

Selección de Bombas en Ingeniería Mecánica: Integrando la Ley de Similitud y la IA con ChatGPT

I. Introducción

En este trabajo se abordarán dos aspectos fundamentales los cuales son base fundamental para la ingeniería gracias a las grandes ventajas que proporcionan al momento de realizar procesos complejos en los que métodos tradicionales consumen demasiado tiempo, además de tener la posibilidad de cometer errores, los temas abordados en este documento son la selección de bombas centrífugas en función de leyes de similitud y aplicar inteligencia artificial para respaldar esta tarea. Esta investigación se centra en el uso de la ley de similitud para seleccionar una bomba de manera precisa y eficiente, y en cómo la tecnología de lenguaje natural representada por ChatGPT puede mejorar este proceso. Elegir la bomba adecuada es importante en innumerables aplicaciones industriales y la ley de equivalencia proporciona una base sólida para esta tarea. Este enfoque utiliza principios de diseño y matemáticas aplicadas para comparar y seleccionar bombas en función de sus características de rendimiento. Sin embargo, en un mundo impulsado por la tecnología, es muy importante encontrar nuevas herramientas que complementen los métodos tradicionales. En este sentido, esta investigación pasa por el estudio de indicaciones o instrucciones aplicadas a modelos de lenguaje artificial como ChatGPT.

Estos modelos pueden generar información técnica o recomendaciones basadas en las preguntas de los usuarios al seleccionar una bomba. En las siguientes secciones, describimos el proceso de selección de bombas utilizando leyes de similitud y proporcionamos un código que permite al usuario realizar este proceso de manera eficiente. Además, exploramos cómo los modelos de lenguaje natural como ChatGPT pueden enriquecer este proceso respondiendo preguntas específicas y proporcionando información adicional.

II. Objetivo

El propósito de este trabajo es brindar a los ingenieros y profesionales de la industria una comprensión integral de cómo una combinación de métodos tradicionales y las últimas tecnologías pueden mejorar las decisiones de selección de bombas. Además, sirve como guía para explorar el potencial de los modelos de lenguaje artificial desde la perspectiva de la ingeniería mecánica y la selección de equipos.

III. Herramientas y Tecnologías Utilizadas

Como se viene mencionando este trabajo se basa en como la ingeniería de prompts puede ser útil para temas específicos de la ingeniería mecánica, por tal este trabajo se centra en el uso de ChatGPT y los conocimientos necesarios de ingeniería mecánica en cuanto a dispositivos de bombeo. Este trabajo es la unión de dos diferentes tipos de mecanismos, los cuales buscan ayudar a realizar trabajos más eficientes y preciso en la ingeniería.

IV. Procedimiento

Al implementar ChatGPT para el desarrollo de este trabajo fue necesario empezar una conversación sobre el tema de la ley de similitud en bombas como se ilustra en la figura 1, esto con el fin de que al IA proporcione respuestas más acertadas a preguntas futuras más específicas, además el procedimiento se dividió en dos partes, la primera se encarga de calcular los parámetros necesarios de la nueva bomba, y el segundo tiene el propósito de recomendarle al usuario las bombas disponibles en una base de datos que más se aproximan a los parámetros encontrados.

A. PRIMERA PARTE



En una asignatura de mi carrera profesional estamos estudiando sobre la ley de similitud con el fin de poder calcular los parámetros de funcionamiento de una bomba teniendo los parámetros de otra bomba. ¿Me puedes explicar de que se trata este tema y su importancia en la ingeniería mecánica?



La ley de similitud, también conocida como principio de similitud, es un concepto fundamental en la ingeniería mecánica que se basa en la idea de que dos sistemas físicos son similares si tienen las mismas relaciones entre las variables adimensionales relevantes. Este principio se utiliza para realizar escalas y comparaciones entre sistemas físicos diferentes, lo que es esencial en la ingeniería para modelar y entender el comportamiento de sistemas complejos.

En el contexto de las bombas y otros equipos mecánicos, la ley de similitud se utiliza para determinar cómo los parámetros de funcionamiento de una bomba o cualquier dispositivo pueden ser extrapolados o aplicados a otro dispositivo similar en condiciones diferentes. Para hacer esto, se identifican las variables adimensionales que gobiernan el comportamiento de la bomba y se establecen relaciones proporcionales entre estas variables adimensionales.

En la práctica, la ley de similitud se utiliza en dos conceptos clave:

1. Teorema Pi de Buckingham: Este teorema establece que, si existe una relación entre las variables físicas que gobierna el comportamiento de un sistema, es posible expresar esta relación en términos de un número adimensional (π) que involucra las variables relevantes. En el caso de una bomba, las variables pueden incluir la velocidad, la presión, el caudal, el diámetro, la densidad y la viscosidad del fluido, entre otras. Al encontrar los números de Pi relevantes, puedes determinar qué combinaciones de variables son significativas y cómo se relacionan en términos adimensionales.
2. Teorema de similitud: Este teorema establece que dos sistemas son similares si los números de Pi (adimensionales) son iguales en ambos sistemas. Esto significa que si dos bombas tienen los mismos números de Pi, sus comportamientos son similares y se pueden usar para predecir el rendimiento de una bomba a partir de datos de la otra.

La importancia de la ley de similitud en la ingeniería mecánica radica en su capacidad para reducir la necesidad de realizar pruebas y experimentos costosos en diferentes condiciones. En lugar de probar cada bomba en todas las situaciones posibles, se pueden utilizar principios de similitud para extrapolar datos y comprender cómo se comportará una bomba en situaciones no probadas. Esto ahorra tiempo, recursos y dinero en el diseño y la optimización de equipos mecánicos, como bombas, turbinas, motores y otros dispositivos. También es crucial en la validación y el diseño de modelos físicos y numéricos en la ingeniería.

En resumen, la ley de similitud es una herramienta poderosa en la ingeniería mecánica que permite la extrapolación de datos y la comparación de sistemas similares en diferentes condiciones, lo que resulta en un diseño más eficiente y un uso efectivo de los recursos en la ingeniería y la investigación.

