Ferramenta de ETL com Criptografia: Planilha Integra

Rui Carlos Poletti Junior [1], **Edgar Noda**[1]

[1]Instituto Federal de Educação, Ciência e Tecnologia de São Paulo - Câmpus Hortolândia
Av. Thereza Ana Cecon Breda, 1896 - Vila Sao Pedro, Hortolândia - SP, 13183-250

Abstract. *In today's environment, the integration of data from different sources and the need to ensure information security are critical challenges for organizations. In this context, this work focuses on the development of an ETL (Extraction, Transformation and Load) Tool capable of reading and processing spreadsheets in XLSX and CSV formats, with an emphasis on integrating and unifying data into a single file. Furthermore, the tool offers data encryption functionality, ensuring the confidentiality and security of information. The ETL process is fundamental to the quality and reliability of data used in various areas, from business analysis to academic research.*

Resumo. *Nos ambientes atuais, a integração de dados provenientes de diferentes fontes e a necessidade de garantir a segurança das informações são desafios críticos para as organizações. Nesse contexto, este trabalho se concentra no desenvolvimento de uma Ferramenta de ETL (Extração, Transformação e Carga) capaz de ler e processar planilhas em formatos XLSX e CSV, com ênfase na integração e unificação de dados em um único arquivo. Além disso, a ferramenta oferece a funcionalidade de criptografia dos dados, garantindo a confidencialidade e a segurança das informações. O processo de ETL é fundamental para a qualidade e confiabilidade dos dados utilizados em diversas áreas, desde análises de negócios até pesquisas acadêmicas.*

1. Introdução

A crescente produção e uso de dados nas organizações [QI.Edu 2023], tanto privada quanto pública, aumentou a importância da qualidade dos dados. No entanto, a qualidade dos dados nem sempre é garantida, pois eles podem ser corrompidos devido a vários motivos, como falha de hardware, problemas de software, falta de padronização etc. Segundo artigo publicado por [Costa 2009], dados corrompidos podem levar a análises errôneas, levando a tomadas de decisão ineficazes. Além disso, a falta de padronização na estrutura e formato dos dados pode dificultar sua análise e utilização. Segundo [Kim and Cho 2016], sistemas diferentes podem gerar dados em formatos diferentes, o que dificulta seu uso em conjunto, mesmo que se refiram à mesma entidade ou objeto.

Seguindo com a pesquisa feita por [Wan et al. 2016], a corrupção de dados em um banco de dados pode levar a problemas graves, como perda de informações cruciais e violação de privacidade. Nesse contexto, muitas organizações consideram fundamental a identificação e correção de dados corrompidos. Assim, as organizações se preocupam com questões como estruturas padronizadas inexistentes, dados corrompidos ou de qualidade inadequada, pois isso dificulta o uso dos dados para fins estratégicos e analíticos.

O projeto tem como principal atividade, desenvolver o Planilha Limpa com Criptografia e que tenha sua principal função ler os dados das planilhas nos formatos XLSX e

CSV, interpretar quais colunas já existem e agregar, criar colunas novas, verificar inconsistência de dados como tipos de dados incoerentes e criptografar o arquivo final gerado pelo sistema desenvolvido. Espera-se que a ferramenta proposta contribua para melhorar a qualidade dos dados utilizados em projetos de análise de dados e, consequentemente, para a tomada de decisões pelas organizações.

2. Objetivos

Este artigo tem como proposta de uma ferramenta que faça leitura de planilhas de excel em CSV e XSLX, faça uma limpeza de linhas vazias, concatene colunas com o mesmo nome junto com a possibilidade de criptografia e descriptografia da mesma.

Objetivo desta pesquisa é desenvolver uma ferramenta ETL (Extração, Transformação e Carregamento) baseada em uma revisão da literatura sobre as melhores práticas e métodos para garantir a qualidade dos dados em projetos de ETL.

Foi estabelecido como um dos objetivos a necessidade do usuário em relação à qualidade de dados, unificação e segurança. Ficou estabelecido que as atividades seriam; realizar pesquisas para identificar as demandas no contexto de qualidade de dados e definir requisitos específicos para a ferramenta, incluindo formatos de entrada/saída desejados e requisitos de segurança.

Selecionar as tecnologias mais adequadas para atender aos requisitos da ferramenta e as atividades, avaliar diferentes linguagens de programação, frameworks e bibliotecas. A escolha de tecnologias que suportem eficientemente a manipulação de planilhas, operações de dados e criptografia.

Para o desenvolvimento do *Front-End* tem como objetivo implementar uma interface do usuário intuitiva e funcional. As atividades necessárias são, criar páginas web interativas para upload, processamento e download de dados e integrar HTML, CSS e JavaScript para uma experiência de usuário otimizada.

No desenvolvimento do *Back-End* o objetivo é implementar a lógica de processamento de dados, limpeza e criptografia. E as atividades definidas são, desenvolver rotas e controladores para lidar com solicitações de upload, processamento e download e utilizar Python e o *framework* Flask para manipulação eficiente de dados e criptografia.

Para realizar a implementação de funcionalidades de limpeza de dados, o objetivo é desenvolver funcionalidades para remoção de duplicatas e linhas vazias. Contendo as atividades para criar classes ou módulos específicos para realizar operações de limpeza nos dados e integrar essas funcionalidades ao fluxo principal de processamento.

Outro objetivo é realizar a criptografia para integrar junto aos dados processados, se necessário conter as atividades, utilizar o *framework* Fernet para criptografar e descriptografar dados, também gerar chaves de criptografia e integrá-las ao processo de manipulação de dados.

E como último objetivo é criar um mecanismo eficiente para unificar dados de diferentes fontes tem as atividades propostas para desenvolver algoritmos ou métodos para combinar DataFrames de maneira eficiente além de garantir que a unificação preserve a consistência e integridade dos dados.

