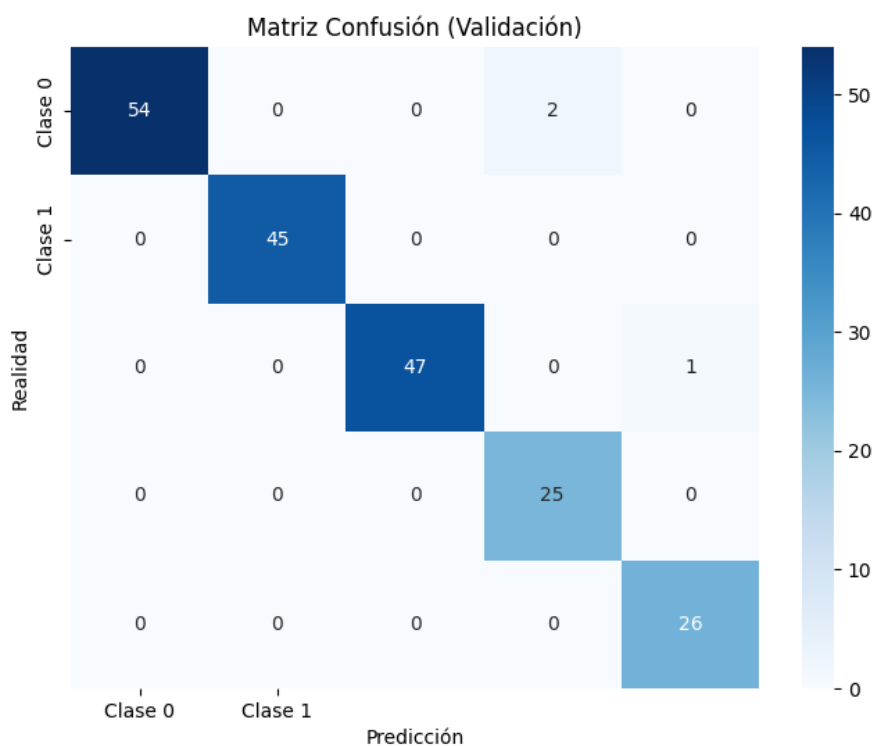
del Modelo de Bosque  
Aleatorio de Regresión

que el modelo explicó el 99.57% de la  
varianza en los datos de validación. En la  
prueba, el  $R^2$  alcanzó 0.9962, confirmando  
que el modelo mantuvo un rendimiento  
efectivo en la predicción.

### *Modelo Clasificador de Bosque Aleatorio para "Membrane Condition"*

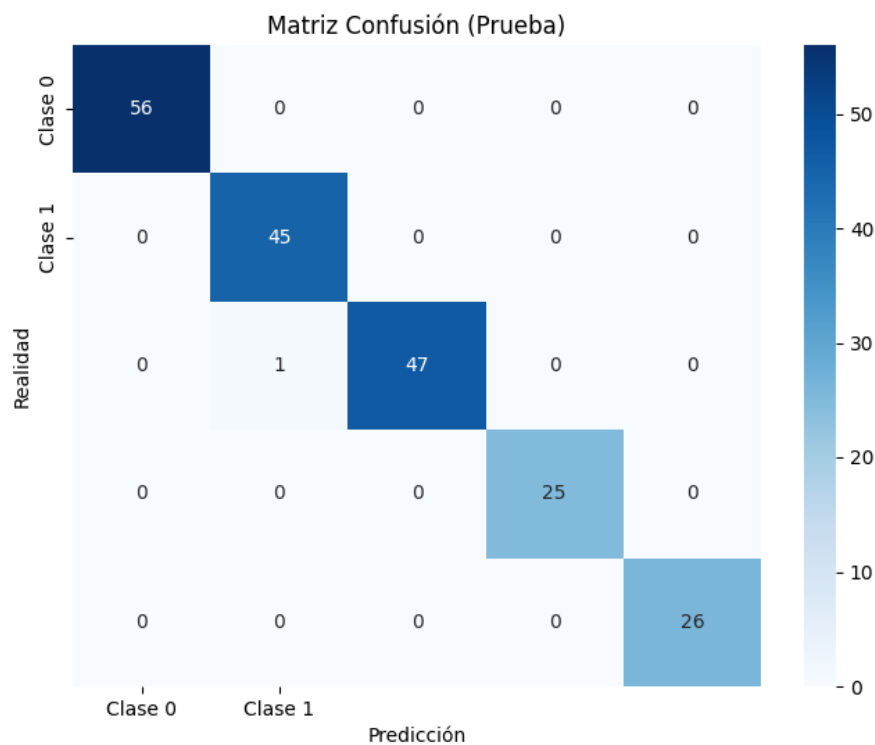
En cuanto a la validación, el modelo alcanzó una precisión del 98%, con un F1-Score muy alto en todas las categorías. La matriz de confusión muestra muy pocos errores de clasificación, lo que indica que el modelo es capaz de identificar correctamente las condiciones de las membranas. Se detallan estos datos en la Figura 1.