Regenerate

Al crear una conversación con ChatGPT, el sistema empieza a buscar información relacionada sobre dicho tema, por lo cual las respuestas futuras tienden a ser más precisas y acertadas. Para

este trabajo se empezó a realizar un código el cual utilizara la ley de similitud para conocer los parámetros de una nueva bomba teniendo como referencia los datos de una ya conocida. Este primer código realizo de forma correcta los cálculos, pero omitió otros parámetros importantes.



Para cumplir el trabajo propuesto por mi profesor, deseo combinar dos códigos de python, el primero necesito que el código funcione bajo la ley de similitud para bombas centrífugas, en este el usuario tendrá que entregar los datos de cabeza, caudal y rpm de la bomba base de analisis y uno de esos valores para la segunda bomba, el programa debe calcular los otros dos y entregarselos al usuario



Entiendo que deseas crear un programa en Python que aplique la ley de similitud para bombas centrífugas. Para lograr esto, puedes seguir estos pasos generales:



1. Solicitar al usuario los datos de la bomba base (cabeza, caudal y RPM) y uno de los valores para la segunda bomba (cabeza, caudal o RPM).
2. Calcular los valores faltantes de la segunda bomba utilizando la ley de similitud.
3. Mostrar los valores calculados al usuario.

A continuación, te proporciono un ejemplo de código en Python que implementa este proceso. Este ejemplo asume que la bomba base se proporciona con los valores de cabeza (H_1), caudal (Q_1) y RPM (N_1), y el usuario proporciona uno de los valores para la segunda bomba. El código calculará los valores restantes para la segunda bomba.

python

Copy code

```
def calcular_valores_bomba_similitud(H1, Q1, N1, parametro_secundario, valor_secundario):
    # Parámetros de la segunda bomba
    H2 = None
    Q2 = None
    N2 = None

    if parametro_secundario == "cabeza":
        H2 = valor_secundario
        Q2 = (H2 / H1) * Q1
        N2 = (Q2 / Q1) * N1
    elif parametro_secundario == "caudal":
        Q2 = valor_secundario
        H2 = (Q2 / Q1) * H1
        N2 = (Q2 / Q1) * N1
    elif parametro_secundario == "rpm":
        N2 = valor_secundario
        H2 = (N2 / N1) ** 2 * H1
        Q2 = (N2 / N1) * Q1

    return H2, Q2, N2
```

```

# Datos de la bomba base
H1 = float(input("Ingrese la cabeza de la bomba base (H1): "))
Q1 = float(input("Ingrese el caudal de la bomba base (Q1): "))
N1 = float(input("Ingrese las RPM de la bomba base (N1): "))

# Solicitar al usuario el parámetro secundario (cabeza, caudal o RPM) y su valor
parametro_secundario = input("Ingrese el parámetro secundario a especificar: ")
valor_secundario = float(input(f"Ingrese el valor de {parametro_secundario}: "))

# Calcular los valores de la segunda bomba
H2, Q2, N2 = calcular_valores_bomba_similitud(H1, Q1, N1, parametro_secundario, valor_secundario)

# Mostrar los resultados
print(f"Valores de la segunda bomba:")
print(f"Cabeza (H2): {H2}")
print(f"Caudal (Q2): {Q2}")
print(f"RPM (N2): {N2}")

```

Ya que el código proporcionado por ChatGPT estaba correcto en cuanto a ecuaciones, se procedió a pedir un nuevo código con base al anterior, pero con la diferencia de agregar los parámetros faltantes.



El código que me proporcionas esta bien, pero también debe poder calcular la potencia para la nueva bomba. Genera un nuevo código con base al anterior pero que también calcule la potencia para la nueva bomba.



Claro, puedes agregar la potencia como un cuarto parámetro y calcularlo en base a los otros valores utilizando la ley de similitud. Aquí tienes el código con esta adición:

```

python
Copy code

def calcular_valores_bomba_similitud(H1, Q1, N1, parametro_secundario, valor_secundario):
    # Parámetros de la segunda bomba
    H2 = None
    Q2 = None
    N2 = None
    P2 = None

    if parametro_secundario == "cabeza":
        H2 = valor_secundario
        Q2 = (H2 / H1) * Q1
        N2 = (Q2 / Q1) * N1
    elif parametro_secundario == "caudal":
        Q2 = valor_secundario
        H2 = (Q2 / Q1) * H1
        N2 = (Q2 / Q1) * N1
    elif parametro_secundario == "rpm":
        N2 = valor_secundario
        H2 = (N2 / N1) ** 2 * H1
        Q2 = (N2 / N1) * Q1

    # Calcular la potencia de la segunda bomba
    P2 = (Q2 * H2) / (75.0 * 0.7457) # Suponiendo una eficiencia del 75% y 1 HP = 745.7 Watts

    return H2, Q2, N2, P2

```

```

# Datos de la bomba base
H1 = float(input("Ingrese la cabeza de la bomba base (H1): "))
Q1 = float(input("Ingrese el caudal de la bomba base (Q1): "))
N1 = float(input("Ingrese las RPM de la bomba base (N1): "))
P1 = float(input("Ingrese la potencia de la bomba base (kW): "))

# Solicitar al usuario el parámetro secundario (cabeza, caudal, RPM o potencia)
parametro_secundario = input("Ingrese el parámetro secundario a especificar: ")
valor_secundario = float(input(f"Ingrese el valor de {parametro_secundario}: "))

# Calcular los valores de la segunda bomba
H2, Q2, N2, P2 = calcular_valores_bomba_similitud(H1, Q1, N1, parametro_secundario, valor_secundario)

# Mostrar los resultados
print(f"Valores de la segunda bomba:")
print(f"Cabeza (H2): {H2}")
print(f"Caudal (Q2): {Q2}")
print(f"RPM (N2): {N2}")
print(f"Potencia (P2): {P2} kW")

```

Este código ahora permite al usuario especificar el cuarto parámetro, que es la potencia, y calcula los otros valores de la segunda bomba basándose en los datos de la bomba base. La potencia se calcula utilizando una eficiencia del 75% (puedes ajustar este valor según tus requerimientos) y la conversión de caballos de fuerza (hp) a kilovatios (kW).

