**PROBLEMÁTICA**

¿Qué deseas predecir o clasificar?

Se desea predecir el diseño de helicoidales para

¿Cuál es el objetivo de negocio o técnico?

Reducir los tiempo de simulación

¿Qué tipo de problema es?

* **Regresión** (predecir valores numéricos)
* **Clasificación** (etiquetas/categorías)
* **Clustering** (agrupar datos)
* **Detección de anomalías**, etc.

Objetivos

Alcances

Interesados

**INVESTIGACIÓN Y CONTEXTO**

Diseño generativo

Parametrizaciones de diseño

Enfoque analitico

Estados de esfuerzo y deformaciones

Cargas

Desplazamiento

Enfoque de simulación

Simulación para ampliar los datos y la predicción es correcta. Para ellos se debe generar datos con CAE.Integrar por medio de python datos de simulación

* Datos CKD formato CAD
* Datos iniciales de arranque
* Simulaciones CAE diseños preliminares

Parámetros

Software CAD - CAE

Solid EDGE

Ansys

Parametrización de manufactura

Materiales

Procesos

Machine learning

Mapa probabilístico

Ejemplos de acción de la dopamina como la molécula encargada de la valoración en la experiencia para dar un campo de posibilidades con el fin de tomar una decisión realizar una acción en los contextos en que se desarrolle. Luego, empieza un proceso de optimización considerando la probabilidad, la magnitud y el momento en que una recompensa podría ocurrirpara tener cada vez mejores resultado como en el de el placer y las recompensas. Esta actividad temprana permite al cerebro construir un mapa predictivo. Refleja una estructura y heterogeneidad en las respuestas neuronales que no se habían apreciado antes (abordar de manera holistica para extraer de forma individual en la experiencia los detalles y dar un campo de posibilidades para escoger)

Neuronas (parámetros o variables)

Exploración de las diferentes posibilidades (Escenarios hipotéticos)

Inferencia

Algoritmos

Predicciones

### **Recopilación: Recolección y comprensión de los datos**

* Obtén los datos desde fuentes disponibles (bases de datos, APIs, sensores, archivos CSV, etc.).
* Entiende qué representan las columnas o características.
* Explora la distribución de los datos y busca patrones.
* Estructura de datos
  + Establecer entradas y salidas

**Preprocesamiento de los datos**

* Limpieza: Eliminar valores nulos, duplicados o inconsistentes.
* Conversión: Transformar variables categóricas en numéricas (one-hot encoding, label encoding).
* Normalización/Estandarización: Escalar datos numéricos.
* División del dataset:

Entrenamiento (70–80%)

Validación (10–15%)

Prueba (10–15%)

### **Entrenamiento del modelo**

* Escoger el modelo adecuado (predictivo o generativo)
* Entrena el modelo con el conjunto de entrenamiento
* Ajusta hiperparámetros si es necesario (GridSearchCV, RandomSearch, etc.).
* Puedes usar validación cruzada para evitar sobreajuste o la verificación por expertos

### **Evaluación del modelo**

* Evalúa el rendimiento usando el conjunto de prueba con métricas apropiadas:  
  + **Clasificación**: accuracy, precision, recall, F1-score, AUC.
  + **Regresión**: MAE, MSE, RMSE, R².
* Revisa posibles errores sistemáticos (sesgo, varianza).

