

**PONTIFÍCIA UNIVERSIDADE CATÓLICA DE GOIÁS
ESCOLA POLITÉCNICA E DE ARTES
GRADUAÇÃO EM ENGENHARIA DA COMPUTAÇÃO**



**DESENVOLVIMENTO DE UMA FERRAMENTA PARA CORREÇÃO DE
DADOS RECUSADOS DURANTE O CENSO ESCOLAR (INEP) UTILIZANDO
A LINGUAGEM PERL**

ALEXANDRE MASCARENHAS SANTOS

GOIÂNIA,
2023

ALEXANDRE MASCARENHAS SANTOS

**DESENVOLVIMENTO DE UMA FERRAMENTA PARA CORREÇÃO DE
DADOS RECUSADOS DURANTE O CENSO ESCOLAR (INEP) UTILIZANDO
A LINGUAGEM PERL**

Trabalho de Conclusão de Curso apresentado a escola Politécnica e de Artes da Pontifícia Universidade Católica de Goiás, como parte dos requisitos para a obtenção do título de Bacharel em Engenharia de Computação.

Orientador(a):

Prof. Rafael Leal Martins

Banca examinadora:

Prof. Me. Fernando Gonçalves Abadias

Prof. Me. Gustavo Siqueira Vinhal

GOIÂNIA
2023

Dedico este trabalho a mim e
às vozes na minha cabeça.

AGRADECIMENTOS

Agradeço primeiramente a Deus, criador de todas as coisas. Aos meus pais que são minha base e que sempre me apoiaram em todos meus projetos. Aos meus amigos Lucas Queiroz e Juliana Maris que sempre me apoiaram e me incentivaram a melhorar.

Ao meu orientador Rafael Leal Martins, pelas sábias orientações, pela paciência, pelo apoio e confiança no desenvolvimento deste trabalho.

"Acredite em si mesmo e tudo
será possível." - Paulo Coelho

RESUMO

O censo escolar consiste no levantamento estatístico realizado pelo INEP, abrangendo todas as etapas e modalidades de ensino da educação básica. O censo escolar é realizado anualmente e possui duas etapas de coleta. Seu objetivo principal é coletar informações detalhadas sobre a estrutura escolar, as turmas, os alunos, os docentes e os recursos pedagógicos. As informações coletadas são cruciais para o planejamento e o monitoramento das políticas públicas educacionais, uma vez que geram indicadores e estatísticas sobre a situação educacional do país. Em muitos momentos, o envio desses dados é algo moroso, pois os dados precisam ser enviados formatados de acordo com um layout e estar em conformidade com as regras estabelecidas pelo INEP. Este trabalho visa o desenvolvimento de uma ferramenta que auxilie na correção de erros de arquivos recusados por não estarem em conformidade com alguma regra ou com a base de dados do INEP.

Palavras-chaves: *Ferramenta, Censo Escolar, Perl.*

ABSTRACT

The school census is a statistical survey conducted by the National Institute for Educational Studies and Research (INEP), encompassing all stages and modalities of basic education. The school census is carried out annually and consists of two data collection stages. Its main objective is to gather detailed information about the school structure, classes, students, teachers, and educational resources. The collected information is crucial for planning and monitoring public education policies, as it generates indicators and statistics about the educational situation in the country. However, the data submission process can be time-consuming, as the data needs to be properly formatted according to a specific layout and comply with the rules established by INEP. This work aims to develop a tool that assists in correcting errors in files that are rejected due to non-compliance with certain rules or the INEP database.