**Figura 1. Matriz de Confusión de Validación**



En cuanto a la prueba, el modelo mantuvo su rendimiento con una precisión del 99%. Los resultados son coherentes entre las categorías y prácticamente no hay errores. Estos datos se presentan en la Figura 1.2.

***Figura 1.2 Matriz de Confusión de Prueba***



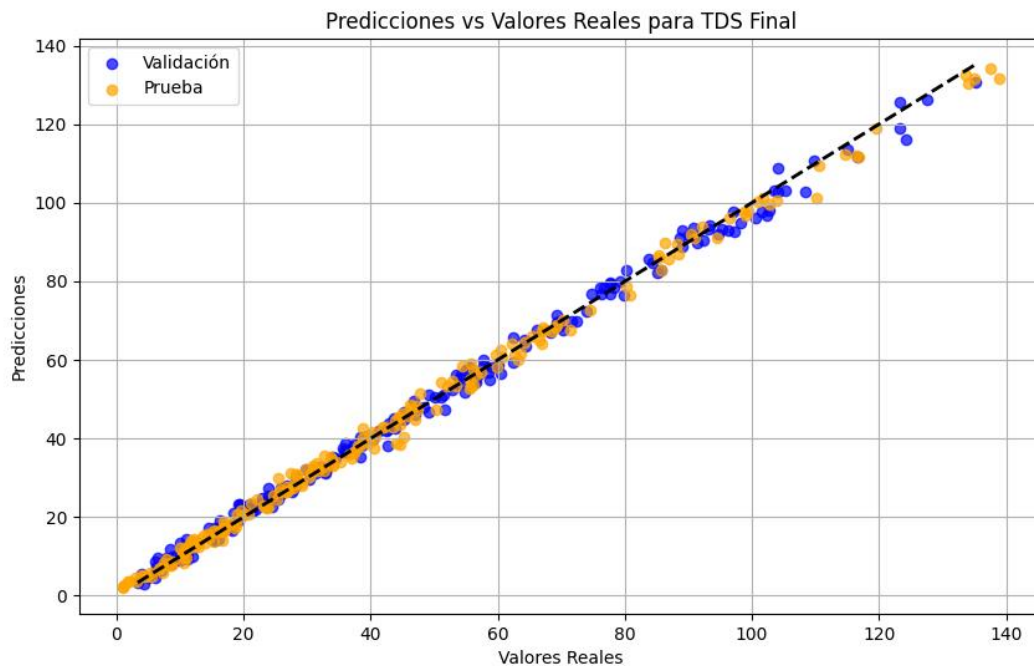
### ***Modelo Bosque Aleatorio de Regresión para "Final TDS"***

Para la validación del modelo de regresión, el Error Cuadrático Medio (MSE) es muy bajo (4.50), lo que indica que el modelo ajusta bien los datos de validación. El Coeficiente de Determinación ( $R^2$ ) es 0.996, mostrando una predicción casi perfecta.



Para la prueba, el MSE de 4.03 es aún mejor, lo que confirma que el modelo generaliza bien en nuevos datos. El valor de  $R^2$  es también extremadamente alto (0.996), lo que significa que la varianza explicada por el modelo es prácticamente completa. Se detallan estos datos en la Figura 2.

***Figura 2. Predicciones y Valores Reales TDS Final***



En resumen, los modelos funcionan muy bien tanto en Clasificación como en Regresión, lo que indica que los datos están bien modelados y que las predicciones son altamente precisas y consistentes.