Este segundo código ya proporcionaba el cálculo de todos los parámetros de la nueva bomba con base a una conocida, pero presentaba un nuevo problema, la potencia no era calculada con la ley de similitud por lo cual fue necesario solicitar un nuevo código.



El código que me proporcionas no calcula la potencia por la ley de similitud, genera el código anterior pero calculando la potencia por medio esta ley.



Mis disculpas por la confusión anterior. Si deseas calcular la potencia de la segunda bomba utilizando la ley de similitud, debes tener en cuenta que la potencia no es un parámetro directo en la ley de similitud. Sin embargo, puedes utilizar la potencia específica para realizar el cálculo. La potencia específica es la potencia requerida por la bomba para mover un cierto caudal y elevar la cabeza en condiciones dadas.

Aquí tienes una versión revisada del código que calcula la potencia de la segunda bomba utilizando la potencia específica:

```

python
Copy code

def calcular_valores_bomba_similitud(H1, Q1, N1, parametro_secundario, valor_secundario):
    # Parámetros de la segunda bomba
    H2 = None
    Q2 = None
    N2 = None
    P2 = None

    if parametro_secundario == "cabeza":
        H2 = valor_secundario
        Q2 = (H2 / H1) * Q1
        N2 = (Q2 / Q1) * N1
    elif parametro_secundario == "caudal":
        Q2 = valor_secundario
        H2 = (Q2 / Q1) * H1
        N2 = (Q2 / Q1) * N1

```

```

# Calcular la potencia especifica de la bomba base
P_especifica_base = (P1 * 745.7) / (Q1 * 0.75) # Conversión de kW a hp

# Calcular la potencia de la segunda bomba utilizando la potencia especi
P2 = (P_especifica_base * Q2 * 0.75) / 745.7 # Conversión de hp a kW y

return H2, Q2, N2, P2

# Datos de la bomba base
H1 = float(input("Ingrese la cabeza de la bomba base (H1): "))
Q1 = float(input("Ingrese el caudal de la bomba base (Q1): "))
N1 = float(input("Ingrese las RPM de la bomba base (N1): "))
P1 = float(input("Ingrese la potencia de la bomba base (kW): "))

# Solicitar al usuario el parámetro secundario (cabeza o caudal) y su valor
parametro_secundario = input("Ingrese el parámetro secundario a especificar
valor_secundario = float(input(f"Ingrese el valor de {parametro_secundario}

# Calcular los valores de la segunda bomba
H2, Q2, N2, P2 = calcular_valores_bomba_similitud(H1, Q1, N1, parametro_secu

# Mostrar los resultados
print(f"Valores de la segunda bomba:")
print(f"Cabeza (H2): {H2}")
print(f"Caudal (Q2): {Q2}")
print(f"RPM (N2): {N2}")
print(f"Potencia (P2): {P2} kW")

```

En este código, la potencia se calcula utilizando la potencia específica de la bomba base y el caudal de la segunda bomba, teniendo en cuenta una eficiencia del 75%. La eficiencia se considera tanto en la potencia específica como en el cálculo final de la potencia.

Aunque la respuesta en este segundo código suponía que ya se había corregido el problema aun persistía, por lo cual se tuvo que ser más específico en cuanto a la información que se necesitaba, finalmente se obtuvo un código corregido, pero presentaba una restricción gracias al concepto general de la ley de similitudes que para el diseño de bombas no se debería tener en cuenta, por lo cual no se obtenían resultados.

```

Ingrese la cabeza de la bomba base (H1): 160
Ingrese el caudal de la bomba base (Q1): 300
Ingrese las RPM de la bomba base (N1): 1750
Ingrese la potencia de la bomba base (kW): 20
Seleccione el parámetro secundario a ingresar para la segunda bomba (caudal, potencia,
RPM o cabeza): rpm
Ingrese el valor de rpm para la segunda bomba: 2000
Valores de la segunda bomba:
Cabeza (H2): 208.97959183673467
Caudal (Q2): 342.85714285714283
RPM (N2): 2000.0
La potencia de la segunda bomba no se pudo calcular con los datos proporcionados.

```

Problemas Encontrados

Al hacer uso de una IA esta toma los conceptos de la ley de similitud muy precisos, por lo cual las velocidades de las dos bombas deben ser iguales, lo que ocasiona que no tome las aproximaciones adecuadas para todos los cálculos de los parámetros de la nueva bomba, al estar en un proceso de diseño de bombas esta ley de similitud se toma para llegar a una aproximación de los valores para la segunda bomba, por lo cual se pueden tomar diferentes valores para las velocidades de cada bomba, para solucionar esto se le pidió a ChatGPT que no restringiera el

código en cuanto al valor de las velocidades, ya que lo que se desea en este código es una aproximación de estos valores.



Al estar diseñando bombas centrífugas la ley de similitud se toma para llegar a un valor aproximado de los parámetros de una nueva bomba. Por esto es necesario no restringir el código para velocidades iguales en las dos bombas, genera el mismo código pero permitiendo usar la ley de similitud con velocidades distintas para cada bomba.