Keywords: *Tool, School Census, Perl.*

LISTA DE TABELAS

Tabela 1 - Tipos de Variáveis.....	11
Tabela 2 - Incremento/Decremento de variável.....	13
Tabela 3 - Operadores Lógicos	14

LISTA DE FIGURAS

Figura 1 - Regras gerais.....	5
Figura 2 - Arquivo de texto final gerado	7
Figura 3 - Relatório de erros.....	8
Figura 4 - Repositório CPAN.....	10
Figura 5 - Diagrama em blocos representando a aplicação	23
Figura 6 - Interface do Visual Studio Code.....	26
Figura 7 - Interface do GitHub Desktop.....	27
Figura 8 - Interface do SQL Server Management Studio (SSMS)	28
Figura 9 - Tela Inicial.....	30
Figura 10 - Tela Matrícula Inicial	31
Figura 11 - Tela Situação do Aluno	32
Figura 12 - Etapas do Projeto.....	33
Figura 13 - Fluxo de Interpretação de Arquivos	34
Figura 14 - Fluxo do Banco de Dados.....	35
Figura 15 - Fluxo do Banco de Dados II.....	35
Figura 16 - Fluxo do Banco de Dados III.....	36
Figura 17 - Fluxo do Banco de Dados IV	37
Figura 18 - Fluxo da API.....	38
Figura 19 - Fluxo da Interface	39
Figura 20 - Tela F-repair C#.....	45

LISTA DE SIGLAS

INEP - Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira

MEC - Ministério da Educação

IDEB - Índice de Desenvolvimento da Educação Básica

SEDUC - Secretaria de Educação

PERL - *Practical Extraction and Reporting Language*

CPAN - *Comprehensive Perl Archive Network*

API - *Application Programming Interface*

REST - *Representational State Transfer*

HTTP - *Hypertext Transfer Protocol*

SQL - *Structured Query Language*

SSMS - *SQL Server Management Studio*

IDE - *Integrated Development Environment*

T-SQL - *Transact-SQL*

SGBDR - *Sistema Gerenciador de Banco de Dados Relacional*

CORS - *Cross-Origin Resource Sharing*

XML - *eXtensible Markup Language*

JSON - *JavaScript Object Notation*

SUMÁRIO

1.	INTRODUÇÃO.....	1
1.1.	Objetivos.....	2
1.1.1.	Objetivos Geral	2
1.1.2.	Objetivos Específicos.....	2
1.2.	Justificativa	2
1.3.	Metodologia	3
2.	REFERENCIAL TEÓRICO	4
2.1.	Perl	8
2.1.1.	História.....	8
2.1.2.	Como funciona o Perl.....	9
2.1.3.	Prós e Contras da Linguagem Perl	9
2.1.4.	CPAN - Módulos e <i>Frameworks</i>	9
2.1.5.	Variáveis e Tipos de Dados	10
2.1.6.	Operadores Aritméticos.....	12
2.1.7.	Operadores Relacionais e Lógicos	13
2.1.8.	Input de Dados.....	14
2.1.9.	Comando Use	15
2.1.10.	Utilização da Linguagem Perl	16
2.2.	API.....	18
2.2.1.	Para que serve API	19
2.2.2.	Por que usar API?.....	19
2.2.3.	API em Perl.....	19
2.2.4.	Rotas.....	20
2.2.5.	Endpoints	20
3.	DESCRIÇÃO GLOBAL	22
3.1.	Aspecto Geral do Produto.....	22
3.2.	Interface do Sistema	23
3.3.	Interface do Usuário.....	24
3.4.	Funções da Ferramenta.....	24
3.5.	Características do Usuário.....	24
3.6.	Suposições e Dependência	25
4.	DESENVOLVIMENTO DA FERRAMENTA.....	26
4.1.	Visual Studio Code	26

4.2.	GitHub.....	27
4.3.	SQL Server Management Studio (SSMS).....	27
5.	IMPLEMENTAÇÃO E RESULTADOS	29
5.1.	Telas da Ferramenta.....	29
5.1.1.	Tela Inicial	29
5.1.2.	Tela Matrícula Inicial	30
5.1.3.	Tela Situação do Aluno	31
5.2.	Fluxo	32
5.3.	Implementação de API.....	40
5.3.1.	Endpoint Busca de Escolas Disponíveis	41
5.3.2.	Endpoint de Pesquisa Escola Específica	42
5.4.	Implementação da Interface Gráfica	44
5.5.	Implementação alternativa da ferramenta utilizando a linguagem C# 45	
6.	CONSIDERAÇÕES FINAIS.....	46
6.1.	Recomendações para Trabalhos Futuros.....	46
	REFERÊNCIAS BIBLIOGRÁFICAS	48

