


Introdução

Problem Statement (Declaração do problema)	<p>Esta aplicação foi desenvolvida para facilitar o acesso rápido e prático às cotações atuais de pares de moedas estrangeiras e criptomoedas em relação ao real brasileiro (BRL). Ao invés de o usuário precisar acessar sites ou plataformas financeiras complexas, basta executar o programa, digitar o par de moedas desejado (como USD-BRL, EUR-BRL, BTC-BRL) e obter instantaneamente o valor de compra atualizado.</p> <p>Problema que resolve</p> <ul style="list-style-type: none">• Dificuldade de acesso rápido à cotação: Muitos usuários precisam consultar valores de câmbio de forma ágil, seja para viagens, investimentos ou comércio, mas acabam perdendo tempo navegando em diversos sites financeiros.• Complexidade técnica: Para quem não tem familiaridade com APIs financeiras, montar consultas e interpretar os dados pode ser complicado.• Automatização simples: Usuários que querem integrar consultas de câmbio em seus scripts ou sistemas podem usar essa aplicação como base para automatizar esse processo sem complicações.
Solução Proposta	<p>A aplicação oferece uma solução simples, eficiente e acessível para consultar cotações de moedas estrangeiras e criptomoedas em relação ao real brasileiro, utilizando uma interface minimalista baseada em terminal.</p> <p>Por meio de uma única interação — a digitação do par de moedas desejado — o usuário recebe imediatamente a cotação atualizada, obtida diretamente de uma API confiável e gratuita (economia.awesomeapi.com.br).</p> <p>Essa abordagem elimina a necessidade de buscar manualmente informações em diversos sites, reduzindo o tempo gasto e a complexidade técnica envolvida. Além disso, a aplicação pode ser facilmente integrada ou ampliada para outras funcionalidades, como atualização automática de cotações, alertas financeiros ou integração com sistemas maiores.</p> <p>Com isso, a solução oferece:</p> <ul style="list-style-type: none">• Agilidade na obtenção dos dados, fornecendo informações em tempo real.• Facilidade de uso, com uma interface de linha de comando simples e direta.• Flexibilidade, permitindo consultas personalizadas a qualquer par de moedas disponível na API.• Base para aprendizado e expansão, servindo como exemplo prático de consumo de APIs em Python.
Explicit Hypothesis	<p>A aplicação foi desenvolvida para ser leve, rápida e simples, funcionando diretamente via terminal e sem interface gráfica. Ela realiza uma única tarefa</p>

<p>(Hipóteses explícita)</p>	<p>de forma eficiente: consultar a cotação atual de um par de moedas por meio de uma API pública.</p> <p>Funcionamento passo a passo:</p> <ol style="list-style-type: none"> 1. Entrada do usuário O usuário é solicitado a digitar um par de moedas no formato MOEDA1-MOEDA2 (por exemplo: USD-BRL, EUR-BRL, BTC-BRL). 2. Requisição para a API O script monta uma URL com o par informado e faz uma requisição GET para a API https://economia.awesomeapi.com.br/last/. 3. Tratamento da resposta A resposta recebida da API é em formato JSON. O programa interpreta esse JSON e verifica se o par de moedas está presente na resposta. 4. Saída para o usuário <ul style="list-style-type: none"> Se o par de moedas for válido, o script exibe o valor de compra atual (bid) da moeda. Se o par não for encontrado, o usuário é informado com uma mensagem de erro clara. <p>Resultado esperado</p> <p>Ao final da execução, o resultado esperado é:</p> <p>Para um par de moedas válido (ex: USD-BRL):  A cotação do USD-BRL é: R\$ 5.3296</p> <ul style="list-style-type: none"> Para um par de moedas inválido ou inexistente (ex: BRL-AAA): Não encontramos a moeda solicitada. <p>Desempenho e eficiência</p> <ul style="list-style-type: none"> Baixo consumo de recursos: por ser uma aplicação de linha de comando, ela roda rapidamente, mesmo em máquinas modestas. Resposta imediata: o tempo de resposta depende apenas da conexão com a API (geralmente menos de 1 segundo). Sem necessidade de instalação de banco de dados ou servidores: é um script autônomo que só depende da internet e da biblioteca requests.
<p>Scope (Incluindo no Escopo)</p>	<p>A aplicação já está completamente funcional em sua proposta principal: consultar e exibir a cotação atual de um par de moedas. As seguintes funcionalidades estão implementadas e operacionais:</p> <p>Entrada dinâmica do usuário</p> <ul style="list-style-type: none"> O programa solicita que o usuário digite o par de moedas no formato MOEDA1-MOEDA2 (como USD-BRL, EUR-BRL, BTC-BRL, etc.). A entrada é flexível e permite múltiplas moedas suportadas pela API.