3. Referencial teórico

O processo de ETL é uma atividade crítica para organizações que desejam integrar e analisar dados de diversas fontes de forma eficiente. Nesse sentido, diversas abordagens e ferramentas têm sido desenvolvidas para auxiliar no processo de ETL.

Segundo [Kimball 2013], o processo de ETL é composto por três etapas: Extração, que envolve a coleta de dados de diferentes fontes; Transformação, que envolve a limpeza, normalização e enriquecimento dos dados coletados; Carga, que envolve a inserção dos dados transformados em um repositório de dados. Para realizar a extração de dados, é comum o uso de ferramentas de extração de dados, como o SQL Server Integration Services (SSIS) e o Pentaho Data Integration. Essas ferramentas possibilitam a extração de dados de diversas fontes, como bancos de dados, planilhas e arquivos de texto. No que se refere à etapa de transformação, diversas técnicas podem ser aplicadas, como a limpeza de dados, a normalização de dados, a agregação de dados e a filtragem de dados. Para realizar essas atividades, é comum o uso de ferramentas de transformação de dados, como o Talend Open Studio e o IBM InfoSphere DataStage.

Por fim, a etapa de carga envolve a inserção dos dados transformados em um repositório de dados, como um *data warehouse* ou um banco de dados relacional. Para realizar a carga de dados, é comum o uso de ferramentas de carga de dados, como o Microsoft SQL Server e o Oracle Database. Portanto, o processo de ETL é um tema relevante para pesquisas científicas na área de banco de dados e *business intelligence*, e é fundamental para a realização de análises precisas e eficientes de dados. As ferramentas e técnicas utilizadas nesse processo têm evoluído ao longo do tempo, e novas abordagens estão surgindo para tornar o processo de ETL mais eficiente e escalável. Como apresentado no livro [Kimball 2013], ele faz a análise de metodologias e análise de dados, na figura 1 contém no livro que explica um fluxo alternativo no processo proposto, onde existe uma nova etapa para analisar e enviar os dados utilizando ferramentas de *Bussiness Intelligence*, tendo como diferente do proposto para realizar o desenvolvimento do Planilha Limpa com Criptografia:

4. Trabalhos correlatos

No contexto da construção de conhecimento em uma aplicação de uma empresa seguradora, diversos estudos têm explorado o processo de ETL como uma etapa essencial para a integração e análise de dados. Esses trabalhos têm como objetivo principal extrair informações relevantes a partir de dados brutos, transformá-los em um formato adequado e carregá-los em um repositório central para posterior análise e tomada de decisões. Pesquisa realizada por [Costa 2018] enfoca a importância do processo de ETL em uma empresa seguradora. Esses estudos destacam que a aplicação de técnicas de ETL permite a consolidação de informações provenientes de diferentes fontes, como apólices de seguros, dados de sinistros e informações dos segurados. Através da extração, transformação e carga desses dados, é possível obter uma visão abrangente e atualizada sobre os riscos, segurados e sinistros, permitindo uma melhor compreensão do negócio e auxiliando na tomada de decisões estratégicas.

Outro tema relevante na área de ETL de dados é o conceito de ETL 2.0, que propõe uma extensão ao processo tradicional de extração, transformação e carga, voltada à integração de dados estruturados e não estruturados. Esse tema tem sido abordado em

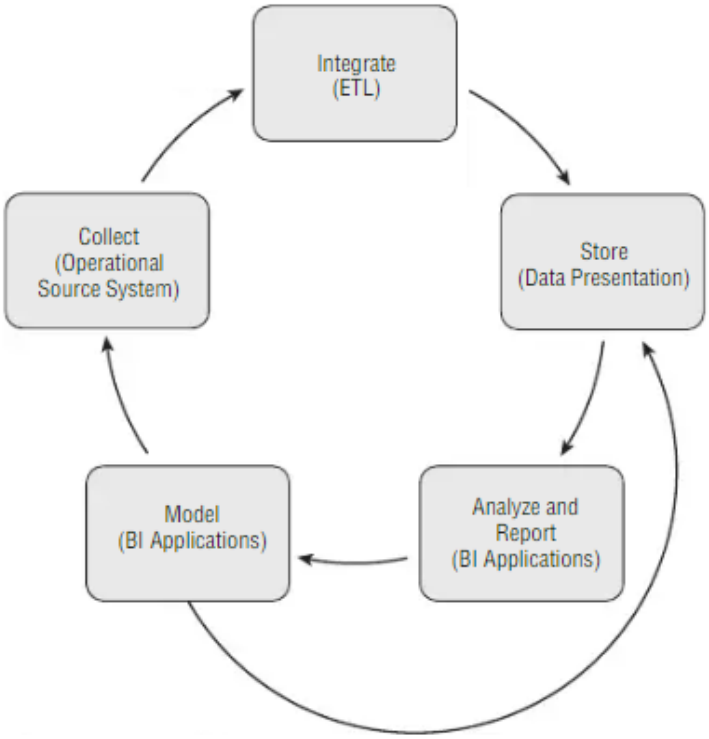

Figura 1. Loop fechado do CRM analítico Fonte: [Kimball 2013]

estudos recentes, como o trabalho de Souza et al. (2019). Nessa pesquisa, é proposta uma abordagem que combina técnicas de Processamento de Linguagem Natural (NLP) e aprendizado de máquina para realizar a extração e transformação de dados não estruturados, como documentos em formato de texto, imagens e áudios, e integrá-los a dados estruturados em um processo de ETL mais abrangente. Essa proposta de ETL 2.0 tem como objetivo superar os desafios relacionados à integração de dados estruturados e não estruturados, ampliando as possibilidades de análise e obtenção de *insights* a partir de informações presentes em diferentes formatos. Além disso, essa abordagem busca explorar o potencial dos avanços em tecnologias como Processamento de Linguagem Natural e aprendizado de máquina para automatizar e otimizar o processo de extração e transformação de dados não estruturados.