1. INTRODUÇÃO

O censo é um processo de coleta, análise, interpretação, publicação e disseminação de dados estatísticos, geralmente em âmbito nacional, sobre a população, economia, educação, condições de vida e outros aspectos de uma sociedade. O censo escolar é um instrumento de coleta de dados que tem como objetivo estudar a população escolar do país, abrangendo aspectos como número de estudantes, faixa etária, níveis de ensino, qualidade da infraestrutura escolar, dentre outros. O conhecimento detalhado destas informações permite aos governos, entidades educacionais e sociedade civil planejar e implementar políticas públicas de educação de maneira mais eficaz e direcionada, contribuindo para o desenvolvimento do país ao garantir uma educação de qualidade para todos, formando cidadãos mais capacitados, críticos e prontos para atuar no mercado de trabalho e na sociedade de forma geral. (CAMARGO, 2022).

O Censo Escolar captura informações relativas à educação básica da rede municipal, estadual e privada de ensino em âmbito nacional. Esse levantamento abrange todos os níveis de ensino (Educação Infantil, Ensino Fundamental e Ensino Médio), bem como as modalidades de ensino (Ensino Regular, Educação Especial, Educação de Jovens e Adultos e Educação Profissional). O Censo Escolar é realizado anualmente e requer que as instituições de ensino forneçam os dados categorizados referentes aos alunos e à estrutura escolar (DINIZ, 1999).

Contudo, o fornecimento desses dados nem sempre é algo fácil de ser executado, pois os dados devem cumprir diversos requisitos, estar adequados a um layout atualizado e passar por diversas validações antes de serem aceitos pelo Instituto Nacional de Estudos e Pesquisas (INEP). No cenário em que um bloco de dados enviado não tenha sucesso, é possível que o arquivo de texto enviado possua erros de inconformidade de dados como CPF, nome, data de nascimento, cep entre outros, dado isso o mesmo deve ser revisado e corrigido. Muitas vezes, esse processo é feito de forma manual, o que resulta em um retrabalho dispendioso e demorado para a correção dos dados.

1.1. Objetivos

O projeto possui os seguintes objetivos:

1.1.1. Objetivos Geral

Desenvolver e documentar uma aplicação para auxiliar o processo de correção dos arquivos de texto gerados para envio ao censo escolar.

1.1.2. Objetivos Específicos

- Auxiliar o processo de correção dos arquivos;
- Proporcionar um ambiente onde a correção seja rápida, objetiva e interativa;
- Disponibilizar uma interface de fácil utilização;
- Implementar a ferramenta utilizando a linguagem de programação Perl;
- Estudar tecnologias *frameworks* para implementação da aplicação;
- Definir a versão do Perl que será utilizada
- Realizar testes na aplicação a ser desenvolvida “F-repair”

1.2. Justificativa

Segundo o MEC (2022), o Censo Escolar é uma ferramenta crucial que fornece informações detalhadas sobre a situação educacional em níveis nacional, estadual, municipal e das instituições de ensino. Esse levantamento permite a compreensão do panorama educacional, monitoramento de resultados e eficácia de políticas públicas.

O Censo Escolar fornece uma análise detalhada da situação da educação no país, sendo fundamental para identificar necessidades e direcionar melhorias na educação básica. Sua realização possibilita a identificação de problemas no sistema educacional, incentivando os responsáveis a tomar ações em benefício da educação. Logo, o fornecimento preciso dos dados para o censo é essencial para melhorar a qualidade da educação. Como objetivo este projeto apresenta uma aplicação que visa facilitar a correção dos dados solicitados para a migração das unidades escolares (MEC, 2022).

