Tarea (3 semanas)

- Desafío:
  - Expandir el repositorio del tutorial de langserve ya generado y usar como cadena un Agente de Retriever que consuma una instancia externa de una BBDD vectorial (sugerido Qdrant)
  - La bbdd vectorial debe estar poblada a priori (con lo que quieran)

1. Crear cuenta en Qdrant
2. Obtener URL y API-KEY qdrant
3. Correr carga de documentos en un script paralelo
4. Modificar código del servidor y cambiar el runnable
5. Deployar (agregar url y api-key como secretos)

```
url = "<---qdrant cloud cluster url here --->"
api_key = "<---api key here--->"
qdrant = Qdrant.from_documents(
    docs,
    embeddings,
    url=url,
    prefer_grpc=True,
    api_key=api_key,
    collection_name="my_documents",
)
```

Tarea (3 semanas)

🎯 Objetivo Principal

Implementar un sistema RAG completo utilizando el stack tecnológico 2025: LangChain + Qdrant Cloud + OpenAI, aplicando estrategias de chunking inteligente y evaluación rigurosa del sistema.

📚 Paso 1: Preparación de Fuentes de Datos

Requisitos mínimos:

- Seleccionar 2-3 fuentes de datos textuales de diferentes tipos:
  - Documentos PDF, Word (.docx) o archivos de texto (.txt)
  - Contenido temático coherente (ej: documentación técnica, libros, artículos especializados, a su elección)

Tarea (3 semanas)

🔧 Paso 2: Procesamiento y Chunking Estratégico

Implementar:

1. Análisis del contenido: Determinar el tipo de documento y estructura.
2. Selección de estrategia: Elegir entre las técnicas vistas en clase:
   - RecursiveCharacterTextSplitter (baseline)
   - SemanticChunker (avanzado)
   - Chunking específico por tipo de documento o otro
3. Configuración optimizada: Definir chunk_size, chunk_overlap, separadores.

Tarea (3 semanas)

Paso 3: Indexación en Qdrant Cloud

Crear vector store robusto:

1. Configurar colección Qdrant con nombre descriptivo
2. Generar metadata enriquecida para cada chunk:
3. Indexar con embeddings OpenAI (text-embedding-3-large o small), seleccionen la cantidad de dimensiones.
4. Mostrar el Qdrant con la información cargada.

Tarea (3 semanas)

🤖 Paso 4: Implementación RAG

Sistema RAG funcional con:

1. Retriever configurado con top-k apropiado
2. Prompt template optimizado siguiendo las mejores prácticas del curso
3. Chain RAG completo usando LangChain LCEL o Langgraph

Tarea (3 semanas)

🧪 Paso 5: Evaluación Sistemática

Crear datasets de evaluación:

Set de Preguntas Respondibles (10-15 preguntas):

- Preguntas que SÍ pueden responderse con la información indexada

Set de Preguntas No Respondibles (5-10 preguntas):

- Preguntas sobre temas NO cubiertos en los documentos
- Evaluar que el sistema responda "No tengo información suficiente"

Tarea (3 semanas)

Detalles de entrega:

- Deadline: Domingo 28 de Diciembre a las 23:59
- Entregables:
  a. URL del servicio deployado en langserve, debe tener acceso al playground para hacer preguntas
  b. Archivo con código de carga de documentos a qdrant, comentando estrategia a utilizar y razón.
  c. Set de preguntas respondibles y no respondibles