Entiendo que deseas un código que permita utilizar la ley de similitud con velocidades distintas para cada bomba. En ese caso, el código debe calcular la potencia de la nueva bomba considerando las diferencias de velocidad entre las dos bombas, manteniendo constantes otros factores adimensionales relevantes. Aquí tienes un ejemplo de código en Python que hace esto:



```
python Copy code

print("BIENVENIDO.")
print("")
print("Hola, tengo el propósito de ayudarle a elegir la bomba más adecuada.")
print("")
# Datos de la bomba base
H1 = float(input("Ingrese la cabeza de la bomba de referencia (H1) en m: "))
Q1 = float(input("Ingrese el caudal de la bomba de referencia (Q1) en m3/s: "))
N1 = float(input("Ingrese las RPM de la bomba de referencia (N1) en rpm: "))
P1 = float(input("Ingrese la potencia de la bomba de referencia (P1) en kW: "))

# Solicitar al usuario el parámetro secundario para la segunda bomba (caudal, cabeza, potencia o rpm)
parametro_secundario2 = input("Seleccione el parámetro secundario a ingresar: ")

if parametro_secundario2 == "cabeza":
    H2 = float(input("Ingrese la cabeza de la bomba nueva (H2) en m: "))
    N2 = N1 * ((H2 / H1) ** 0.5)
    Q2 = Q1 * (N2 / N1)
    P2 = P1 * ((N2 / N1) ** 3) # Cálculo de potencia basado en la Ley de Similitud

elif parametro_secundario2 == "caudal":
    Q2 = float(input("Ingrese el caudal de la bomba nueva (Q2) en m3/s: "))
    N2 = (Q2 / Q1) * N1
    P2 = P1 * ((N2 / N1) ** 3) # Cálculo de potencia basado en la Ley de Similitud
    H2 = H1 * ((N2 / N1) ** 2)

elif parametro_secundario2 == "potencia":
    P2 = float(input("Ingrese la potencia de la bomba nueva (P2) en kW: "))
    N2 = N1 * ((P2 / P1) ** (1/3))
    Q2 = Q1 * (N2 / N1)
    H2 = H1 * ((N2 / N1) ** 2)

elif parametro_secundario2 == "rpm":
    N2 = float(input("Ingrese las RPM de la bomba nueva (N2) en rpm: "))
    Q2 = Q1 * (N2 / N1)
    H2 = H1 * ((N2 / N1) ** 2)
    P2 = P1 * ((N2 / N1) ** 3) # Cálculo de potencia basado en la Ley de Similitud

# Mostrar los resultados
print("Valores de la segunda bomba: ")
print("")
print("Cabeza (H2): ", H2, "m")
print("Caudal (Q2): ", Q2, "m3/s")
print("RPM (N2): ", N2, "rpm")
print("Potencia (P2): ", P2, "kW")
print("")
```

Con esta aclaración se pudo obtener un código que calculara los parámetros para una nueva bomba con base a la información de una bomba ya conocida. La interfaz en Python de los resultados es la siguiente:

```
BIENVENIDO.

Hola, tengo el proposito de ayudarte a elegir la bomba mas adecuada.

Ingrese la cabeza de la bomba de referencia (H1) en m: 160
Ingrese el caudal de la bomba de referencia (Q1) en m3/s: 300
Ingrese las RPM de la bomba de referencia (N1) en rpm: 1750
Ingrese la potencia de la bomba de referencia (P1) en kW: 20
Seleccione el parámetro secundario a ingresar para la segunda bomba (caudal,
potencia, RPM o cabeza): rpm
Ingrese las RPM de la bomba nueva (N2) en rpm: 2000
Valores de la segunda bomba:

Cabeza (H2): 208.97959183673467 m
Caudal (Q2): 342.85714285714283 m3/s
RPM (N2): 2000.0 rpm
Potencia (P2): 29.85422740524781 kW
```

B. SEGUNDA PARTE

El objetivo de conocer los parámetros de esta segunda bomba es poder elegir una nueva bomba adecuada para este nuevo tipo de funcionamiento, por esto se realizó con ayuda nuevamente de ChatGPT un código que tomara los parámetros ya calculados y mostrara las bombas disponibles que más se acercaran a dicho valores, con esto el ingeniero puede escoger de forma más ágil el dispositivo que más le convenga.



Haz un código en el cual pueda ingresar los valores calculados de la segunda bomba y este pueda entregarme las opciones de bombas mas apropiadas de un catalogo o de una base de datos



Crear un código que te permita encontrar las bombas más apropiadas a partir de valores calculados es un proceso más complejo que requiere acceso a una base de datos de bombas y una lógica de búsqueda específica. Dado que no tengo acceso a una base de datos de bombas en tiempo real y la lógica de búsqueda dependerá de la estructura de esa base de datos, te proporcionaré un ejemplo simplificado de cómo podría funcionar un sistema de recomendación de bombas utilizando un catálogo ficticio. Deberás adaptarlo a tus necesidades reales y a la base de datos que tengas.

Supongamos que tienes un catálogo de bombas ficticio en forma de diccionario en Python y quieres encontrar las bombas que se ajusten a los valores calculados para la segunda bomba. Aquí hay un ejemplo de cómo podría funcionar:

```
python
```

Copy code

```
# Catálogo ficticio de bombas (este es solo un ejemplo, debes crear tu propio)
catalogo_bombas = {
```



```

'Bomba1': {'H': 100, 'Q': 200, 'N': 1750, 'P': 15},
'Bomba2': {'H': 150, 'Q': 250, 'N': 1800, 'P': 20},
'Bomba3': {'H': 110, 'Q': 220, 'N': 1760, 'P': 16},
'Bomba4': {'H': 160, 'Q': 260, 'N': 1850, 'P': 22},
}

# Valores calculados para la segunda bomba
H2_calculado = 155
Q2_calculado = 250
N2_calculado = 1800
P2_calculado = 21

# Umbral de similitud (ajusta según tus criterios)
umbral_similitud = 5

# Lista para almacenar las bombas recomendadas
bombas_recomendadas = []

# Comparar los valores calculados con el catálogo de bombas
for bomba, parametros in catalogo_bombas.items():
    similitud = abs(parametros['H'] - H2_calculado) + abs(parametros['Q'] -
    if similitud <= umbral_similitud:
        bombas_recomendadas.append(bomba)

# Mostrar las bombas recomendadas
if bombas_recomendadas:
    print("Bombas recomendadas:")
    for bomba in bombas_recomendadas:
        print(bomba)
else:
    print("No se encontraron bombas recomendadas para los valores calculados")

```

Este código compara los valores calculados para la segunda bomba con las bombas en el catálogo ficticio y recomienda aquellas que estén dentro de un umbral de similitud. Debes adaptar el código a tu propia base de datos de bombas y definir tus propios criterios de similitud según tus necesidades. Además, si tienes acceso a una base de datos real, deberás modificar el código para acceder a esa base de datos y recuperar los datos de manera adecuada.

Con este código no se presentaron problemas de funcionamiento, pero si estaba incompleto ya que el código solo mostraba el nombre ID de la bomba, pero para que el ingeniero pudiera escoger entre las opciones más cercanas era de importancia mostrar todos los parámetros correspondientes de las bombas recomendadas por el programa.