1.3. Metodologia

Este trabalho apresenta um levantamento bibliográfico e experimental, por meio de teses, artigos, livros, websites e outros meios que disponibilizam informações relevantes sobre o censo escolar. Quanto aos seus procedimentos técnicos, esta pesquisa visa o estudo da linguagem Perl e possíveis módulos e ferramentas que auxiliem no desenvolvimento de uma aplicação utilizando a linguagem Perl, a fim de facilitar a correção de arquivos de texto. O objetivo é aumentar a produtividade e reduzir o retrabalho em todo o processo de correção de arquivos de texto (WAZLAWICK, 2014).

Perl foi a linguagem escolhida devido à sua alta performance na manipulação de textos e às diversas funções de expressões regulares que serão úteis para o objetivo do trabalho.

2. REFERENCIAL TEÓRICO

O Censo Escolar é um processo realizado de forma descentralizada e conta com o apoio dos estados e municípios. Para que a coleta seja realizada com êxito, é necessário que cada ator envolvido no processo cumpra com suas responsabilidades. O INEP é o responsável por gerir prazos, definir o *layout* e disponibilizar aos demais atores todos os meios necessários para a realização da coleta. O Censo Escolar possui um fluxo de coleta dados, esse fluxo é dividido em duas etapas sendo elas, matrícula inicial e situação do aluno. O fornecimento dos dados para ambas as etapas é de caráter obrigatório.

A primeira etapa refere-se à matrícula inicial, iniciando-se na última quarta-feira de maio e estendendo-se até o mês de agosto. Os dados solicitados nessa etapa abrangem informações sobre a escola como unidade de ensino, incluindo a estrutura física do prédio, os equipamentos existentes, os dados sobre o quadro de pessoal técnico, administrativo e docente, os níveis de atuação e o grau de formação. Também é necessário informar o número de salas disponíveis na unidade, os níveis e tipos de modalidades de ensino oferecidas, além de informações relativas aos alunos, como nome, sexo, raça, turno, turma, série e período.

A segunda etapa ocorre no início do ano seguinte, após a conclusão do ano letivo. Nessa etapa, o objetivo é coletar informações sobre a situação e movimentação dos alunos após o término do ano letivo, contemplando opções como transferência, desistência, falecimento e mudança de cidade. Além disso, são solicitados dados relativos ao desempenho dos alunos ao longo do ano letivo, como aprovação e reprovação, que serão fundamentais para o cálculo do Índice de Desenvolvimento da Educação Básica (IDEB) (INEP, 2022).

As instituições de ensino possuem a responsabilidade de transmitir os dados da instituição (estrutura escolar, alunos, docentes etc.) e garantir a veracidade das informações repassadas nos prazos estabelecidos.

Há duas formas de enviar os dados escolares para o Censo. A primeira forma é através do acesso direto ao *website* do Educacenso, onde é possível preencher manualmente as informações requeridas no próprio site. A segunda forma é por meio do *upload* dos dados por um arquivo de texto.

Nesse contexto, se faz necessária a contratação de um *software* particular de gestão escolar adequado ao *layout* anual disponibilizado pelo INEP, por parte da unidade de ensino. Esse serviço torna a migração de dados para o site do Educacenso mais otimizada e eficiente, porém não é acessível a todas as unidades de ensino.

O *layout* disponibilizado pelo Instituto Nacional de Estudo e Pesquisa (INEP) possui diversas regras, conforme exibido na Figura 1. O *layout* possui regras gerais que definem a quantidade de campos que cada tipo de registro deve conter, assim como as mensagens de erro para cada uma das regras gerais caso elas sejam violadas. O *layout* pode sofrer diversas mudanças de versão durante o processo inicial de coleta então juntamente com as regras é declarado uma legenda que indica o que foi incluído, alterado ou excluído, realçado por cores.

Para gerar um arquivo de texto com alta possibilidade de aceitação, é necessário que o sistema de gestão esteja em conformidade com cada uma das regras (INEP, 2022).