Esses trabalhos correlatos mostram a importância do processo de ETL na construção de conhecimento em uma aplicação de uma empresa seguradora, destacando os benefícios da integração e transformação de dados para a tomada de decisões estratégicas. Além disso, a proposta de ETL 2.0 demonstra uma perspectiva inovadora, buscando abordar os desafios de integração de dados estruturados e não estruturados, ampliando as possibilidades de análise e obtenção de informações relevantes para as organizações.

4.1. O processo de ETL na construção de conhecimento em uma aplicação de uma empresa seguradora

A monografia desenvolvida na Universidade Federal de Juiz de Fora no Curso de Bacharelado em Ciência da Computação realizou estratégias para auxiliar no processo de tomadas de decisão. Dentre essas estratégias, destacam-se o processo de KDD (*Knowledge Disco-*

very in Databases) e de DW (*Data Warehouse*). Para que o desenvolvimento do processo de KDD e do DW seja feito com sucesso é preciso realizar um tratamento nos dados das bases utilizadas. Este tratamento é conhecido como ETL (*Extraction, Transformation and Load*) e consiste em extrair os dados dos bancos de dados, realizar um processo de limpeza e transformação e, então, realizar a carga. Este é o foco principal deste trabalho, que além de apresentar os princiais conceitos de DW e KDD é feito um processo prático de ETL utilizando uma ferramenta própria para isto [Costa 2009].

4.2. ETL 2.0: Uma proposta de extensão ao processo de extração, transformação e carga voltada à integração de dados estruturados e não estruturados

Trabalho de Conclusão de Curso da Universidade Federal de Santa Catarina no Curso de Sistemas de Informação propõe uma extensão ao processo de ETL tradicional, integrando as duas visões (estruturada e não estruturada) em um modelo genérico para apoiar a tomada de decisão. Para atingir esses objetivos, foram desenvolvidas extensões a uma ferramenta de ETL de código aberto. Posteriormente, a ferramenta foi utilizada sobre dados reais para suportar o processo de ETL, produzindo como resultado um DW. Analisando-se os dados inseridos no DW a partir do processo completo, foi possível encontrar informações de correlação entre entidades pertencentes a ambas as classificações de dados. A principal contribuição do trabalho reside na extensão ao processo de ETL tradicional e na proposição, ainda que inicial, de um modelo de DW genérico para análise de relações entre entidades que promovem suporte a diversos cenários de tomada de decisão [Zorzo 2009].

5. Tecnologias

Neste projeto, uma variedade de tecnologias foram empregadas para desenvolver o Planilha Limpa com criptografia. Cada uma desempenha um papel fundamental na implementação e objetivo da solução proposta. Nesta seção, apresentamos um resumo das tecnologias utilizadas.

HTML (*Hypertext Markup Language*): É a espinha dorsal de qualquer página da web. Ele é responsável por definir a estrutura e o conteúdo do documento da web. O HTML foi usado para criar a estrutura da interface do usuário, incluindo a organização de elementos, formulários e a apresentação dos resultados da ferramenta de qualidade de dados.

CSS (*Cascading Style Sheets*): é essencial para estilizar a interface do usuário. Ele controla a formatação, o layout e a aparência visual dos elementos HTML. Durante o desenvolvimento da ferramenta, o CSS foi aplicado para garantir uma experiência de usuário atraente e amigável.

JavaScript: é uma linguagem de programação essencial para tornar a aplicação web interativa e dinâmica. Nesse contexto, o JavaScript foi usado para melhorar a usabilidade da ferramenta, validando formulários, interagindo com elementos da página e fornecendo *feedback* em tempo real.

Flask: é um *framework* web em Python que simplifica o desenvolvimento de aplicativos web. Ele foi usado para criar o servidor web e gerenciar as rotas da aplicação, recebe solicitações do *front-end* além de interagir com a lógica de negócios para fornecer os resultados aos usuários.

Pandas: é uma biblioteca usada para manipulação e análise de dados em Python. O Pandas desempenhou a funcionalidade na limpeza, transformação e análise de dados. Ele permitiu que a ferramenta de qualidade de dados identificasse erros, realizasse operações complexas e preparasse os dados para visualização ou armazenamento.

Fernet: é um *framework* de criptografia simétrica em Python que oferece segurança e simplicidade na criptografia e descriptografia de dados. Ele faz parte da biblioteca cryptography e é utilizado para proteger informações confidenciais. O Fernet utiliza uma chave compartilhada para criptografar e descriptografar dados, tornando-o eficiente e adequado para várias aplicações, incluindo segurança de dados em trânsito e armazenamento. Sua simplicidade de uso e robustez o tornam uma escolha popular para desenvolvedores que precisam de uma solução de criptografia segura em seus aplicativos Python.

O Visual Studio Code é uma IDE (*Integrated Development Environment*) que oferece uma experiência de desenvolvimento eficiente e personalizável. Foi a principal ferramenta de desenvolvimento para escrever, depurar e gerenciar o código deste projeto.

O Github é uma plataforma de controle de versão que permite o gerenciamento de código-fonte, colaboração e rastreamento de alterações. Foi usado para armazenar, versionar e compartilhar o código do projeto, tornando-o acessível a outros desenvolvedores e facilitando a colaboração.

O Canvas é uma ferramenta de prototipagem que auxiliou no design e planejamento da interface do usuário da ferramenta. Ele permitiu criar esboços visuais, modelos de página e fluxos de interação antes da implementação, economizando tempo e recursos durante o desenvolvimento.

