**1. Definição do Projeto**

* Objetivo: Automatizar o download, processamento e armazenamento de dados em arquivos CSV.
* Tecnologias: Python, Pandas, MySQL, CSV, Agendamento de Tarefas (Task Scheduler e Cron).

**2. Estrutura do Projeto**

projeto\_auto\_extrator/  
│── scripts/ → Código-fonte do projeto  
│── dados/ → Armazena os CSVs gerados  
│── logs/ → Armazena logs de execução  
│── config.yaml → Credenciais e configurações (usar variáveis de ambiente)  
│── main.py → Script principal  
│── requirements.txt → Dependências do projeto  
│── tests/ → Testes unitários e de integração

**3. Desenvolvimento**

**3.1 Conexão com o Banco de Dados**

* Criar função para conectar ao MySQL.
* Executar query para extrair dados já existentes ou realizar atualizações de acordo com a lógica de negócios.
* Segurança: .env para armazenar credenciais.

**3.2 Download dos Dados do Kaggle**

* Autenticação Kaggle: Utilizar a chave API do Kaggle para autenticação.
* Download do Dataset: Baixar o dataset desejado diretamente do Kaggle usando a biblioteca kaggle no Python.
* Descompactar Dados (se necessário): Caso o dataset esteja em formato comprimido (zip), descompactar e preparar os dados para manipulação.

**3.3 Manipulação dos Dados**

* Carregar os dados no Pandas.
* Aplicar transformações necessárias conforme os requisitos do projeto (limpeza de dados, renomeação de colunas, conversão de tipos, etc.).
* Validação: Verificar schema, tratar dados nulos e duplicados.

**3.4 Armazenamento no Banco de Dados**

* Criar ou atualizar tabela no MySQL com os dados extraídos do Kaggle.
* Inserção ou Atualização: Decidir se deve inserir dados novos ou atualizar dados existentes com base na lógica de negócios.
* Validação de Schema: Verificar se os dados estão no formato correto antes de realizar a inserção no banco.

**3.5 Download e Novo Processamento**

* Baixar os dados do MySQL após a inserção.
* Aplicar novas transformações conforme necessário.

**3.6 Exportação Final**

* Salvar os dados processados como CSV na pasta dados/ com timestamp.
* Garantir que o CSV esteja livre de inconsistências.

**3.7 Automação**

* Criar script de agendamento para execução automática do processo:
  + Windows: Task Scheduler
  + Linux: cron job
* Monitoramento: Registrar logs detalhados e configurar alertas (e-mail/Slack) para eventuais falhas.

**3.8 Logging e Monitoramento**

* Implementar logs estruturados usando o módulo logging.
* Configurar notificações automáticas em caso de falhas.

**5. Segurança e Configurações**

* Uso de variáveis de ambiente para credenciais (API Key do Kaggle e MySQL).
* Segurança de arquivos de configuração: Proteger arquivos como config.yaml para que não sejam expostos a ambientes não autorizados.
* Tratamento de Erros: Implementar tratamento robusto com try/except para falhas de conexão com Kaggle, MySQL ou durante manipulação de dados.