Contenido

1. Descripción del sistema

El proyecto Capstone finalizado consiste en un **Sistema RAG Multi-Agente con búsqueda híbrida**. El sistema ofrece acceso unificado a la documentación dispersa de un proyecto de software, permitiendo consultas funcionales, técnicas o de gestión, conceptuales y de contenido con alta precisión.

A diferencia de la propuesta de la memoria inicial, la solución final no solo clasifica por área, sino que implementa una arquitectura dual:

1. **Búsqueda Semántica:** Para preguntas conceptuales, utilizando embeddings y bases de datos vectoriales.
2. **Búsqueda Léxica:** Para consultas técnicas precisas (nombres de variables, logs, códigos de error), utilizando algoritmos de búsqueda exacta en la estructura de archivos.

El sistema incorpora además un **Agente Sintetizador** capaz de fusionar respuestas de múltiples fuentes y un pipeline ETL robusto para normalizar fuentes heterogéneas.

2. Componentes y vinculación con el temario

A continuación, se detalla cada módulo de la solución final y su justificación teórica basada en el Máster Desarrollador 10x de IIA:

A. Entorno de desarrollo y metodología

Se ha utilizado un enfoque de "Exponential Programming" para desarrollar una solución compleja partiendo de un conocimiento base de Python.

- **Implementación:** Configuración de entorno virtual, gestión de secretos (`.env`), y uso intensivo de **GitHub Copilot** para la generación de lógica y refactorización de código.
- **Vinculación temario:**
  - **Módulo 0 (Introducción a Python):** Uso de estructuras de datos (Listas, Diccionarios) y manejo de excepciones (0.3, 0.4).
  - **Módulo 1 y 8 (Exponential Programming):** Uso de **Chat** y **Autocomplete** de GitHub Copilot para acelerar el desarrollo (1.3, 1.6) y refactorización de código asistida (8.9).

B. Ingesta y procesamiento de datos (ETL)

Se ha desarrollado una herramienta propia, **`doc_to_md.py`**, que actúa como pipeline ETL robusto.

- **Implementación:** Script que ingesta documentos heterogéneos (Word, PDF, Excel), limpia ruido (cabeceras, pies de página mediante Regex) y estandariza todo a formato Markdown para maximizar la legibilidad por parte del LLM.
- **Vinculación temario:**
  - **Módulo 0 (Bibliotecas):** Uso de `pandas` para procesar excels (0.9) y librerías externas.
  - **Módulo 2 (EDA):** Limpieza y preparación de datos no estructurados (texto) para asegurar la calidad de la recuperación (2.8).

C. Base de datos vectorial y optimización

Almacenamiento eficiente de vectores para la ruta semántica.

- **Implementación:** Uso de **ChromaDB** para indexar los **chunks** de documentación. Se ha implementado un **Patrón Singleton** en la conexión a la base de datos para optimizar la latencia y evitar recargas innecesarias.
- **Vinculación temario:**
  - **Módulo 4 (Modelos de Lenguaje):** Uso de embeddings (`text-embedding-3-small`) para representación vectorial.
  - **Módulo 6 (RAG):** Implementación de almacenamiento vectorial persistente y estrategias de **chunking** (6.2).

D. Agente Orquestador (Clasificación Dual)

El "cerebro" del sistema ha evolucionado de un clasificador simple a uno bidimensional.

- **Implementación:** Un agente que clasifica la intención del usuario en dos ejes simultáneos:
  1) **Categoría:** Funcional, Técnica o Gestión.
  2) **Tipo de Búsqueda:** Semántica (conceptos) o Léxica (términos exactos).
- **Vinculación temario:**
  - **Módulo 5 (Prompting):** Diseño de prompts complejos (**Chain-of-Thought** implícito) para clasificación estructurada (5.3, 5.9).
  - **Módulo 6 (Agentes):** Lógica de enrutamiento condicional y toma de decisiones autónoma (6.5).

E. Agentes Especialistas y Búsqueda Híbrida

El núcleo de recuperación se ha desdoblado para mayor precisión.