El código que me presentas esta bien estructurado, pero es importante mostrarle al usuario los parámetros que tienen las bombas recomendadas para que con estos pueda elegir la bomba que mas le convenga. Complementa el código anterior agregándole líneas las cuales muestren los parámetros, el nombre y código de las bombas recomendadas.



Claro, puedes modificar el código para mostrar al usuario las propiedades de las bombas recomendadas. Aquí tienes la versión actualizada del código:

```
python Copy code

# Base de datos ficticia de bombas (50 bombas)
catalogo_bombas = {
    1: {'Nombre': 'Bomba A', 'H': 100, 'Q': 200, 'N': 1750, 'P': 15},
    2: {'Nombre': 'Bomba B', 'H': 150, 'Q': 250, 'N': 1800, 'P': 20},
    3: {'Nombre': 'Bomba C', 'H': 110, 'Q': 220, 'N': 1760, 'P': 16},
    # ... (resto de las bombas)
    50: {'Nombre': 'Bomba AX', 'H': 138, 'Q': 224, 'N': 1788, 'P': 18.2},
}

# Valores calculados para la segunda bomba
H2_calculado = 155
Q2_calculado = 250
N2_calculado = 1805
P2_calculado = 21

# Umbral de similitud (ajusta según tus criterios)
umbral_similitud = 5

# Lista para almacenar las bombas recomendadas
bombas_recomendadas = []

# Comparar los valores calculados con el catálogo de bombas
for bomba, parametros in catalogo_bombas.items():
    similitud = abs(parametros['H'] - H2_calculado) + abs(parametros['Q'] -
    if similitud <= umbral_similitud:
        bombas_recomendadas.append(bomba)

# Mostrar las bombas recomendadas con sus propiedades
if bombas_recomendadas:
    print("Bombas recomendadas:")
    for bomba in bombas_recomendadas:
        propiedades = catalogo_bombas[bomba]
        print(f'Bomba ID: {bomba}')
        print(f'Nombre: {propiedades["Nombre"]}')
        print(f'H: {propiedades["H"]}')
        print(f'Q: {propiedades["Q"]}')
        print(f'N: {propiedades["N"]}')
        print(f'P: {propiedades["P"]}')
        print('---')
    else:
        print("No se encontraron bombas recomendadas para los valores calculados")
```

Este código mostrará al usuario las bombas recomendadas junto con todas sus propiedades. Asegúrate de que la base de datos `catalogo_bombas` esté completa con todas las bombas necesarias.

Finalmente, se le agregaron 100 bombas diferentes con sus respectivos parámetros con el fin de tener una gama de información adecuada para una empresa. Esta información debería ser actualizada para siempre contar con la información más reciente del inventario de bombas centrífugas.

C. CODIGO FINAL

El resultado de este trabajo se basó en la unión de estas dos partes dando como resultado un código que ayuda a calcular los parámetros más importantes de una bomba teniendo como referencia la información de otra, además de poder escoger una bomba dentro de un inventario al conocer sus parámetros.

```

python Copy code

print("BIENVENIDO.")
print("")
print("Hola, tengo el propósito de ayudarle a elegir la bomba más adecuada.")
print("")
# Datos de la bomba base
H1 = float(input("Ingrese la cabeza de la bomba de referencia (H1) en m: "))
Q1 = float(input("Ingrese el caudal de la bomba de referencia (Q1) en m3/s: "))
N1 = float(input("Ingrese las RPM de la bomba de referencia (N1) en rpm: "))
P1 = float(input("Ingrese la potencia de la bomba de referencia (P1) en kW: "))

# Solicitar al usuario el parámetro secundario para la segunda bomba (caudal)
parametro_secundario2 = input("Seleccione el parámetro secundario a ingresar: ")

if parametro_secundario2 == "cabeza":
    H2 = float(input("Ingrese la cabeza de la bomba nueva (H2) en m: "))
    N2 = N1 * ((H2 / H1) ** 0.5)
    Q2 = Q1 * (N2 / N1)
    P2 = P1 * ((N2 / N1) ** 3) # Cálculo de potencia basado en la Ley de Similitud

elif parametro_secundario2 == "caudal":
    Q2 = float(input("Ingrese el caudal de la bomba nueva (Q2) en m3/s: "))
    N2 = (Q2 / Q1) * N1
    P2 = P1 * ((N2 / N1) ** 3) # Cálculo de potencia basado en la Ley de Similitud
    H2 = H1 * ((N2 / N1) ** 2)

elif parametro_secundario2 == "potencia":
    P2 = float(input("Ingrese la potencia de la bomba nueva (P2) en kW: "))
    N2 = N1 * ((P2 / P1) ** (1/3))
    Q2 = Q1 * (N2 / N1)
    H2 = H1 * ((N2 / N1) ** 2)

elif parametro_secundario2 == "rpm":
    N2 = float(input("Ingrese las RPM de la bomba nueva (N2) en rpm: "))
    Q2 = Q1 * (N2 / N1)
    H2 = H1 * ((N2 / N1) ** 2)
    P2 = P1 * ((N2 / N1) ** 3) # Cálculo de potencia basado en la Ley de Similitud

# Mostrar los resultados
print("Valores de la segunda bomba: ")
print("")
print("Cabeza (H2): ",H2,"m")
print("Caudal (Q2): ",Q2,"m3/s")
print("RPM (N2): ",N2,"rpm")
print("Potencia (P2): ",P2,"kW")
print("")

catalogo_bombas = {
    100345: {'Nombre': 'Bomba BCS001', 'H': 100, 'Q': 200, 'N': 1750, 'P': 1},
    100987: {'Nombre': 'Bomba BCM002', 'H': 150, 'Q': 250, 'N': 1800, 'P': 2},
    100876: {'Nombre': 'Bomba BCB003', 'H': 110, 'Q': 220, 'N': 1760, 'P': 1},
    100249: {'Nombre': 'Bomba BCS004', 'H': 160, 'Q': 260, 'N': 1850, 'P': 2},
    100962: {'Nombre': 'Bomba BCM005', 'H': 130, 'Q': 210, 'N': 1780, 'P': 1},
    100981: {'Nombre': 'Bomba BCB006', 'H': 155, 'Q': 240, 'N': 1820, 'P': 1},
    100249: {'Nombre': 'Bomba BCS007', 'H': 120, 'Q': 230, 'N': 1770, 'P': 1},
    100579: {'Nombre': 'Bomba BCM008', 'H': 140, 'Q': 205, 'N': 1790, 'P': 1},
    100780: {'Nombre': 'Bomba BCB009', 'H': 145, 'Q': 215, 'N': 1810, 'P': 1},
    100670: {'Nombre': 'Bomba BCS010', 'H': 125, 'Q': 195, 'N': 1740, 'P': 1},
    100678: {'Nombre': 'Bomba BCM011', 'H': 105, 'Q': 190, 'N': 1730, 'P': 1},
    100237: {'Nombre': 'Bomba BCB012', 'H': 135, 'Q': 210, 'N': 1790, 'P': 1},
    100097: {'Nombre': 'Bomba BCS013', 'H': 112, 'Q': 198, 'N': 1735, 'P': 1},
    100598: {'Nombre': 'Bomba BCX014', 'H': 148, 'Q': 225, 'N': 1785, 'P': 1},
    100567: {'Nombre': 'Bomba BCX015', 'H': 122, 'Q': 205, 'N': 1775, 'P': 1},
    100871: {'Nombre': 'Bomba BCX016', 'H': 165, 'Q': 260, 'N': 1855, 'P': 2},
    100092: {'Nombre': 'Bomba BCX017', 'H': 119, 'Q': 215, 'N': 1780, 'P': 1}
}

