


# Agente de IA para Cotações de Materiais

## Desenvolvido em Python

 Data: 02/05/2025

 Autor: André Luiz Gava Moraes

---

### 1. Introdução

Este projeto tem como objetivo a criação de um Agente de IA capaz de buscar e organizar cotações de materiais automaticamente. Ele fornecerá preços, fornecedores, links de compra e contatos, facilitando a pesquisa e tornando-a mais eficiente.

Com o uso de Python, o sistema será flexível o suficiente para ser desenvolvido tanto como aplicação web quanto aplicação desktop, dependendo da necessidade do usuário.

---

### 2. Objetivo

- ✓ Automatizar a busca de cotações de preços de materiais.
  - ✓ Facilitar a comparação entre fornecedores.
  - ✓ Integrar inteligência artificial para melhorar a coleta e análise dos dados.
  - ✓ Reduzir o tempo gasto com pesquisas manuais de preços.
- 

### 3. Tecnologias Utilizadas

● **Linguagem:** Python

● **Bibliotecas:**

- Requests / BeautifulSoup → Para web scraping e coleta de dados.
  - Selenium → Para interação com sites dinâmicos.
  - FastAPI / Flask → Para criação de API (caso seja web).
  - Pandas → Para manipulação e organização de dados.
  - OpenAI API (ChatGPT) → Para inteligência artificial e interpretação de texto.
-

## 4. Arquitetura do Sistema

### **Arquitetura Desktop (Local)**

#### **O que significa?**

Uma aplicação desktop é um programa que roda diretamente no computador do usuário, sem necessidade de conexão com a internet. Assim como programas comuns, como o Microsoft Word ou Excel, o usuário precisa baixar e instalar o software para usá-lo.

#### **Como funciona?**

- ◆ O usuário abre o programa instalado em seu computador.
- ◆ O sistema busca os preços de materiais automaticamente.
- ◆ Os dados são armazenados localmente ou exportados para um arquivo.

#### **Componentes principais:**

- ✓ Aplicação Python Standalone – Código executado diretamente no sistema do usuário.
- ✓ Interface Gráfica (PyQt ou Tkinter) – Para interação visual sem necessidade de navegador.
- ✓ Banco de Dados Local (SQLite) – Armazena os dados sem precisar de servidores externos.
- ✓ Web Scraping e APIs – Similar à versão web, mas tudo rodando localmente.
- ✓ Sem necessidade de hospedagem na nuvem – Não requer servidores externos.

#### **Vantagens:**

- ✓ Fácil de usar sem necessidade de internet.
- ✓ Menos complicado para criar, pois não precisa de servidores.
- ✓ O usuário tem controle total sobre os dados no próprio computador.

#### **Desvantagens:**

- ✗ Só pode ser utilizado no computador onde foi instalado.
  - ✗ Se precisar de atualizações, o usuário precisa baixar uma nova versão manualmente.
  - ✗ Não pode ser acessado de qualquer lugar ou dispositivo.
-

## **Arquitetura Web (Nuvem)**

### **O que significa?**

Uma aplicação web é um sistema que funciona dentro do navegador, como um site. Isso significa que ele pode ser acessado de qualquer dispositivo com internet, sem necessidade de instalar nada. Exemplos de serviços web são o Gmail, Google Docs e plataformas de compras online.

### **Como funciona?**

- ◆ O usuário acessa o sistema pelo navegador (Google Chrome, Firefox, etc.).
- ◆ O agente de IA faz a pesquisa de preços automaticamente.
- ◆ Os dados são salvos na nuvem e podem ser acessados de qualquer lugar.

### **Componentes principais:**

- ✓ Backend em Python (FastAPI ou Flask) – Gerencia requisições e processamento.
- ✓ Banco de Dados (PostgreSQL, SQLite ou MongoDB) – Armazena as informações coletadas.
- ✓ Serviço de Web Scraping (Requests, BeautifulSoup ou Selenium) – Busca preços e cotações online.
- ✓ Hospedagem na Nuvem (AWS, Azure ou Google Cloud, ou nuvem escolhida) – Permite que o sistema seja acessível de qualquer lugar.
- ✓ Interface Frontend (HTML/CSS + JS com Vue.js ou React) – Opcional, para exibição dos dados ao usuário.

### **Vantagens:**

- ✓ Pode ser acessado de qualquer lugar, sem necessidade de instalação.
- ✓ Atualizações são feitas automaticamente sem esforço do usuário.
- ✓ Pode permitir que várias pessoas acessem o sistema ao mesmo tempo.

### **Desvantagens:**

- ✗ Precisa de conexão com a internet para funcionar.
  - ✗ Exige um servidor (computador remoto) para armazenar os dados.
  - ✗ O desenvolvimento pode ser mais complexo do que uma versão desktop.
-

## 5. Comparação: Web vs Desktop

Critério	Web (Nuvem)	Desktop (Local)
Acessibilidade	Qualquer dispositivo com internet	Apenas no computador local
Facilidade de uso	Pode ser acessado via navegador	Precisa instalar o software
Manutenção	Atualizações automáticas no servidor	Usuário precisa atualizar manualmente
Banco de Dados	Armazenado na nuvem	Armazenado localmente
Complexidade	Maior (envolve servidores e APIs)	Menor (funciona localmente)

### 🔴 Conclusão para quem não entende programação:

- Aplicação Desktop: Funciona como um programa comum instalado no computador. Pode ser mais fácil de desenvolver, mas só funciona localmente.
- Aplicação Web: Funciona como um site, acessível de qualquer lugar. Permite que várias pessoas usem ao mesmo tempo, mas é mais difícil de criar porque precisa de servidores e internet.

✅ Se você quer algo simples e rápido, a versão desktop pode ser melhor.

✅ Se precisa de acessibilidade e escalabilidade, a versão web será mais poderosa.

---

## 6. Conclusão

Este agente de IA ajudará a automatizar a busca de materiais, tornando o processo mais rápido e eficiente. Com Python, o projeto pode ser desenvolvido tanto como desktop quanto web, dependendo da necessidade do usuário.