Figura 1 - Regras gerais

	A	B	C	D	E
1	Layout de Importação e Exportação do Educacenso 2022 (Matrícula Inicial) - Versão 5				
2					
3	regra	mensagem de erro	momento da verificação	status da escola	Legenda:
4	1. Deve ser utilizado o padrão ISO-8859-1 de codificação de caracteres.	Para a geração do arquivo deve ser utilizado o padrão ISO-8859-1 de codificação de caracteres.	validação	Recusada	Inclusão
5	2. O registro 00 deve ter 93 campos.	Registro 00 com número de campos diferente de 93, foram encontrados <número de campos> campos.	validação	Recusada	Mudança
6	3. O registro 10 deve ter 168 campos.	Registro 10 com número de campos diferente de 168, foram encontrados <número de campos> campos.	validação	Recusada	Exclusão
7	4. O registro 20 deve ter 74 campos.	Registro 20 com número de campos diferente de 74, foram encontrados <número de campos> campos.	validação	Recusada	
8	5. O registro 30 deve ter 95 campos.	Registro 30 com número de campos diferente de 95, foram encontrados <número de campos> campos.	validação	Recusada	
9	6. O registro 40 deve ter 7 campos.	Registro 40 com número de campos diferente de 7, foram encontrados <número de campos> campos.	validação	Recusada	
10	7. O registro 50 deve ter 31 campos.	Registro 50 com número de campos diferente de 31, foram encontrados <número de campos> campos.	validação	Recusada	
11	8. O registro 60 deve ter 45 campos.	Registro 60 com número de campos diferente de 45, foram encontrados <número de campos> campos.	validação	Recusada	
12	9. O registro 00 deve vir no começo do arquivo ou após um registro 40 ou após um registro 60.	Registro 00 declarado em linha inadequada, após o registro <registro>.	validação	Recusada	
13	10. O registro 10 deve vir após um registro 00.	Registro 10 declarado em linha inadequada, após o registro <registro>.	validação	Recusada	
14	11. O registro 20 deve vir após um registro	Registro 20 declarado em linha	validação	Recusada	

Fonte: Elaborado pelo autor

Os dados são disponibilizados para abertura e fechamento por cerca de 60 dias, período no qual podem ocorrer problemas que impossibilitem o envio dos dados, como indisponibilidade de acesso à internet ou à plataforma (gerada pela alta demanda de acesso). Por esse motivo, a Secretaria da Educação do Estado de Goiás (SEDUC) assume a responsabilidade por realizar a migração dos dados das unidades de ensino públicas municipais e estaduais do estado de Goiás que não possuem suporte de algum software de gestão.

Para que seja possível o envio dos dados ao educacenso, inicialmente é necessário a criação de rotinas no banco de dados para obter dados categorizados, a fim de atender a essa finalidade. E para seguir o *layout* atualizado é desenvolvido rotinas específicas para cada tópico. O *layout* da primeira etapa a ser seguido se divide em 7 categorias:

- Registro00 - Identificação escolar
- Registro10 - Caracterização física da escola
- Registro20 - Cadastro de turma
- Registro30 - Cadastro pessoa física (aluno/docente)
- Registro40 - Gestor escolar
- Registro50 - Identificação profissional escolar em turma
- Registro60 - Identificação aluno em turma

Cada categoria possui suas próprias regras e características. Para cada categoria, é criada uma entidade no banco de dados, que receberá uma carga de dados de acordo com sua categoria. Por fim, os dados serão cruzados a fim de obter os dados consolidados, possibilitando a geração dos arquivos finais, como mostrado na Figura 2.