```

100580:	{ 'Nombre': 'Bomba BCX018', 'H': 111, 'Q': 205, 'N': 1765, 'P': 1
100067:	{ 'Nombre': 'Bomba BCX019', 'H': 131, 'Q': 235, 'N': 1795, 'P': 1
100171:	{ 'Nombre': 'Bomba BCM020', 'H': 115, 'Q': 200, 'N': 1768, 'P': 1
100960:	{ 'Nombre': 'Bomba BCM021', 'H': 108, 'Q': 192, 'N': 1755, 'P': 1
100279:	{ 'Nombre': 'Bomba BCB022', 'H': 157, 'Q': 245, 'N': 1810, 'P': 2
100189:	{ 'Nombre': 'Bomba BCB023', 'H': 128, 'Q': 218, 'N': 1778, 'P': 1
100105:	{ 'Nombre': 'Bomba BCB024', 'H': 137, 'Q': 222, 'N': 1785, 'P': 1
100191:	{ 'Nombre': 'Bomba BCB025', 'H': 103, 'Q': 192, 'N': 1745, 'P': 1
100579:	{ 'Nombre': 'Bomba BCS026', 'H': 142, 'Q': 228, 'N': 1792, 'P': 1
100099:	{ 'Nombre': 'Bomba BCS027', 'H': 132, 'Q': 226, 'N': 1782, 'P': 1
100100:	{ 'Nombre': 'Bomba BCS028', 'H': 127, 'Q': 214, 'N': 1775, 'P': 1
100111:	{ 'Nombre': 'Bomba BCS029', 'H': 123, 'Q': 204, 'N': 1762, 'P': 1
100116:	{ 'Nombre': 'Bomba BCS030', 'H': 113, 'Q': 198, 'N': 1757, 'P': 1
100884:	{ 'Nombre': 'Bomba BCM031', 'H': 146, 'Q': 236, 'N': 1797, 'P': 1
100001:	{ 'Nombre': 'Bomba BCM032', 'H': 107, 'Q': 194, 'N': 1752, 'P': 1
100330:	{ 'Nombre': 'Bomba BCM033', 'H': 134, 'Q': 220, 'N': 1779, 'P': 1
100184:	{ 'Nombre': 'Bomba BCM034', 'H': 149, 'Q': 230, 'N': 1794, 'P': 1
100037:	{ 'Nombre': 'Bomba BCM035', 'H': 121, 'Q': 202, 'N': 1767, 'P': 1
100189:	{ 'Nombre': 'Bomba BCM036', 'H': 141, 'Q': 232, 'N': 1788, 'P': 1
100285:	{ 'Nombre': 'Bomba BCS037', 'H': 126, 'Q': 212, 'N': 1773, 'P': 1
100239:	{ 'Nombre': 'Bomba BCS038', 'H': 109, 'Q': 196, 'N': 1758, 'P': 1
100206:	{ 'Nombre': 'Bomba BCS039', 'H': 153, 'Q': 248, 'N': 1805, 'P': 2
100294:	{ 'Nombre': 'Bomba BCB040', 'H': 118, 'Q': 208, 'N': 1769, 'P': 1
100938:	{ 'Nombre': 'Bomba BCB041', 'H': 147, 'Q': 238, 'N': 1790, 'P': 1
100179:	{ 'Nombre': 'Bomba BCX042', 'H': 124, 'Q': 206, 'N': 1764, 'P': 1
100028:	{ 'Nombre': 'Bomba BCX043', 'H': 110, 'Q': 200, 'N': 1760, 'P': 1
100593:	{ 'Nombre': 'Bomba BCX044', 'H': 152, 'Q': 246, 'N': 1803, 'P': 2
100781:	{ 'Nombre': 'Bomba BCX045', 'H': 120, 'Q': 204, 'N': 1766, 'P': 1
100041:	{ 'Nombre': 'Bomba BCX046', 'H': 116, 'Q': 202, 'N': 1762, 'P': 1
100708:	{ 'Nombre': 'Bomba BCX046', 'H': 130, 'Q': 220, 'N': 1780, 'P': 1
100586:	{ 'Nombre': 'Bomba BCX048', 'H': 112, 'Q': 196, 'N': 1756, 'P': 1
100066:	{ 'Nombre': 'Bomba BCX049', 'H': 155, 'Q': 250, 'N': 1805, 'P': 2
100986:	{ 'Nombre': 'Bomba BCX050', 'H': 138, 'Q': 224, 'N': 1788, 'P': 1
150096:	{ 'Nombre': 'Bomba BCX051', 'H': 103, 'Q': 196, 'N': 1755, 'P': 1
150230:	{ 'Nombre': 'Bomba BCX052', 'H': 152, 'Q': 246, 'N': 1803, 'P': 2
150730:	{ 'Nombre': 'Bomba BCX053', 'H': 120, 'Q': 204, 'N': 1766, 'P': 1
150380:	{ 'Nombre': 'Bomba BCX054', 'H': 116, 'Q': 202, 'N': 1762, 'P': 1
150891:	{ 'Nombre': 'Bomba BCX055', 'H': 130, 'Q': 220, 'N': 1780, 'P': 1
150184:	{ 'Nombre': 'Bomba BCX056', 'H': 112, 'Q': 196, 'N': 1756, 'P': 1
150592:	{ 'Nombre': 'Bomba BCB057', 'H': 155, 'Q': 250, 'N': 1805, 'P': 2
150260:	{ 'Nombre': 'Bomba BCB058', 'H': 138, 'Q': 224, 'N': 1788, 'P': 1
150751:	{ 'Nombre': 'Bomba BCS059', 'H': 133, 'Q': 218, 'N': 1778, 'P': 1
150491:	{ 'Nombre': 'Bomba BCM060', 'H': 125, 'Q': 210, 'N': 1768, 'P': 1
150042:	{ 'Nombre': 'Bomba BCM061', 'H': 128, 'Q': 214, 'N': 1772, 'P': 1
150804:	{ 'Nombre': 'Bomba BCM062', 'H': 122, 'Q': 208, 'N': 1766, 'P': 1
150945:	{ 'Nombre': 'Bomba BCB063', 'H': 135, 'Q': 222, 'N': 1782, 'P': 1
150501:	{ 'Nombre': 'Bomba BCB064', 'H': 149, 'Q': 236, 'N': 1796, 'P': 1
150100:	{ 'Nombre': 'Bomba BCB065', 'H': 123, 'Q': 206, 'N': 1765, 'P': 1
150580:	{ 'Nombre': 'Bomba BCB066', 'H': 109, 'Q': 192, 'N': 1752, 'P': 1
