RAG Codebase Assistant

🤖 An intelligent code analysis tool that uses Google Cloud's Vertex AI and RAG (Retrieval-Augmented Generation) to help you understand and interact with your codebase.

Overview

This project provides two main components:

1. **Python Script (`rag_codebase_local.py`)**: A standalone application for analyzing local code repositories
2. **Jupyter Notebook (`rag_codebase.ipynb`)**: An interactive notebook for analyzing GitHub repositories

Both components use Google Cloud's Vertex AI RAG Engine to create an AI assistant that can understand and answer questions about your code.

Features

- 🔍 Analyze both local and remote (GitHub) code repositories
- 🧠 Use AI to understand and explain code functionality
- 💬 Interactive Q&A about your codebase
- 📚 Support for multiple programming languages and file types
- 🌐 Integration with Google Cloud's Vertex AI
- 🔒 Secure handling of code and data

Prerequisites

- Python 3.x
- Google Cloud Project
- Google Cloud Storage bucket
- Appropriate Google Cloud credentials and permissions
- Required Python packages (installed automatically)

Configuration

Required Settings

```
{
    'PROJECT_ID': "your-project-id",    # Your Google Cloud Project ID
    'BUCKET_NAME': "your-bucket-name",  # Your Google Cloud Storage bucket
    'CAMINHO_CODIGO': "./meu_codigo",   # Path to your code directory
}
```

Optional Settings

- `LOCATION`: Google Cloud region (default: "us-central1")
- `PASTA_GCS`: Folder in GCS bucket (default: "codigo-para-analise")
- `TAMANHO_MAX_MB`: Maximum file size to process (default: 10MB)
- `MODELO_EMBEDDING`: Text embedding model (default: "publishers/google/models/text-embedding-005")
- `MODELO_IA`: AI model for responses (default: "gemini-2.5-flash")

Supported File Types

The tool supports a wide range of file types, including:

- Programming Languages: `.py`, `.java`, `.js`, `.ts`, `.go`, `.c`, `.cpp`, etc.
- Web Technologies: `.html`, `.css`, `.vue`, `.jsx`, `.tsx`
- Documentation: `.md`, `.txt`, `.rst`
- Configuration: `.yaml`, `.json`, `.toml`, `Dockerfile`
- And many more! (See full list in code)

Usage

Local Code Analysis (`rag_codebase_local.py`)

1. Configure your Google Cloud settings in the script
2. Run the script:

```
python rag_codebase_local.py
```

3. The script will:
    - Upload your code to Google Cloud Storage
    - Create a RAG corpus from your code
    - Start an interactive Q&A session

GitHub Repository Analysis (`rag_codebase.ipynb`)

1. Open the notebook in Jupyter/Colab
2. Configure your settings
3. Run each cell sequentially
4. Use the interactive Q&A section to analyze the code

Security Notes

- 🔐 Your code is uploaded to your private Google Cloud Storage bucket
- 👀 Access is controlled by your Google Cloud credentials
- 🗑️ Option to clean up resources after analysis
- ⚠️ Be careful with sensitive code and credentials

Contributing

Feel free to:

- Report issues
- Suggest improvements
- Submit pull requests

License

See license file for details.

Made with ♥ using Google Cloud Vertex AI