- **Implementación:**
  1) **Ruta Semántica:** Agentes (Técnico, Funcional, Gestión) consultan ChromaDB filtrando por metadatos.
  2) **Ruta Léxica:** Algoritmos de búsqueda de texto exacto (Regex) sobre los archivos Markdown cuando se detectan nombres de variables o códigos específicos.
- **Vinculación temario:**
  - **Módulo 6 (RAG y Agentes):** Arquitecturas multi-agente y uso de herramientas (**tools**) específicas por agente (6.8).
  - **Módulo 0 (Python):** Manipulación avanzada de cadenas y expresiones regulares para la búsqueda léxica.

F. Agente Sintetizador

Componente añadido para manejar la complejidad de múltiples respuestas, especialmente en búsquedas léxicas transversales.

- **Implementación:** Un agente final que recibe los inputs de los agentes especialistas, elimina redundancias y redacta una respuesta única y coherente para el usuario.
- **Vinculación temario:**
  - **Módulo 5 (Prompting):** Técnicas de resumen y síntesis de información mediante LLMs (5.4).

G. Interfaz de usuario

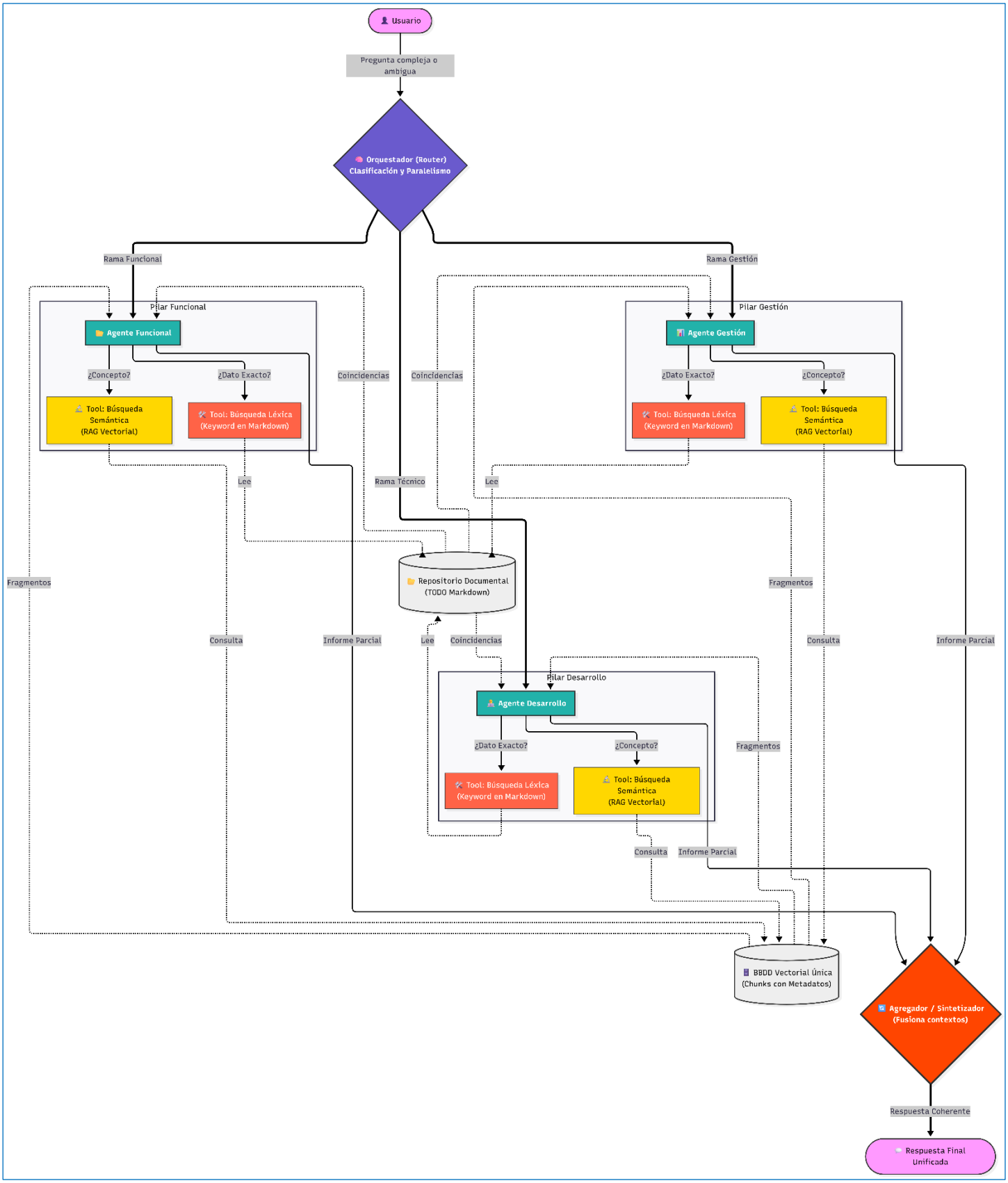
Frontend interactivo para consumo del servicio.