150732:	{ 'Nombre': 'Bomba BCB067', 'H': 144, 'Q': 228, 'N': 1790, 'P': 1
150439:	{ 'Nombre': 'Bomba BCM068', 'H': 127, 'Q': 210, 'N': 1767, 'P': 1
150900:	{ 'Nombre': 'Bomba BCM069', 'H': 148, 'Q': 234, 'N': 1794, 'P': 1
150757:	{ 'Nombre': 'Bomba BCX070', 'H': 130, 'Q': 218, 'N': 1778, 'P': 1
150039:	{ 'Nombre': 'Bomba BCX071', 'H': 119, 'Q': 204, 'N': 1764, 'P': 1
150641:	{ 'Nombre': 'Bomba BCX072', 'H': 110, 'Q': 196, 'N': 1756, 'P': 1
150662:	{ 'Nombre': 'Bomba BCX073', 'H': 158, 'Q': 252, 'N': 1807, 'P': 2

```

150868: {'Nombre': 'Bomba BCX074', 'H': 131, 'Q': 222, 'N': 1781, 'P': 1}
150081: {'Nombre': 'Bomba BCX075', 'H': 121, 'Q': 212, 'N': 1771, 'P': 1}
200863: {'Nombre': 'Bomba BCE076', 'H': 152, 'Q': 244, 'N': 1801, 'P': 2}
200984: {'Nombre': 'Bomba BCE077', 'H': 140, 'Q': 228, 'N': 1790, 'P': 1}
200259: {'Nombre': 'Bomba BCE078', 'H': 125, 'Q': 214, 'N': 1774, 'P': 1}
200009: {'Nombre': 'Bomba BCE079', 'H': 115, 'Q': 204, 'N': 1764, 'P': 1}
200643: {'Nombre': 'Bomba BCX080', 'H': 183, 'Q': 200, 'N': 1760, 'P': 1}
200973: {'Nombre': 'Bomba BCX081', 'H': 190, 'Q': 330, 'N': 1789, 'P': 1}
200168: {'Nombre': 'Bomba BCX082', 'H': 185, 'Q': 216, 'N': 1776, 'P': 1}
200865: {'Nombre': 'Bomba BCX083', 'H': 205, 'Q': 340, 'N': 2001, 'P': 2}
200096: {'Nombre': 'Bomba BCX084', 'H': 193, 'Q': 332, 'N': 1778, 'P': 1}
200997: {'Nombre': 'Bomba BCM085', 'H': 188, 'Q': 240, 'N': 1795, 'P': 1}
200775: {'Nombre': 'Bomba BCM086', 'H': 200, 'Q': 210, 'N': 1770, 'P': 1}
200895: {'Nombre': 'Bomba BCM087', 'H': 197, 'Q': 212, 'N': 1772, 'P': 1}
200866: {'Nombre': 'Bomba BCM088', 'H': 186, 'Q': 306, 'N': 1766, 'P': 1}
200543: {'Nombre': 'Bomba BCX089', 'H': 205, 'Q': 346, 'N': 1997, 'P': 3}
200612: {'Nombre': 'Bomba BCX090', 'H': 182, 'Q': 232, 'N': 1788, 'P': 1}
}

200098: {'Nombre': 'Bomba BCX091', 'H': 208, 'Q': 224, 'N': 1780, 'P': 1}
205688: {'Nombre': 'Bomba BCX092', 'H': 190, 'Q': 202, 'N': 1765, 'P': 1}
205804: {'Nombre': 'Bomba BCX093', 'H': 147, 'Q': 300, 'N': 1790, 'P': 1}
205437: {'Nombre': 'Bomba BCX094', 'H': 122, 'Q': 208, 'N': 1767, 'P': 1}
205421: {'Nombre': 'Bomba BCB095', 'H': 210, 'Q': 200, 'N': 1760, 'P': 1}
205095: {'Nombre': 'Bomba BCE096', 'H': 141, 'Q': 228, 'N': 1792, 'P': 1}
205886: {'Nombre': 'Bomba BCB097', 'H': 129, 'Q': 214, 'N': 1776, 'P': 1}
205340: {'Nombre': 'Bomba BCB098', 'H': 210, 'Q': 244, 'N': 1801, 'P': 2}
205877: {'Nombre': 'Bomba BCB099', 'H': 135, 'Q': 222, 'N': 1781, 'P': 1}
205055: {'Nombre': 'Bomba BCB100', 'H': 130, 'Q': 218, 'N': 1777, 'P': 1}
}

# Valores calculados para la segunda bomba
H2_calculado = H2
Q2_calculado = Q2
N2_calculado = N2
P2_calculado = P2

# Umbral de similitud (ajusta según tus criterios)
umbral_similitud = 10

# Lista para almacenar las bombas recomendadas
bombas_recomendadas = []

# Comparar los valores calculados con el catálogo de bombas
for bomba, parametros in catalogo_bombas.items():
    similitud = abs(parametros['H'] - H2_calculado) + abs(parametros['Q'] -
    if similitud <= umbral_similitud:
        bombas_recomendadas.append(bomba)

# Mostrar las bombas recomendadas
if bombas_recomendadas:
    print("Bombas recomendadas:")
    for bomba in bombas_recomendadas:
        propiedades = catalogo_bombas[bomba]
        print(f'Bomba ID: PUMP - {bomba}')
        print(f'Nombre: {propiedades["Nombre"]}')
        print(f'H: {propiedades["H"]}', "m")
        print(f'Q: {propiedades["Q"]}', "m3/s")
        print(f'N: {propiedades["N"]}', "rpm")
        print(f'P: {propiedades["P"]}', "kW")
        print('---')
else:
    print("No se encontraron bombas recomendadas para los valores calculados")