6. Metodologia

A metodologia Aspiral (*Adaptive Spiral for Information Systems Development*) é uma abordagem iterativa e incremental para o desenvolvimento de sistemas de informação. Essa metodologia foi proposta por [Westfechtel and Conradi 1998] e foi escolhida para aplicar no desenvolvimento deste projeto devido ao seu formate de iterações e testes.

A ferramenta proposta pode ser uma maneira a integração e transformação de dados de diferentes fontes em um ambiente centralizado de processamento de dados. O desenvolvimento de uma ferramenta de ETL envolve várias etapas, desde o planejamento e design até a implementação e testes. Além disso, é necessário considerar aspectos de segurança, como a proteção dos dados durante a extração, transformação e carga.

A abordagem Aspiral é baseada em um ciclo de desenvolvimento que se repete várias vezes, com cada ciclo consistindo em uma série de etapas de análise, *design*, implementação e teste. A cada ciclo, o sistema é aprimorado com base no *feedback* recebido dos usuários e em novos objetivos identificados. Essa abordagem permite a adaptação do processo de desenvolvimento às mudanças nas necessidades dos usuários e nos requisitos do sistema. A metodologia de desenvolvimento em ASPIRAL também pode ser combinada com a metodologia Agile, que se baseia em sprints (iterações curtas de desenvolvimento) e na colaboração intensiva entre as equipes de desenvolvimento e os usuários do sistema. A combinação dessas metodologias pode ser benéfica, pois a abordagem Agile é focada em fornecer software funcional rapidamente, enquanto a abordagem ASPIRAL é focada em fornecer um sistema completo e de alta qualidade. Diversas empresas têm adotado a

abordagem Aspiral em seus projetos de desenvolvimento de sistemas, como a IBM, de acordo com o estudo da Regina Lopez, publicado no *website* Awari [Lopez 2023]. Além disso, a abordagem foi utilizada com sucesso em diversos projetos de pesquisa, como o desenvolvimento de sistemas de informação para gestão de recursos hídricos conforme citado por [Vijay et al. 2012] e [Prakash et al. 2015] para gestão de projetos de TI.

Portanto, a abordagem Aspiral é uma metodologia iterativa e incremental para o desenvolvimento de sistemas de informação que permite a adaptação do processo de desenvolvimento às mudanças nas necessidades dos usuários e nos requisitos do sistema. A combinação da abordagem Aspiral com a metodologia Agile, por meio da utilização de sprints, pode ser benéfica para projetos de desenvolvimento de sistemas que buscam entregar valor de forma rápida e eficiente.

7. Desenvolvimento do projeto

Neste projeto de pesquisa, propõe-se desenvolver uma ferramenta para utilizar durante o ETL de dados utilizando a abordagem Aspiral e as tecnologias Python, JavaScript, HTML e CSS. A ferramenta foi projetada para permitir a extração, transformação e carregamento de dados de diferentes fontes, a fim de integrá-los e prepará-los para análises posteriores.

No inicio do projeto foram convidados 4 usuários que trabalham com extração de dados para a produção de *dashboards* e fazem as extração de vários sistemas diferentes (que geram planilhas de formatos diferentes) para entender melhor as suas dificuldades atuais ao efetuar a atividade de unir 2 ou mais planilhas de diferente sitemas.

Foi realizado o levantamento dos requisitos, priorização e a análise e divisão das atividades necessárias para ter um melhor aproveitamento das as atividades foram separadas por grau de dificuldade, tecnologia e necessidade da funcionalidade. O processo de desenvolvimento foi iterativo, com ciclos de análise, implementação e testes. Esses ciclos permitem que a ferramenta seja adaptada às necessidades dos usuários e aos requisitos do projeto. A Figura 2 apresenta visualmente o fluxo de processamento da ferramenta, atuando na extração e transformação dos dados, na parte de carregar os dados é após o uso da ferramenta.

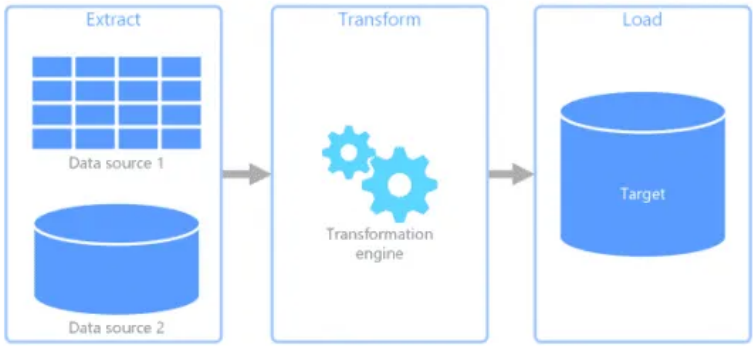

Figura 2. Processamento de dados ETL fonte: [InetSoft 2023]

Extração de Dados: a ferramenta é capaz de ler e extrair dados de planilhas XLSX e CSV, permitindo a coleta de informações a partir dessas fontes.

Transformação de Dados: após a extração, a ferramenta interpreta as colunas existentes, possibilitando a criação de novas colunas quando necessário. Além disso, verifica e trata inconsistências de tipos de dados, garantindo a integridade das informações.

Criptografia de Dados: a ferramenta oferece a opção de criptografar o arquivo final gerado. Isso contribui para a segurança dos dados, protegendo informações sensíveis contra acesso não autorizado.