- **Implementación:** Aplicación web con **Gradio** que incluye controles avanzados:
  1) **Categoría:** Visualización de la categoría detectada.
  2) **Fuentes:** Citación de fuentes consultadas (transparencia).
- **Vinculación temario:**
  - **Módulo 5 (Gradio):** Creación de interfaces de chat con gestión de estado y parámetros adicionales (5.6).
  - **Módulo 9 (Software 3.0):** Desarrollo orientado al usuario final, ocultando la complejidad del modelo probabilístico (9.2).

3. Diagrama de Arquitectura Final

El flujo de información en la solución final es el siguiente:

1. **Entrada:** El usuario introduce una pregunta.
2. **Agente Orquestador:** Clasifica la Categoría y decide entre Búsqueda Semántica o Léxica.
3. **Ejecución: Agentes específicos**
   - *Semántica:* Un agente específico consulta vectorial en ChromaDB.
   - *Léxica:* Todos los agentes específicos realizan un barrido de texto en la estructura de carpetas Markdown.
4. **Agente Sintetizador:** El Agente Sintetizador unifica los hallazgos.
5. **Salida:** Respuesta final con referencias a las fuentes.


Usuario
Pregunta compleja o ambigua
Orquestador (Router)
Clasificación y Paralelismo
Rama Funcional
Rama Técnico
Rama Gestión
Pilar Funcional
Agente Funcional
¿Concepto?
¿Dato Exacto?
Tool: Búsqueda Semántica (RAG Vectorial)
Tool: Búsqueda Léxica (Keyword en Markdown)
Pilar Gestión
Agente Gestión
¿Dato Exacto?
¿Concepto?
Tool: Búsqueda Léxica (Keyword en Markdown)
Tool: Búsqueda Semántica (RAG Vectorial)
Coincidencias
Coincidencias
Lee
Lee
Repositorio Documental (TODO Markdown)
Fragmentos
Fragmentos
Consulta
Informe Parcial
Lee
Coincidencias
Consulta
Informe Parcial
Pilar Desarrollo
Agente Desarrollo
¿Dato Exacto?
¿Concepto?
Tool: Búsqueda Léxica (Keyword en Markdown)
Tool: Búsqueda Semántica (RAG Vectorial)
Fragmentos
Consulta
Informe Parcial
BBDD Vectorial Única (Chunks con Metadatos)
Agregador / Sintetizador (Fusiona contextos)
Respuesta Coherente
Respuesta Final Unificada

4. Documentación del proyecto

La documentación del proyecto está publicada en:

- **GitHub** https://github.com/arodriguezminguela/Capstone-ScanGestor

Dentro del proyecto, en la ruta "/doc/doc_capstone/" se encuentran los documentos técnicos y de desarrollo del proyecto, pero también está el video de la presentación del proyecto:

- **Presentación** Proyecto Capstone - Presentación.mp4

Para cualquier duda o consulta estoy disponible en el siguiente correo electrónico:

- **Correo** angelrodriguezminguela@gmail.com