```

Asegúrate de que tengas instalado Python para ejecutar este código en tu entorno de desarrollo.

Rege

A continuación, se presentan dos ejemplos de como es el funcionamiento de este trabajo.

Ejemplo 1.

```
BIENVENIDO.

Hola, tengo el proposito de ayudarle a elegir la bomba mas adecuada.

Ingrese la cabeza de la bomba de referencia (H1) en m: 160
Ingrese el caudal de la bomba de referencia (Q1) en m3/s: 300
Ingrese las RPM de la bomba de referencia (N1) en rpm: 1750
Ingrese la potencia de la bomba de referencia (P1) en kW: 20
Seleccione el parámetro secundario a ingresar para la segunda bomba (caudal,
potencia, RPM o cabeza): rpm
Ingrese las RPM de la bomba nueva (N2) en rpm: 2000
Valores de la segunda bomba:

Cabeza (H2): 208.97959183673467 m
Caudal (Q2): 342.85714285714283 m3/s
RPM (N2): 2000.0 rpm
Potencia (P2): 29.85422740524781 kW

Bombas recomendadas:
Bomba ID: PUMP - 200865
Nombre: Bomba BCX083
H: 205 m
Q: 340 m3/s
N: 2001 rpm
P: 28.4 kW
```

Ejemplo 2.

```
BIENVENIDO.

Hola, tengo el proposito de ayudarle a elegir la bomba mas adecuada.

Ingrese la cabeza de la bomba de referencia (H1) en m: 118
Ingrese el caudal de la bomba de referencia (Q1) en m3/s: 212
Ingrese las RPM de la bomba de referencia (N1) en rpm: 1762
Ingrese la potencia de la bomba de referencia (P1) en kW: 20
Seleccione el parámetro secundario a ingresar para la segunda bomba (caudal,
potencia, RPM o cabeza): cabeza
Ingrese la cabeza de la bomba nueva (H2) en m: 118
Valores de la segunda bomba:

Cabeza (H2): 118.0 m
Caudal (Q2): 212.0 m3/s
RPM (N2): 1762.0 rpm
Potencia (P2): 20.0 kW

Bombas recomendadas:
Bomba ID: PUMP - 100294
Nombre: Bomba BCB040
H: 118 m
Q: 208 m3/s
N: 1769 rpm
P: 17.1 kW
---
```

```
Bomba ID: PUMP - 100041
Nombre: Bomba BCX046
H: 116 m
Q: 202 m3/s
N: 1762 rpm
P: 17.2 kW
---
Bomba ID: PUMP - 150380
Nombre: Bomba BCX054
H: 116 m
Q: 202 m3/s
N: 1762 rpm
P: 17.2 kW
---
Bomba ID: PUMP - 150804
Nombre: Bomba BCM062
H: 122 m
Q: 208 m3/s
N: 1766 rpm
P: 17.6 kW
---
Bomba ID: PUMP - 150039
Nombre: Bomba BCX071
H: 119 m
Q: 204 m3/s
N: 1764 rpm
P: 17.4 kW
---
Bomba ID: PUMP - 150081
Nombre: Bomba BCX075
H: 121 m
Q: 212 m3/s
N: 1771 rpm
P: 17.1 kW
```

V. Lecciones Aprendidas

En la ingeniería mecánica se deben realizar cálculos extenuantes en los procesos de diseño lo que dificulta este procedimiento, esto no solo en el tema de hidráulica como son las bombas centrífugas si no en todos los aspectos en cuanto a esta disciplina, por eso al poder implementar la ingeniería de prompts ayuda a minimizar los tiempos de diseño, además ayuda a encontrar valores con una alta precisión lo cual es de suma importancia para garantizar el funcionamiento de estos dispositivos. Cabe resaltar que para poder llegar a ese punto de precisión es necesario entender el funcionamiento de la IA, ya que de eso depende la certeza de las respuestas que esta proporcione, sin dejar a un lado la importancia de tener los conocimientos necesarios en el tema que se desea estudiar.

VI. Conclusiones

Los resultados que se obtuvieron en este trabajo fueron los esperados, ya que gracias a la ayuda de ChatGPT se logró crear un código que ayudara a un ingeniero a encontrar una bomba por medio de la ley de similitudes, claro está que en el proceso se presentaron diferentes dificultades, pero al realizar las correcciones en cuanto a la forma de solicitar información a la IA y la reestructuración del código general se logró implementar la ingeniería de prompts en un tema de alta complejidad de ingeniería mecánica.

Camilo Andrés Bohórquez Piñeros estudiante de Ingeniería Mecánica.