No inicio do desenvolvimento, foi utilizado HTML, CSS e JavaScript no *front-end* para que o usuário pudesse inserir as planilhas desejadas, escolher o formato de saída da tabela e se deseja criptografar a tabela na saída. Com a tela pronta, cumprindo os objetivos necessários, foram realizados os testes em diferentes navegadores, e verificados que não é possível o envio de arquivos com o formato diferente de XLSX e CSV, e apresentar as opções ao usuário de selecionar o formato de saída do arquivo, a possibilidade de criptografar e selecionar o nome do arquivo. Após fim da primeira *sprint*, foi iniciado o desenvolvimento para o processamento de dados. O resultado da tela inicial está apresentado na Figura 3.

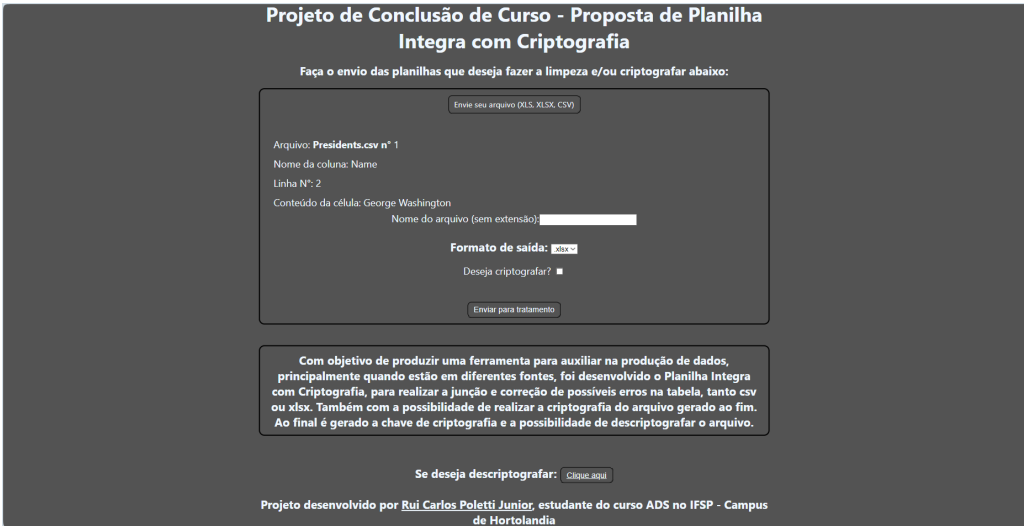

Figura 3. Tela Inicial do Sistema

Na segunda *sprint* do desenvolvimento foi utilizado o Pyhton para lidar com o envio e processamento de arquivos. A classe é usada para processar arquivos com base em sua extensão de arquivo (CSV e XLSX) e realizar operações de limpeza e edição nos dados. O código também combina os *DataFrames* resultantes em um único *DataFrame*, que pode ser salvo como um arquivo CSV.

Módulos e Bibliotecas: o código faz uso de módulos e bibliotecas, como Flask (a partir de import Flask as flask) e Pandas (import pandas as pd), comuns em aplicações web para o desenvolvimento de aplicativos e a manipulação de dados, respectivamente.

O código define uma rota *upload* que aceita solicitações POST. Essa rota é acionada quando o usuário envia um formulário contendo os arquivos. No contexto de uma aplicação web, uma rota é uma URL que corresponde a uma função.

A função *upload* lida com o envio de arquivos. Ela aceita arquivos enviados pelo usuário e os processa com base em sua extensão. A variável *uploaded-files* contém os

arquivos enviados pelo usuário. O código verifica se o número de arquivos não excede um limite máximo definido em anteriormente na variável global MAX-FILES.

O código combina os DataFrames resultantes em um único DataFrame usando o método pd.concat(). Isso é feito para que todos os dados processados de diferentes arquivos estejam em um só lugar.

Redirecionamento do Usuário: se o número de arquivos carregados estiver dentro do limite, o usuário é redirecionado para uma página chamada download.html. Caso contrário, é retornada uma mensagem informando que o número de arquivos excede o limite.

O código da Figura 4, faz parte de um sistema que permite aos usuários fazer upload de arquivos, editá-los e combiná-los em um único *DataFrame*, que pode ser posteriormente usado ou exportado. O código é usado para processamento de dados de arquivos CSV e XLSX.

```python
# metodo chamado após envio do formulário
@app.route('/upload', methods=['POST'])
def upload():
    """
    Esta função lida com uploads e processamento de arquivos. Ele aceita solicitações POST com arquivos anexados e os processa
    com base em sua extensão de arquivo. Os arquivos CSV e XLSX  são lidos em um Pandas DataFrame, edição de dados e são salvos como um novo
    Arquivo sendo XLSX ou CSV. Os DataFrames resultantes são combinados em um único DataFrame, que é salvo como um arquivo CSV. Finalmente, o usuário é
    redirecionado para uma página de download.

    Retorna:
        Se o número de arquivos enviados estiver dentro do limite permitido, a função retornará um download.html renderizado
        modelo. Caso contrário, retorna uma string indicando que o número de arquivos excede o limite.
    """
    uploaded_files = request.files.getlist('arquivo')  # variavel recebe do metodo post com id/name arquivo em uma lista.
    # instancia um array de arquivos processados para tratar depois no combined.xlsx ou .csv
    arquivos_processados = []
    # recebe parametro do metodo post o formato da saída escolhido pelo usuário
    formato_saida = request.form.get('formatoSaida')
    # recebe parametro para realizar ou não a criptografia
    criptografia = request.form.get('CriptoSim')

    # verifica se a quantidade de arquivos é menor ou igual à quantidade de arquivos definido logo acima
    if len(uploaded_files) <= MAX_FILES:
        combinado_df = pd.DataFrame()  # Crie um DataFrame vazio para combinar as planilhas
        FileXlsx = None  # criamos uma váriavel nula para utilizar dentro da condição se a plalinha for XLSX
        FileCsv = None  # criamos uma váriavel nula para utilizar dentro da condição se a plalinha for CSV
        for uploaded_file in uploaded_files:  # enquanto tiver arquivos feitos uploads ele executa as condições abaixo
            if uploaded_file.filename != '':  # se o arquivo não for vazio ele vazio
                # ele pega todo conteúdo depois do ponto, o esperado é que seja XLSX ou CSV
                extensao_arquivo = uploaded_file.filename.split('.')[-1]
                # verifica se a extensão estiver previamente declarada no array ALLOWED_EXTENSIONS
                if extensao_arquivo in ALLOWED_EXTENSIONS:
                    if extensao_arquivo in ['csv']:
                        # metodo para varivavel FileCSV receber pandas lendo o CSV e como parametro o arquivo feito upload
                        FileCsv = pd.read_csv(uploaded_file)

                        # Tratamento de dados aqui (Classes de Limpeza, Identificação de erros e concatenação de Colunas)
                        cleaner = DataCleaner(FileCsv)
                        cleaner.limpeza_dado()

                        FileCsv = cleaner.get_dados_clean()
```

