

# Agentes Autônomos – Relatório da Atividade Extra

---

Rodrigo Mibielli Peixoto – 07/10/2025

Synapse 7

## 1. Contextualização

O presente trabalho descreve um sistema de inteligência artificial que possui a capacidade de analisar qualquer arquivo nos formatos CSV e XLS e que também é capaz de gerar gráficos e apresentar conclusões.

Também apresentamos qual foi a *framework* escolhida, os desafios encontrados (e suas respectivas soluções) e por fim concluímos com *insights* obtidos ao realizar este trabalho.

## 2. Requisitos do Sistema

Para acessar o sistema *online*, o único requisito consiste em possuir um navegador *web* instalado e acesso à Internet.

Para rodar o sistema localmente, além dos mesmos requisitos para o acesso *online*, adicionalmente é necessário ter a linguagem *Python 3.11+* instalada em sua máquina. Além disso, o sistema utiliza a LLM do *Google Gemini*, o *gemini-2.5-flash-lite*. Por isso, é necessário ter em mãos a chave da API de acesso (no sistema *online* não é necessário configurar nada). Esta chave deve ser atribuída na variável `GOOGLE_API_KEY` localizada no arquivo `.env`, na base do projeto.

## 3. Acesso ao Sistema

É possível acessar o sistema de forma *online* ou baixar o código-fonte e executá-lo na máquina local.

### 3.1 – Acesso *online*

Abrir o seu navegador favorito com acesso à Internet e digitar o seguinte endereço:

<https://desafioextra-synapse7.streamlit.app/>

### 3.2 – Acesso local

Partindo da premissa que os requisitos do sistema são atendidos na seção anterior, para executá-lo localmente, a partir da base do código fonte do projeto, basta digitar o seguinte comando:

`./bin/run.sh (Linux)`

`./bin/run.bat (Windows)`

Em seguida, abra o navegador *web* e coloque o endereço informado pelo sistema (geralmente o endereço é <http://localhost:8501/>)

## 4. ***Frameworks* escolhidos**

Neste projeto utilizamos o *LangChain* como *framework* principal para orquestração do agente de IA. O *LangChain* foi escolhido devido à sua flexibilidade para integrar LLMs, ferramentas de execução *Python* e memória.

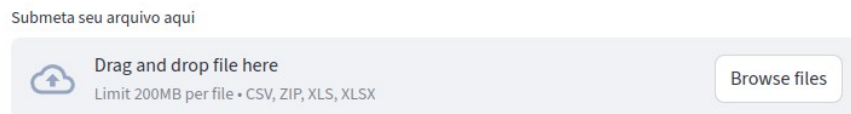
A interface com o usuário foi construída em *Streamlit*, permitindo upload de arquivos CSV ou XLS (ou arquivos ZIPs contendo o arquivo CSV ou XLS) para posterior envio de perguntas em linguagem natural através de uma *interface* de *chat messages*.

## 5. Estrutura da Solução

A solução foi organizada em três camadas principais:

### 1. Interface (*Streamlit*)


- Upload de arquivos CSV. Excel ou até arquivo zipado contendo a planilha compactada (Figura 1):




**Figura 1: O sistema exibe uma interface para subir a planilha.**

- Campo de perguntas em linguagem natural (Figura 2):


Submeta seu arquivo aqui

 Drag and drop file here  
Limit 200MB per file • CSV, ZIP, XLS, XLSX

Browse files

 Kaggle - Credit Card Fraud.zip 69.2MB

×

 O dataframe foi carregado com sucesso. Seguem as primeiras linhas...

↓ 🔍 ↺

	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11
0	0	-1.3598	-0.0728	2.5363	1.3782	-0.3383	0.4624	0.2396	0.0987	0.3638	0.0908	-0
1	0	1.1919	0.2662	0.1665	0.4482	0.06	-0.0824	-0.0788	0.0851	-0.2554	-0.167	1
2	1	-1.3584	-1.3402	1.7732	0.3798	-0.5032	1.8005	0.7915	0.2477	-1.5147	0.2076	0
3	1	-0.9663	-0.1852	1.793	-0.8633	-0.0103	1.2472	0.2376	0.3774	-1.387	-0.055	-0
4	2	-1.1582	0.8777	1.5487	0.403	-0.4072	0.0959	0.5929	-0.2705	0.8177	0.7531	-0

Quantos atributos há nesta planilha? ➤

Figura 2: Após submeter o arquivo, pode-se perguntar ao agente sobre o seu conteúdo.