### **Mejoras y optimización**

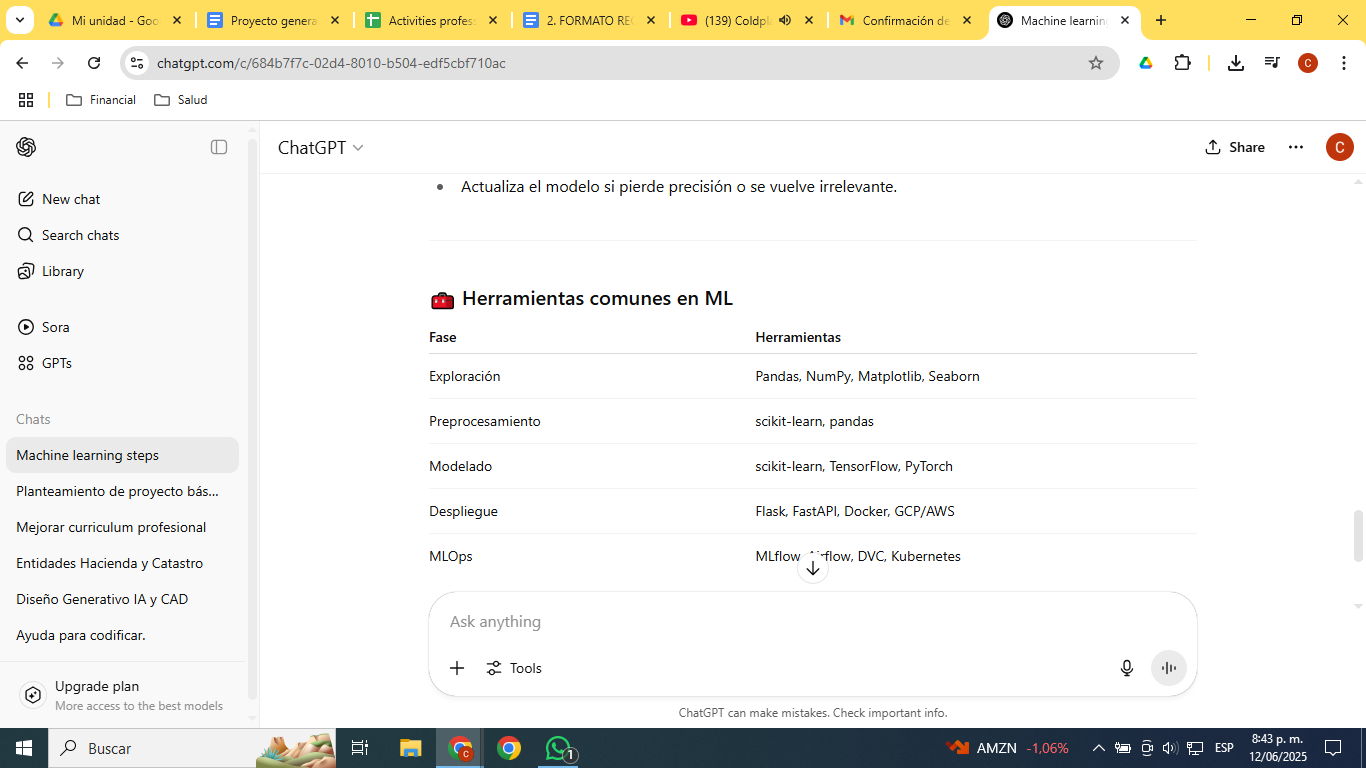
* Ajusta hiperparámetros (tuning).
* Prueba diferentes algoritmos o combinaciones (ensemble).
* Realiza ingeniería de características (feature engineering).
* Aumenta el tamaño de los datos (data augmentation, más muestras).

### **Despliegue del modelo**

* Empaqueta el modelo entrenado.
* Usa herramientas como:  
  + **Flask/FastAPI** para crear una API.
  + **Docker** para contenerización.
  + **GCP, AWS, Azure** para el despliegue en la nube.
* Monitoriza el rendimiento en producción.

### **Mantenimiento y retraining**

* Supervisa el modelo en producción.
* Reentrena periódicamente con nuevos datos (drift de datos).
* Actualiza el modelo si pierde precisión o se vuelve irrelevante.



Este es un excelente tema de trabajo de grado, relevante para la industria de autopartes y que combina diseño mecánico, simulación CAE y ciencia de datos.

Basándome en tu solicitud y los documentos académicos que has proporcionado (enfocados en metodologías de optimización paramétrica, KBE, algoritmos genéticos y aplicaciones de IA), la mejor manera de estructurar tu proyecto es utilizando

**Python** como el "cerebro" que gestiona la optimización (usando Machine Learning y algoritmos genéticos 111111111) y

**ANSYS** como la herramienta "muscular" para la simulación y generación de datos (CAE/FEM)2222.

Aquí tienes una estructura metodológica paso a paso para coordinar ambos softwares en tu trabajo de grado, enfocada en resortes helicoidales3333.

**Estructura Metodológica del Trabajo de Grado**

Esta metodología integra el modelado paramétrico en ANSYS con un algoritmo de optimización basado en ML (construido en Python) para crear un "gemelo digital" rápido (modelo sustituto) que pueda ser optimizado.