	<h2>Consumo de API em tempo real</h2> <ul style="list-style-type: none"> A aplicação realiza uma requisição GET à API pública AwesomeAPI - Economia. <p>O endpoint utilizado é: https://economia.awesomeapi.com.br/last/{MOEDA1}-{MOEDA2}</p> <h2>Processamento e validação da resposta</h2> <ul style="list-style-type: none"> O programa trata o JSON retornado pela API. Converte o nome do par de moedas para o formato correto exigido pela API (USD-BRL vira USDBRL). Verifica se o par solicitado existe na resposta. <h2>Exibição clara da cotação</h2> <p>Em caso de sucesso, o valor de compra da moeda (bid) é mostrado ao usuário de forma amigável. Exemplo: A cotação do USD-BRL é: R\$ 5.3296</p> <p>Em caso de erro (par inválido ou não suportado), uma mensagem clara é exibida: Não encontramos a moeda solicitada.</p> <h2>Ambiente isolado com Docker</h2> <ul style="list-style-type: none"> A aplicação já pode ser empacotada em uma imagem Docker, garantindo portabilidade e fácil execução em qualquer máquina com Docker instalado. Um Dockerfile funcional está disponível no projeto. <h2>Automação de build com pipeline CI</h2> <ul style="list-style-type: none"> Um pipeline básico de Integração Contínua (CI) usando GitHub Actions já está configurado. A cada push ou pull request, a imagem Docker é construída automaticamente, garantindo que o código esteja funcional.
<p>Scope (Não incluindo no Escopo)</p>	<h2>Funcionalidades ainda não implementadas</h2> <p>Embora a aplicação já cumpra seu objetivo principal de consultar a cotação de moedas, ainda existem funcionalidades que não foram implementadas e que podem ser consideradas em versões futuras para torná-la mais completa e robusta:</p> <hr/> <h2>Validação avançada da entrada do usuário</h2> <ul style="list-style-type: none"> Atualmente, o programa aceita qualquer entrada e só valida se o par existe após a chamada à API.

- Não há validação prévia para formatos incorretos, como **USDBRL** (sem hífen), **123-ABC**, ou campos vazios.

Suporte a múltiplos pares de moedas ao mesmo tempo

- A aplicação trata apenas **um par de moedas por execução**.
- Apesar da API suportar múltiplos pares separados por vírgula (ex: **USD-BRL, EUR-BRL**), o script atual não está preparado para lidar com mais de um par por vez.

Exibição de outras informações além da cotação de compra

- A API retorna vários dados úteis além do **bid** (como **ask, high, low, pctChange, timestamp**, etc.).
- No momento, a aplicação mostra apenas o valor de compra (**bid**), sem opção de visualizar mais detalhes.

Interface gráfica (GUI) ou interface web

- A aplicação é totalmente baseada em terminal (CLI).
- Ainda não há integração com interfaces gráficas ou frameworks web para uso mais intuitivo por usuários não técnicos.

Logs ou armazenamento de histórico

- Não há registro local das consultas feitas.
- Recursos como salvar o histórico de cotações ou exportar resultados (CSV, TXT, etc.) ainda não foram implementados.

Testes automatizados

- O projeto ainda **não possui testes automatizados** para validar o comportamento do código.
- Isso dificulta a verificação de erros e a garantia de funcionamento em futuras atualizações.

Deploy automático da imagem Docker (CD)

- A aplicação já conta com um pipeline de **integração contínua (CI)**, mas **ainda não faz deploy contínuo (CD)**.
- A imagem Docker ainda não é publicada automaticamente no Docker Hub ou executada em ambientes de produção.

Solução

Foi criada uma aplicação simples em **Python**, empacotada com **Docker** e integrada a um pipeline de **Integração Contínua (CI)** com GitHub Actions, com o objetivo de facilitar o acesso à **cotação atual de moedas estrangeiras e criptomoedas em relação ao real brasileiro (BRL)**.

A aplicação permite que o usuário, via terminal, consulte de forma prática o valor atual de compra de qualquer par de moedas suportado pela API pública da AwesomeAPI.