- Exibição de respostas textuais e gráficos (Figura 3):

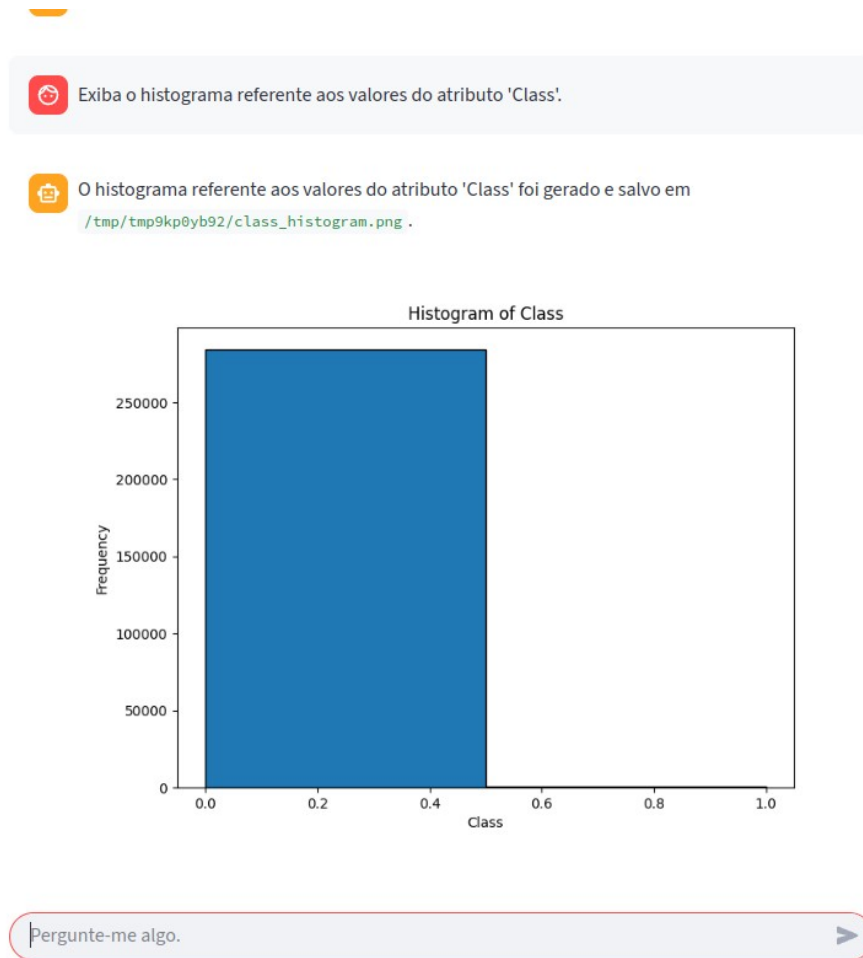


Figura 3: Resposta do Agente exibindo um gráfico

O arquivo *main.py* é o ponto de entrada no sistema e responde por um controle entre as camadas de exibição e de modelo, definindo o fluxo de exibição dos *widgets* na interface do usuário.

O arquivo *ui.py* é responsável por prover funções responsáveis por exibição dos componentes da interface gráfica e obtenção dos dados de entrada e exibição dos dados de saída.

## 2. Agente (LangChain)

- Configurado com Python REPL Tool e acesso a bibliotecas como Pandas e Matplotlib.
- Possui memória para manter o contexto das conversas.
- Responsável por interpretar perguntas e gerar código Python dinâmico para análise.

Os arquivo *agente.py* provê um agente responsável por interpretar os dados de um *dataframe* Pandas e responder ao usuário. Este *dataframe* é carregado ao subir a planilha no sistema e passado no construtor do agente.

O agente carrega a LLM e o *prompt* customizado para esta aplicação. No *prompt* define-se a estrutura de resposta ao usuário (formato JSON) que pode conter um caminho para o arquivo da imagem gerada.

No construtor deste agente também é passada a memória da conversa entre o usuário e o agente do tipo *StreamlitChatMessageHistory* de forma que o agente consiga “lembrar” do contexto da conversa.