Figura 4. Função de Upload em Pyhton

Na terceira *Sprint* foi desenvolvido a função para criptografar o arquivo gerado pelo usuário após as alterações necessárias feitas pelo sistema. Após escolher o formato desejado, ao retornar o arquivo no formato selecionado, se for selecionado para criptografar, ele entrega junto ao arquivo a chave de criptografia gerada.

A entrega da quarta *Sprint* o foco do desenvolvimento foi a função de descriptografar o arquivo, no caso do Planilha Limpa com Criptografia, Nesse contexto, o *framework* Fernet desempenhou um papel fundamental na segurança e integridade dos dados. O Fernet é uma escolha sólida para a descriptografia, pois é uma implementação moderna e eficiente da criptografia simétrica. Com ele, os dados criptografados são protegidos por uma chave compartilhada, garantindo a confidencialidade das informações.

A função de descriptografia desenvolvida nessa etapa permite que os usuários do Planilha Limpa com Criptografia, restaurem os arquivos anteriormente criptografados,

utilizando a chave de descriptografia correta. Isso garante que os dados críticos sejam acessíveis apenas para aqueles com as credenciais apropriadas. Além disso, o uso do Fernet simplifica a implementação, já que ele lida com muitos aspectos complexos da criptografia, como geração de chaves e proteção de dados, de maneira transparente para os desenvolvedores.

Este código representa uma função Python definida em um aplicativo Flask que lida com a descriptografia de um arquivo criptografado com base nas entradas fornecidas pelo usuário em um formulário da web. Vou explicar as principais partes do código:

Rota e Método HTTP: a função é associada à rota "/descriptografar"e só responde a solicitações HTTP do tipo POST. Isso significa que a função será chamada quando um usuário enviar um formulário da web.

Docstring: a docstring (comentário dentro de três aspas triplas) fornece uma descrição detalhada do que a função faz e como ela funciona. Isso ajuda os desenvolvedores a entenderem a função e também pode ser usado para gerar documentação automática.

Obtenção dos Parâmetros a partir da solicitação do usuário: a chave de criptografia fornecida pelo usuário no formulário apresentado na página HTML e o formato de saída desejado (por exemplo, "xlsx"ou "csv").

Obtenção do Arquivo Criptografado: a função é chamada para obter o arquivo criptografado enviado pelo usuário. Isso pressupõe que o formulário da web inclui um campo de arquivo com o nome "arquivoCriptografado".

Verificação do Formato de Saída: a função verifica se o formato de saída desejado é "xlsx"ou "csv". Dependendo do formato escolhido, o código apropriado é executado.

Descriptografia do Arquivo: se um arquivo criptografado foi enviado e o formato de saída é válido, a função instancia um objeto DescriptografarArquivo e chama o método decrypt-file() para descriptografar o arquivo.

Mensagens de Notificação: depois de descriptografar com sucesso o arquivo, a função define uma mensagem de notificação indicando que o arquivo foi descriptografado com sucesso. Caso nenhum arquivo tenha sido enviado, a função fornece uma mensagem informando que nenhum arquivo foi enviado para descriptografar. Se ocorrer algum erro, uma mensagem de erro é definida.

Renderização de Templates HTML: por fim, a função renderiza templates HTML para exibir as mensagens de notificação. Dependendo do resultado da descriptografia, o usuário verá uma mensagem de sucesso ou uma mensagem de erro em uma página HTML.

No geral, essa função faz parte de um aplicativo web que permite aos usuários enviar arquivos criptografados, escolher o formato de saída desejado e, em seguida, descriptografar o arquivo e receber notificações sobre o processo. É uma parte importante de uma ferramenta mais ampla para lidar com dados criptografados e contribui para tornar o processo de descriptografia mais acessível aos usuários finais. A Figura 5 apresenta o código gerado para realizar a descriptografia quando solicitado.