Além do script principal, o projeto conta com:

- Um **Dockerfile** para garantir portabilidade e facilidade de execução
 - Um arquivo **requirements.txt** para gerenciamento de dependências
 - Uma **pipeline automatizada** que realiza o build da imagem Docker a cada atualização no repositório
-

Como funciona

1. Interação com o usuário

Ao executar o script, o usuário é solicitado a informar o par de moedas que deseja consultar, no formato **MOEDA1-MOEDA2** (por exemplo, **USD-BRL**, **BTC-BRL**, **EUR-BRL**, etc.).

2. Requisição à API

O programa monta uma URL e faz uma requisição HTTP **GET** à AwesomeAPI de Moedas, que retorna um JSON com os dados da cotação.

3. Processamento da resposta

A resposta da API é verificada. Caso o par de moedas exista, o programa extrai o valor de compra (**bid**) e o exibe para o usuário.

Se a moeda não for encontrada, uma mensagem de erro é apresentada.

4. Execução via Docker (opcional)

Para garantir que a aplicação rode em qualquer sistema sem necessidade de instalar Python ou bibliotecas, é possível executar tudo dentro de um container Docker criado a partir do **Dockerfile**.

5. Pipeline CI (Integração Contínua)

Toda vez que o código é alterado e enviado para o repositório GitHub, uma pipeline automatizada:

- Faz o build da imagem Docker
 - Verifica se tudo está funcionando corretamente
 - Garante que o código esteja sempre em condições de uso
-

O que essa solução faz para resolver o problema

O problema:

Muitas pessoas precisam consultar cotações de moedas com rapidez e precisão, mas nem sempre têm acesso fácil a sites confiáveis, ou enfrentam interfaces lentas, complexas e poluídas. Além disso, para desenvolvedores ou sistemas que precisam de cotações programaticamente, usar APIs diretamente pode ser complicado.

A solução:

Esta aplicação resolve esse problema ao oferecer uma ferramenta:

- **Simples e rápida**, acessível diretamente pelo terminal
 - **Automatizada**, com código enxuto que consome dados em tempo real
 - **Portável**, graças ao uso de Docker
 - **Confiável**, por utilizar uma API atualizada e gratuita
 - **Segura e escalável**, com base em boas práticas como CI e controle de dependências
-

Como utilizar?

A seguir, você verá um exemplo realista de como utilizar esta solução para consultar a cotação de moedas estrangeiras de forma rápida, simples e confiável.

1. Configurando o ambiente

Você pode rodar a aplicação de duas formas:

Opção A — Usando Docker (recomendado)

Essa é a forma mais prática e portátil. Basta ter o Docker instalado.

Passos:

Clone o repositório do projeto:

```
git clone https://github.com/seu-usuario/seu-repo.git
cd seu-repo
```

Construa a imagem Docker:

```
docker build -t cotacao-app .
```

Opção B — Usando Python localmente

Ideal se você quiser rodar diretamente no seu sistema.

Pré-requisitos: Python 3.x instalado

Crie e ative um ambiente virtual:

```
python3 -m venv venv
source venv/bin/activate # Linux/macOS
venv\Scripts\activate    # Windows
```

Instale as dependências:

```
pip install -r requirements.txt
```

2. Executando a aplicação

Com Docker:

```
docker run -it cotacao-app
```

Com Python diretamente:

```
python cotacao.py
```

3. Exemplo realista de uso

Cenário:

Diego precisa saber quanto está o dólar hoje para fazer uma compra online em um site dos EUA.

Ele roda a aplicação e insere o par de moedas **USD-BRL**.

Entrada:

Digite o par de moedas (ex: USD-BRL, EUR-BRL, BTC-BRL): USD-BRL

Resultado esperado:

A cotação do USD-BRL é: R\$ 5.3287

4. Exemplo com erro (caso de uso inválido)

Entrada incorreta:

Digite o par de moedas (ex: USD-BRL, EUR-BRL, BTC-BRL): BRL

Resultado esperado:

Não encontramos a moeda solicitada.

Conclusão

Com essa aplicação, o usuário consegue consultar de forma rápida e segura o valor de compra de moedas em tempo real, sem precisar acessar sites ou lidar com dados confusos. Ela é ideal tanto para uso pessoal quanto para integrar em processos maiores que dependem de cotações atualizadas.