Figura 2 - Arquivo de texto final gerado

	52022706	2172022																															
00	52022706	2172022	01/12/2022	01/12/2022	CENTRO DE ENSINO EM PERIODO INTEGRAL ARY RIBEIRO VALADAO																												
10	52022706	202210	1411	1A	1 07 30 17	00 0 1 1 1 1 1 0 1 0 0 1 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 1 1 0 1																											
20	52022706	202210	6666	AEEA	1 07 30 09 30 0 1 1 1 1 1 0 0 0 1																												
20	52022706	202211	6666	AEEB	1 09 35 11 55 0 1 0 0 0 1 0 0 0 1																												
20	52022706	202212	6666	AEEC	1 12 00 13 00 0 1 1 0 0 0 0 0 0 1																												
20	52022706	202213	6666	AEEB	1 13 00 15 00 0 1 1 1 1 1 0 0 0 1																												
20	52022706	202214	6666	AEEE	1 15 05 17 00 0 0 0 0 1 0 0 0 0 1																												
20	52022706	202221	1411	1B	1 07 30 17	00 0 1 1 1 1 1 0 1 0 0 1 1 0																											
20	52022706	202223	1412	2A	1 07 30 17	00 0 1 1 1 1 1 0 1 0 0 1 1 0																											
20	52022706	202224	1412	2B	1 07 30 17	00 0 1 1 1 1 1 0 1 0 0 1 1 0																											
20	52022706	202225	1412	2C	1 07 30 17	00 0 1 1 1 1 1 0 1 0 0 1 1 0																											
20	52022706	202226	1413	3A	1 07 30 17	00 0 1 1 1 1 1 0 1 0 0 1 1 0																											
20	52022706	202227	1413	3B	1 07 30 17	00 0 1 1 1 1 1 0 1 0 0 1 1 0																											
20	52022706	202228	1413	3C	1 07 30 17	00 0 1 1 1 1 1 0 1 0 0 1 1 0																											
20	52022706	202229	1411	1C	1 07 30 17	00 0 1 1 1 1 1 0 1 0 0 1 1 0																											
30	52022706	19116672970	122184217474	DANIEL ALVARENGA TAVARES	22/08/2006	1	LUCIANA DA CUNHA																										

Fonte: Elaborado pelo autor.

O processo de migração é realizado por meio de arquivos de texto formatados de acordo com o layout disponibilizado pelo INEP. No entanto, a plataforma possui várias validações e, caso alguma incoerência seja encontrada em algum arquivo enviado, ele será rejeitado, resultando em retrabalho. Será gerado um relatório de erros (exibido na Figura 3) que apontará as inconsistências identificadas no arquivo submetido. Portanto, será necessário corrigir manualmente o arquivo de texto ou gerar um novo, dependendo do motivo da recusa.

Figura 3 - Relatório de erros



MINISTÉRIO DA EDUCAÇÃO
INSTITUTO NACIONAL DE ESTUDOS E PESQUISAS EDUCACIONAIS ANÍSIO TEIXEIRA
DIRETORIA DE ESTATÍSTICAS EDUCACIONAIS

RELATÓRIO DOS ERROS IDENTIFICADOS NA VALIDAÇÃO DO ARQUIVO DE MIGRAÇÃO - COLETA 2022

Fonte: Inep/MEC - Migra Dados - OnLine - Tipo: MC
 Data de envio: 15/06/2022 13:56:37
 Arquivo verificado: 52097129_1562022.txt (480810)
 Tamanho do arquivo: 57030 bytes
 Data de geração do relatório: 15/06/2022 14:15:42
 Versão do Migração: 7.4.325
 Versão do Negócio: 2.4.873

Este documento está limitado a 500 erros.

Nº	Entidade	Reg./Reg.	Linha	Campo	Conteúdo	Id. na escola	Mensagem
1	52097129	0/2	1	"18 - Código do órgão regional de ensino"	00000	52097129	"O campo deve estar de acordo com a tabela de órgãos regionais de ensino."
2	52097129	0/3	1	"18 - Código do órgão regional de ensino"	00000	52097129	"O campo deve ser preenchido de acordo com o estado da escola."
3	52097129	0/1	1	"33 - Secretaria estadual"		52097129	"O campo não foi preenchido quando deveria ser preenchido."
4	52097129	0/1	1	"35 - Não possui parceria ou convênio"		52097129	"O campo não foi preenchido quando deveria ser preenchido."
5	52097129	0/1	1	"360041 - Formas de contratação entre a Administração Pública e outras instituições"		52097129	"Formas de contratação entre a Administração Pública e outras instituições" não foi preenchido corretamente. Não podem ser informadas todas as opções com valor igual a 0 (Não)."
6	52097129	0/2	1	"420084 - Número de matrículas atendidas por meio da parceria ou convênio"		52097129	"O campo não foi preenchido quando deveria ser preenchido."
7	52097129	10/1	2	"145 - Nenhum dos instrumentos listados"		52097129	"O campo não foi preenchido quando deveria ser preenchido."
8	52097129	20/3	3	"34 - Série/ano (séries anuais)"	0	20221	"O campo foi preenchido quando deveria não ser preenchido."