Durante o desenvolvimento da ferramenta de ETL, foi usado uma abordagem modular. A extração de dados pode ser implementada utilizando a biblioteca *requests* para

```python
@app.route('/descriptografar', methods=['POST'])
def descriptografar():
    """
    Descriptografa um arquivo criptografado com base na entrada do usuário.

    A função recebe uma solicitação POST com a entrada do usuário, incluindo a chave de criptografia e o formato de saída desejado.
    Se o formato de saída for xlsx ou csv, a função verifica se o arquivo criptografado existe e o descriptografa usando a chave fornecida.
    Se a descriptografia for bem-sucedida, a função retornará uma mensagem de sucesso ao usuário.
    Caso o arquivo criptografado não seja encontrado, a função retorna uma mensagem de erro ao usuário.

    Retorna:
        Um modelo HTML renderizado com uma mensagem de sucesso ou de erro, dependendo do resultado do processo de descriptografia.
    """
    chave = request.form.get('chave') #varivavel chave recebe parametro do HTML chave
    formato_saida = request.form.get('formatoSaida') #varivavel formato_saida recebe parametro do HTML formatoSaida

    arquivo_criptografado = request.files.get('arquivoCriptografado')

    if formato_saida == "xlsx":
        if arquivo_criptografado and arquivo_criptografado.filename:  # Verifica se um arquivo foi enviado

            local_arquivo_caminho = 'fileupload/uploads/descriptografado_' +  arquivo_criptografado.filename
            chave = request.form.get('chave')
            descriptografador = DescriptografarArquivo(chave, arquivo_criptografado, local_arquivo_caminho)
            descriptografador.decrypt_file()

            notificacao = "Arquivo descriptografado com sucesso!"
            return render_template('info.html', notificacao=notificacao)
        else:
            notificacao = "Nenhum arquivo enviado para descriptografar."
            return render_template('info.html', notificacao=notificacao)
```

Figura 5. Função para descriptografar o arquivo enviado

fazer solicitações via HTTP. Em seguida, foi aplicado a transformação de dados usando a biblioteca Pandas para filtrar, limpar e reformatar os dados conforme as necessidades do projeto. O uso de um servidor é necessário devido aos protocolos de segurança, o uso de uma linguagem *client-side*para gravar arquivos no disco rígido do usuário, apenas recebemos a última pasta, por questões de segurança. Por esse motivo foi necessário a instalação do servidor local para realizar os testes e então foi utilizado a biblioteca Flask para ser o servidor até final do desenvolvimento.

Ao fim do projeto, a ferramenta desenvolvida Planilha Limpa com Criptografia, pode ser utilizada em diferentes contextos e projetos. Além disso, o uso da metodologia Aspiral permite uma abordagem adaptativa e iterativa no desenvolvimento da ferramenta, tornando-a mais adequada às necessidades dos usuários e do projeto.

8. Resultados

O desenvolvimento e conclusão do projeto de Trabalho de Conclusão de Curso (TCC), que envolve a criação do Planilha Limpa com Criptografia, conseguiu atingir os objetivos esperados e realizar a leitura e concatenação das planilhas inseridas e criptografar se selecionado a opção. O projeto tinha como objetivo principal desenvolver uma solução que permitisse a leitura e unificação de dados a partir de planilhas nos formatos XLSX e CSV, além de fornecer a opção de criptografar o arquivo resultante. Ao término do projeto, os resultados alcançados foram:

Unificação de Dados: a ferramenta projetada provou ser capaz de ler e combinar eficazmente dados a partir de várias fontes, permitindo que diferentes conjuntos de informações fossem consolidados em um único arquivo. Isso demonstrou ser uma solução eficiente para simplificar o gerenciamento de dados dispersos e facilitar análises posteriores.

Tratamento de Dados: a ferramenta incluiu funcionalidades para a limpeza e transformação de dados. Durante o processo de unificação, foram aplicadas técnicas de limpeza para remover duplicatas e linhas vazias, resultando em dados mais consistentes e

confiáveis. Isso se mostrou valioso para a melhoria da qualidade dos dados.

Segurança de Dados: a opção de criptografar o arquivo final gerado pela ferramenta foi implementada com sucesso. O uso do *framework* Fernet permitiu a proteção dos dados, tornando-os seguros para transferência e armazenamento. Os usuários têm a capacidade de criptografar seus dados sensíveis, o que é essencial para a privacidade e conformidade com regulamentações.

Personalização de Resultados: a ferramenta permitiu que os usuários personalizassem o nome do arquivo final, facilitando a identificação do arquivo gerado.

Controle de Versão e Colaboração: o uso do Github https://github.com/ruipoletti28/Data-Quality) como plataforma de controle de versão facilitou o rastreamento de alterações e compartilhamento de código. Isso resultou em um processo de desenvolvimento mais eficiente por conta da facilidade em controlar as versões no código fonte, quando se é usado metodologia de iterações.

Em resumo, o projeto desenvolvido proporcionou uma ferramenta que aborda uma necessidade essencial no cenário de gerenciamento de dados. A capacidade de unificar, limpar, criptografar e descriptografar informações provenientes de várias fontes representa um avanço notável na qualidade de dados. Espera-se que essa ferramenta seja útil em contextos nos quais a qualidade e a segurança dos dados são prioridades, contribuindo para a tomada de decisões informadas e eficientes nas organizações.

9. Conclusão

O desenvolvimento de uma ferramenta eficiente de ETL de dados é essencial para empresas e organizações que desejam integrar informações provenientes de diversas fontes em um banco de dados centralizado. Para alcançar esse objetivo, a combinação de linguagens de programação e tecnologias apropriadas desempenha um papel crucial. Este estudo demonstrou que a integração das linguagens Python e JavaScript proporciona uma base sólida para a criação de uma ferramenta de ETL altamente funcional. A pesquisa destacou as capacidades e vantagens de cada linguagem ao longo do desenvolvimento do sistema, ressaltando o potencial dessa abordagem para aprimorar os processos de integração, transformação e carga de dados.

Eficiência do Python na manipulação de dados: O Python emergiu como uma linguagem de programação poderosa e amplamente utilizada para lidar com tarefas de manipulação de dados. Durante a pesquisa, foi observado que o Python oferece uma variedade de bibliotecas e recursos que facilitam a extração, transformação e análise de dados. A biblioteca Pandas é um exemplo notável, permitindo a leitura de diferentes formatos de arquivo, a manipulação de *DataFrames* de dados e a aplicação de transformações complexas aos dados. Isso é fundamental para garantir a qualidade e a consistência dos dados durante o processo de ETL.