O arquivo *ferramentas.py* possui ferramentas customizadas que auxiliam o agente em seu trabalho. Uma ferramenta criada, a *FetchTemporaryFilenameTool* é invocada pelo agente sempre que ele precisar obter um caminho temporário no disco rígido para armazenar as imagens geradas e que serão exibidas como respostas para o usuário.

### 3. Camada de Dados (Pandas DataFrame)

- Conversão automática dos CSV e XLS em DataFrame.
- Operações de análise exploratória (EDA).

Por “baixo dos panos”, o sistema invoca a ferramenta Python REPL Tool para gerar código Python com Pandas e/ou Matplotlib ou Seaborn para responder ao usuário.

Fluxo resumido:

Usuário → Streamlit → CSV → Pandas → Agente LangChain → Resposta/Gráfico

## 6. Exemplos de Perguntas e Respostas

### Exemplos 1 – Tipos de dados

Abaixo seguem algumas imagens contendo perguntas do usuário e respostas do sistema via *chat*:

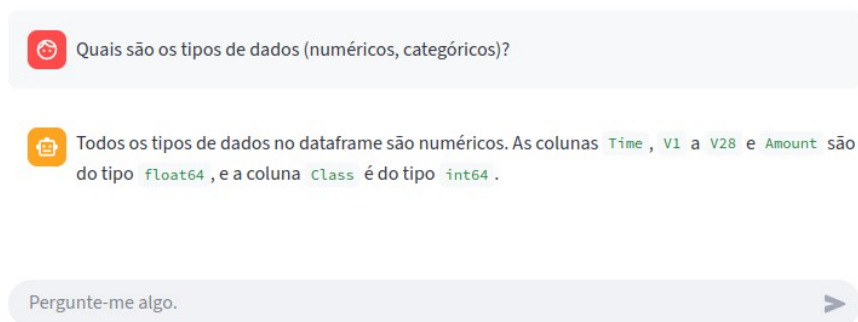


Figura 4: Pergunta sobre os tipos de dados da planilha.

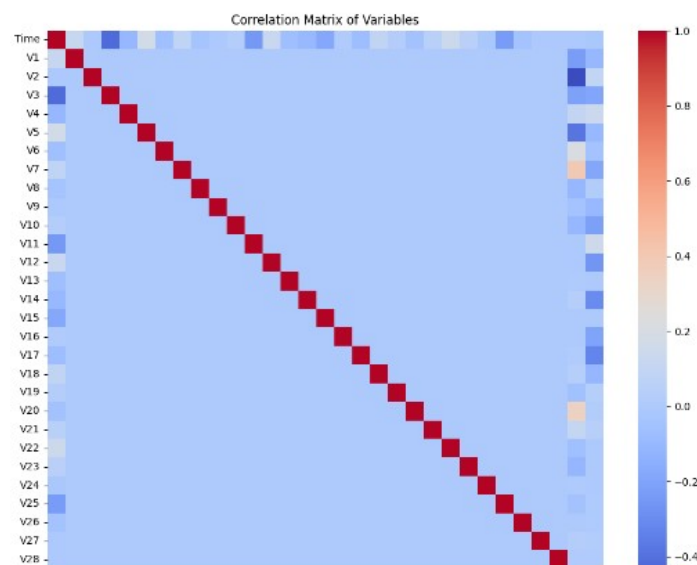




Existe alguma correlação entre as variáveis? Se sim, por favor exiba um gráfico de forma a facilitar o entendimento desta correlação.



Sim, existe correlação entre as variáveis. O gráfico abaixo, um mapa de calor da matriz de correlação, exibe a força e a direção das relações lineares entre os pares de variáveis. Cores mais quentes (vermelho) indicam correlação positiva, enquanto cores mais frias (azul) indicam correlação negativa. A intensidade da cor representa a magnitude da correlação.



Pergunte-me algo.



Figura 5: Resposta sobre correlação entre as variáveis com gráfico.