Fonte: Elaborado pelo autor.

Diante desse contexto, este projeto visa utilizar a linguagem Perl para facilitar o processo de correção dos arquivos de texto que venham a ser recusados, reduzindo assim o retrabalho e otimizando o processo de correção.

2.1. Perl

2.1.1. História

Perl é uma linguagem de programação dinâmica e interpretada criada em 1987, pelo desenvolvedor Larry Wall. Em sua rotina diária Larry lidava com muitos relatórios e tinha que efetuar muitas tratativas em arquivos de texto, com base nessa necessidade ele desenvolveu a linguagem Perl cujo nome significa "*Practical Extraction and Reporting Language*" (em português, Linguagem Prática de Extração e Relatórios).

Inicialmente o Perl tinha como objetivo facilitar a manipulação de textos, porém a linguagem se expandiu muito e atualmente possui diversos recursos e pode ser aplicada em inúmeros contextos como desenvolvimento web, gerenciamento de sistemas operacionais, gerenciadores de banco de dados, interface gráfica entre outros (BARRO, 2022).

2.1.2. Como funciona o Perl

Perl é uma linguagem de programação que se comporta como uma linguagem interpretada; no entanto, ela compila dados sem gravá-los em um executável. Em vez disso, mantém os dados armazenados em memória, o que permite uma execução eficiente. Os programas desenvolvidos em Perl podem ser escritos e executados tanto em linha de comando quanto em *scripts*. Os arquivos são compreendidos pelo interpretador/compilador, que converte o código para linguagem de máquina, e então o processador lê e executa (MARIANO e de MELO-MINARDI, 2016).

2.1.3. Prós e Contras da Linguagem Perl

Assim como qualquer linguagem de programação, o Perl possui prós e contras. Não fica de fora a existência de diversas vantagens em sua utilização, que vão além da alta performance na manipulação de textos. O Perl possui boa compatibilidade com linguagens de marcação, possibilitando grandes resultados no desenvolvimento web. Além disso, pode ser utilizado em diversos sistemas operacionais e possui extensa integração com sistemas gerenciadores de banco de dados. Apresenta alta versatilidade, podendo ser usado para diversas finalidades, e conta com uma grande quantidade de módulos que visam aumentar suas funcionalidades. Há também uma grande comunidade de usuários (Perl Mongers), além de ser *open source*. No entanto, nos quesitos que são insuficientes, pode-se citar a questão do suporte à interface gráfica. Existem diversos módulos que auxiliam nesse aspecto, porém, o entendimento não é algo trivial. Contudo, isso não impossibilita sua utilização (BARRO, 2022).

2.1.4. CPAN - Módulos e *Frameworks*

A linguagem de programação Perl possui uma comunidade considerável de usuários e desenvolvedores, que apoiam o desenvolvimento e aprimoramento de diversos módulos, classes, *scripts* e *frameworks*, e reúne todos esses recursos em uma rede abrangente de arquivos CPAN. O CPAN é um repositório de arquivos Perl que possui uma vasta gama de recursos já desenvolvidos em Perl. Entre eles, encontram-se:

- DBIx::Class - Modelo de mapeamento Objeto Relacional

- *Template Toolkit* - Sistema de processamento de *templates*
- WWW::Mechanize - Simula um navegador Web em um objeto Perl
- Catalyst (software) - *Framework MVC* para aplicações Web

Além desses recursos, o repositório CPAN, possui uma grande diversidade de módulos e distribuições e está ativo desde 1995, em constante crescimento conforme exibido na Figura 4 (HIETANIEMI,1995).