JavaScript para uma interface de usuário dinâmica: Além das funcionalidades de processamento de dados, o JavaScript desempenha um papel fundamental na criação de uma interface de usuário interativa e dinâmica. A pesquisa enfatizou a capacidade do JavaScript de fornecer uma experiência de usuário amigável, com interfaces intuitivas que simplificam a interação do usuário com as funcionalidades da ferramenta de ETL. Essa combinação de Python para o processamento de dados nos bastidores e JavaScript

para a interface de usuário oferece uma solução completa.

Limitações e desafios futuros: É fundamental reconhecer que, embora a combinação de Python e JavaScript tenha se mostrado eficaz, existem limitações a serem consideradas. A escalabilidade do sistema pode ser um desafio, principalmente ao lidar com grandes volumes de dados. Portanto, futuras pesquisas e desenvolvimentos podem se concentrar em estratégias de otimização e escalabilidade para garantir o desempenho adequado em cenários de alta demanda.

Além disso, questões de segurança, como a proteção dos dados durante o processo de ETL, devem ser abordadas de forma adequada. A segurança dos dados é uma preocupação crítica [Travasso 2023], e o desenvolvimento de medidas de proteção robustas deve ser incorporado à ferramenta.

Portanto, a combinação de Python e JavaScript oferece um pontencial para o desenvolvimento de ferramentas de ETL, proporcionando um impacto positivo nas operações de análise de dados e no gerenciamento de informações.

10. Trabalhos Futuros

Como trabalhos futuros, inicialmente seria uma etapa de desenvolvimento de classes em Python, projetadas para analisar e compreender os campos com dados incompletos. Pois a identificação e o tratamento de dados incompletos são tarefas na garantia da qualidade dos mesmos. As classes que serão desenvolvidas podem fazer parte do fluxo na análise de dados e fornecem *insights* valiosos sobre a integridade e a consistência dos conjuntos gerados.

Além disso, a expansão do trabalho abrange a criação de funções que calculam a quantidade de linhas agregadas, excluídas e modificadas. Essas funções não apenas facilitam a análise estatística dos dados, mas também contribuem para o processo de limpeza e transformação de dados. A capacidade de quantificar as mudanças é um aspecto essencial no monitoramento da qualidade dos dados e no acompanhamento de melhorias ao longo do tempo.

O desenvolvimento utilizando o Python direcionado para aumentar a capacidade de processamento se torna um objetivo no futuro para qualquer organização que lide com grandes volumes de dados. Além disso, a melhoria do *front-end*, para uma melhor usabilidade e novas funcionalidades visando as necessidades do usuário final contribuindo para a adoção da ferramenta.

Futuramente, está prevista a implementação de novas telas no *front-end*, com o objetivo de aprimorar a visualização das tabelas, apresentando graficamente pontos de atenção. Esses pontos de atenção podem ser definidos como linhas vazias ou dados incompletos. Atualmente, o Planilha Limpa com Criptografia interpreta e mostra a linha pré-definidas no sistema; no entanto, a pretensão é permitir que essa personalização seja feita pelo usuário para que visualizem os dados da célula na localização da coluna e linha definida.

Referências

Costa, M. A. S. D. (2009). O processo de etl na construção de conhecimento em uma aplicação de uma empresa seguradora.

Costa, M. A. S. D. (2018). O processo de etl na construção de conhecimento em uma aplicação de uma empresa seguradora e o tema.

InetSoft (2023). *The Definition of ETL and Its Advantages and Disadvantages*.

Kim, H.and Kwon, Y. and Cho, S. (2016). *Data quality management, data usage experience and acquisition of data users*.

Kimball, M. R. (2013). *The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling*. John Wiley Sons, Inc.

Lopez, R. (2023). Metodologia Ágil em espiral: Abordagem iterativa e controlada para projetos. *https://awari.com.br/metodologia-agil-em-espiral-abordagem-iterativa-e-controlada-para-projetos/*.

Prakash, S., Vijay, R., and Subramanian, K. (2015). Aspiral-based decision support system for water resource management. pages 1–7.

QI.Edu, B. (2023). Big data: Entendendo seu impacto através de 10 estudos de caso notáveis. *https://qi.edu.br/big-data-entendendo-seu-impacto-atraves-de-10-estudos-de-caso-notaveis/*.

Travasso, B. (2023). Dados críticos, quais são? onde estão? *revistasegurancaeletronica*.

Vijay, R., Prakash, S., and Subramanian, K. (2012). Adaptive spiral project management methodology: An agile software development approach. pages 1–6.

Wan, J., Cai, H., Zhou, K., and Zhang, D. (2016). From machine-to-machine communications towards cyber-physical systems.

Westfechtel, B. and Conradi, R. (1998). System configuration management.

Zorzo, A. L. (2009). *ETL 2.0: Uma proposta de extensão ao processo de extração, transformação e carga voltada à integração de dados estruturados e não estruturados.*

Documento Digitalizado Público

Artigo Final de TCC Rui Poletti - Planilha Ingegra

Assunto: Artigo Final de TCC Rui Poletti - Planilha Ingegra
Assinado por: Edgar Noda
Tipo do Documento: Comprovante
Situação: Finalizado
Nível de Acesso: Público
Tipo do Conferência: Documento Digital

Documento assinado eletronicamente por:

- **Edgar Noda**, **PROFESSOR ENS BASICO TECN TECNOLOGICO**, em 14/03/2024 15:17:17.

Este documento foi armazenado no SUAP em 14/03/2024. Para comprovar sua integridade, faça a leitura do QRCode ao lado ou acesse https://suap.ifsp.edu.br/verificar-documento-externo/ e forneça os dados abaixo:

Código Verificador: 1612249
Código de Autenticação: 5147bab081