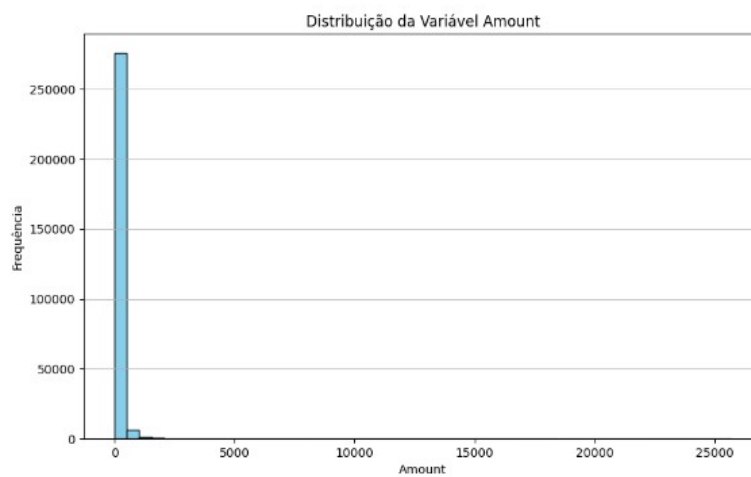
## Exemplo 2 – Distribuição de variável



Qual a distribuição da variável Amount?



A distribuição da variável 'Amount' é apresentada no gráfico abaixo. O histograma mostra a frequência de diferentes valores na coluna 'Amount'.



Pergunte-me algo.



Figura 6: Pergunta sobre distribuição com resposta gráfica.

### Exemplo 3 – Medidas estatísticas



Existem valor atípicos nos dados?



Sim, existem valores atípicos nos dados. A análise utilizando a regra de 1.5 vezes o Intervalo Interquartil (IQR) identificou valores atípicos nas seguintes colunas: 'Time', 'V1', 'V2', 'V3', 'V4', 'V5', 'V6', 'V7', 'V8', 'V9', 'V10', 'V11', 'V12', 'V13', 'V14', 'V15', 'V16', 'V17', 'V18', 'V19', 'V20', 'V21', 'V22', 'V23', 'V24', 'V25', 'V26', 'V27', 'V28' e 'Amount'. 27, 'V28' e 'Amount'.

Figura 7: Pergunta sobre valores atípicos dos dados.

### Exemplo 4 – Conclusões



Quais são as conclusões que você obteve a partir dos dados e das análises realizadas?



O dataset contém 284.807 transações com 31 features e nenhum valor ausente. Uma observação significativa é o extremo desequilíbrio de classes: existem 284.315 transações legítimas (Classe 0) e apenas 492 transações fraudulentas (Classe 1).

As principais conclusões das análises realizadas são:

- **Distribuição do Valor ('Amount'):** A feature 'Amount' apresenta uma distribuição altamente assimétrica, com uma média de 88.35 e uma mediana de 22.00. O desvio padrão é de 250.12 e a variância é de 62560.07, indicando uma grande dispersão nos valores das transações, especialmente para as legítimas. Um boxplot comparando 'Amount' por 'Class' evidencia diferenças, sugerindo que as transações fraudulentas podem ter padrões de valor distintos.
- **Padrões Temporais:** A análise das transações fraudulentas ao longo do tempo (visualizada em um gráfico de dispersão) indica que essas atividades não são distribuídas uniformemente. Elas tendem a se concentrar em períodos específicos e frequentemente envolvem valores de transação mais baixos, sugerindo padrões temporais que podem ser explorados para detecção.
- **Agrupamentos (Clustering):** Com base no gráfico de dispersão de 'Amount' versus 'Time' por 'Class', observam-se diferenças entre transações legítimas e fraudulentas. Enquanto as transações legítimas se espalham pelo eixo do tempo com valores variados, as transações fraudulentas mostram uma tendência a se agrupar em certos períodos e, muitas vezes, possuem valores menores. Essa distribuição diferencial sugere a existência de agrupamentos distintos nos dados.

Em resumo, os dados são caracterizados por um severo desequilíbrio de classes, uma distribuição assimétrica dos valores das transações e padrões temporais e potenciais

Pergunte-me algo.



Figura 8: Resposta do agente com suas observações em função das análises realizadas.

## 7. Considerações Finais

- O sistema demonstrou ser capaz de realizar análises exploratórias interativas em qualquer dataset CSV ou XLS;
- O uso de agentes autônomos permitiu respostas contextualizadas e geração automática de gráficos;

- O desafio reforçou a importância do uso de memória em agentes para manter coerência nas respostas;

O desafio também reforçou a importância do *prompt* para guiar as ações e estruturar a resposta do agente.

## 8. Informações Complementares

- Códigos-fonte completo (atividade\_extra.zip).

- Acesso do código fonte via Git:

[https://github.com/synapsesete/desafio\\_extra](https://github.com/synapsesete/desafio_extra)