**1. Definición del Problema y Parametrización (ANSYS)**

El primer paso es definir el modelo base del resorte helicoidal y establecer qué parámetros controlarán su geometría.

* **Software:** ANSYS DesignModeler o SpaceClaim.
* **Acción:** Construye tu modelo CAD de resorte helicoidal. En lugar de usar valores fijos, define las dimensiones clave como parámetros (activando la casilla "P" en ANSYS Workbench).
* **Parámetros de Diseño (Variables de Entrada):** Basado en la teoría clásica de diseño de resortes (mencionada en documentos como Design-Optimization...pdf), tus variables de entrada principales serán:
  1. Diámetro del alambre (

d)4.

* 1. Diámetro medio de la espira (

D)5.

* 1. Número de espiras activas (

Na​ o n)6.

**2. Definición de Objetivos y Restricciones (ANSYS)**

Debes definir qué quieres minimizar/maximizar (Objetivos) y qué límites no puedes violar (Restricciones), extrayendo reglas de diseño (Ingeniería del Conocimiento o KBE 777777777).

* **Software:** ANSYS Mechanical (Análisis Estructural y/o Modal).
* **Objetivos (Salidas a Optimizar):**
  + **Minimizar la Masa (Peso):** Este es tu objetivo principal para el sector de autopartes8888. ANSYS calcula esto automáticamente.
  + **Maximizar el Factor de Seguridad:** (O minimizar el esfuerzo máximo de Von Mises)9.
* **Restricciones (Salidas de Control):**
  + **Esfuerzo Máximo:** El esfuerzo de corte (τtrab​) no debe superar el esfuerzo de fluencia (τperm​) del material (considerando factores de seguridad y corrección de Wahl, Ks​)10101010.
  + **Pandeo (Buckling):** El resorte no debe pandearse bajo la carga máxima. Deberás ejecutar un análisis de pandeo lineal (Linear Buckling) en ANSYS para asegurar que el multiplicador de carga sea > 1.
  + **Frecuencia Natural (Surging):** La frecuencia natural fundamental del resorte debe ser significativamente más alta (ej. >15x) que la frecuencia de operación para evitar la resonancia (surging). Esto requiere un análisis

**Modal** en ANSYS11111111.

* + **Restricciones Geométricas:** El diámetro exterior no debe superar el espacio disponible en el ensamblaje del vehículo.

**3. Generación de Datos (DOE) y Creación del Modelo Sustituto (ANSYS + Python)**

Aquí es donde ANSYS y Python se conectan. El objetivo es entrenar un modelo de Machine Learning que pueda *predecir* los resultados de ANSYS (que son lentos) de forma casi instantánea. Este modelo predictivo se llama

**Modelo Sustituto (Surrogate Model)** o Metamodelo12121212.

1. **Generar el Dataset (ANSYS):**
   * Usa el módulo **DesignXplorer** de ANSYS Workbench.
   * Configura un

**Diseño de Experimentos (DOE)**13131313. Un método robusto, mencionado en los archivos, es el

**Latin Hypercube Sampling (LHS)**14.

* + Indica a ANSYS que ejecute, por ejemplo, 200 combinaciones (puntos de diseño) de tus parámetros (d, D, N).
  + ANSYS ejecutará 200 simulaciones completas (Estructural, Modal y Pandeo) y generará una tabla de resultados.
  + **Exporta** esta tabla como un archivo .csv.

1. **Entrenar el Modelo ML (Python):**
   * **Software:** Python (usando librerías como Pandas, Scikit-learn, y TensorFlow/Keras 15151515151515151515151515151515).
   * **Acción:**
     + Carga el .csv de ANSYS en Python (usando Pandas).
     + Tus **Inputs (X)** son las columnas de parámetros (d, D, N).
     + Tus **Outputs (Y)** son las columnas de resultados (Masa, Esfuerzo, Frecuencia, Factor de Pandeo).
     + Entrena un modelo de regresión de ML (como una Red Neuronal 16161616, Bosque Aleatorio (Random Forest) 17, o Regresión de Procesos Gaussianos (Kriging) 18) para aprender la relación:

(d, D, N) -> (Masa, Esfuerzo, Frecuencia, Pandeo).

**4. Optimización con Algoritmos Genéticos (Python)**

Ahora que tienes un modelo sustituto (ML) que predice el comportamiento del resorte en milisegundos, puedes usar un algoritmo de optimización para explorar millones de posibilidades sin volver a abrir ANSYS. El archivo

content.pdf que subiste se enfoca específicamente en esta automatización usando **Algoritmos Genéticos (GA)**191919191919191919.

* **Software:** Python (usando librerías como DEAP, SciPy.optimize, o escribiendo un GA simple como el del content.pdf 20).
* **Acción:**
  1. **Población Inicial:** El GA crea una "población" de 1000 resortes "cromosoma" (listas de Python) con valores aleatorios para [d, D, N]21212121.
  2. **Función de Aptitud (Fitness Function):** Esta es la parte crucial. Tu función

*fitness* es tu modelo de ML entrenado22. Para cada uno de los 1000 resortes, el GA usa el modelo ML (

model.predict()) para obtener sus resultados (Masa, Esfuerzo, etc.)23.

* 1. **Selección y Evolución:** El GA "elimina" los diseños malos (los que fallan por pandeo, tienen esfuerzo muy alto, o son muy pesados) y "cruza" (recombina) 24los mejores (ligeros y fuertes), aplicando también "mutaciones" (pequeños cambios aleatorios)25.
  2. **Convergencia:** El algoritmo repite este ciclo por cientos de "generaciones" (iteraciones), evolucionando la población hacia un diseño óptimo que minimiza la masa cumpliendo todas las restricciones26.

**5. Validación de Resultados (ANSYS)**

El proceso finaliza volviendo a ANSYS para verificar que el resultado predicho por Python es correcto en la simulación de alta fidelidad.

1. Toma el diseño "Campeón" (los parámetros [d, D, N] óptimos) identificado por el algoritmo genético en Python.
2. Ingresa manualmente estos valores exactos en tus parámetros de ANSYS Workbench.
3. Ejecuta una simulación final de alta fidelidad.
4. Compara los resultados de ANSYS (ej. Masa: 1.5kg, Esfuerzo: 790 MPa) con los resultados *predichos* por tu modelo de ML (ej. Masa: 1.49kg, Esfuerzo: 795 MPa). Si están cercanos, tu modelo sustituto es válido.

Este resultado validado es la conclusión central de tu tesis: el diseño óptimo paramétrico encontrado mediante la integración de ANSYS y Machine Learning.

**PLANIFICACIÓN**

Objetivos específicos

* + Caracterización teórica resortes helicoidales en un estado de esfuerzo y deformaciones
  + Parametrizar las entradas y salidas
  + Datos de entrenamiento y validación con simulación integrando los entornos CAD y CAE por medio de python.
  + Proponer estructura en función de las métricas de eficacia del aprendizaje (generación y entrenamiento modelo).
  + Verificación

Actividades

Recursos

Tiempos

Cronograma

Responsabilidades Riesgos

**Desarrollo**

Documentación

**Evaluación**

Documento intermedio la semana 8

Avance de acuerdo a la propuesto en la planificación

Documento final semana 15

Lineamiento de trabajo de grado

Introducción . estado del arte

Planteamiento del problema

Objetivos

Conclusión de recomendaciones

**Lanzamiento**

Presentación final 16

Compartir trabajo de trabajo de grado para modelo de entrega final y información de trabajos

Bibliografía

*Google DeepMind*

*[1]* [*https://www.linkedin.com/advice/0/what-best-mechanical-design-optimization?lang=es&originalSubdomain=es*](https://www.linkedin.com/advice/0/what-best-mechanical-design-optimization?lang=es&originalSubdomain=es)

*[2]* [*https://www.simulexa.com/es\_es/explorando-la-optimizacion-topologica-y-parametrica-disenando-para-la-eficiencia/*](https://www.simulexa.com/es_es/explorando-la-optimizacion-topologica-y-parametrica-disenando-para-la-eficiencia/)

*[3]* [*https://www.goebel-engineering.com/es/modelacion-parametrica/amp/*](https://www.goebel-engineering.com/es/modelacion-parametrica/amp/)