Figura 4 - Repositório CPAN



Fonte: Elaborado pelo autor.

2.1.5. Variáveis e Tipos de Dados

As variáveis são elementos declarados que armazenam dados na memória e podem ser utilizadas posteriormente dentro de um programa. Em Perl, as variáveis são escalares, o que significa que podem armazenar qualquer tipo de dado e sua manipulação é flexível, permitindo alterações sem gerar erros no código. Para exibir informações na tela, utiliza-se o comando *print*, que pode ser utilizado tanto para mostrar mensagens fixas quanto para exibir o conteúdo de uma variável.

Em Perl, isso é feito de forma simples. O ambiente de trabalho Visual Studio Code foi utilizado para desenvolver o código.

```

1  # Declaração de variavel
2  my $frase;
3  # Atribuição de conteudo a variavel frase
4  $frase = "Hello world!\n";
5  # Imprime o texto sem utilizar variavel
6  print ("Hello world!\n\n");
7  # Imprime o conteudo da variavel frase
8  print $frase;
9  # Fim do program

```

Em Perl, existem três tipos principais de variáveis: escalares, *arrays* e *hashes* (também conhecidos como *arrays* associativos). Essas variáveis podem armazenar diferentes tipos de dados, incluindo textos (*strings*), números inteiros (*int*) e números de ponto flutuante (*float*). O quadro 1 ilustra essa diversidade de tipos de variáveis em Perl.

Tabela 1 - Tipos de Variáveis

Tipo de variável	Característica	Exemplo de declaração
String	Representam cadeias de caracteres. Devem ser declarados sempre entre aspas simples ou duplas.	\$variavel = "Isto eh uma string";
Int	Tipo de variável que representa nú-meros inteiros.	\$variavel = 1;
Float	Variável que permite números decimais.	\$variavel = 1.022;

Fonte: Elaborado pelo autor.

As variáveis escalares em Perl são representadas pelo símbolo "\$" (cifrão). Uma variável escalar pode armazenar um único elemento ou valor, seja ele um número ou um texto. Por outro lado, os *arrays* em Perl são representados pelo símbolo "@" (arroba) e são semelhantes a listas, permitindo a atribuição de vários valores. Um *array* pode conter múltiplas variáveis escalares atribuídas. Já os *hashes* em Perl são representados pelo símbolo "%" (porcentagem) e são conhecidos como listas associativas. Em resumo, cada item de um *hash* possui um valor associado a uma chave (*key* → *value*). O código ilustra a utilização dos três tipos de variáveis mencionados.

```

1 # Variavel do tipo: Escalar
2 my $nome = "Alexandre";
3 my $morada = "Brasil, América do Sul";
4 my $numero = 7110432
5 print $nome;
6
7 # Varivel do tipo: Array
8 my @array = (teste, seila, portugal);
9 my $valor = @array;
10 my @array = (1..10);
11 my @array = (teste, tx, px);
12 my @array = qw(teste tx pc);
13 print(@array);
14
15 # Variavel do tipo: Hash
16 my %hash = ("linhal" => "Teste");
17 my %hash = ("linhal" , Teste);
18 print %hash
19
20
21
22

```

```

1 # Variavel local do tipo: Escalar
2 my $textoLocal = "Hello World\n";
3 print $textoLocal;
4
5 # Variavel global do tipo: Escalar
6 our $textoGlobal;
7 print $textoGlobal;
8
9 # Variaveis do tipo: Hash
10 my %profile = (
11     Nome => "Fulano de tal",
12     "Endereco" => "Rua 123 Quadra XX Lote XX",
13     Cidade => "Goiania",
14     Contatos => {
15         Telefone => "62 9999999999",
16         Email => 'fulano@gmail.com',
17         Outros => {
18             Caixa_Postal => 'fulano123'
19         }
20     }
21 );
22 print $profile{Contatos}->{Outros}->{Caixa_Postal};

```

2.1.6. Operadores Aritméticos

Operadores matemáticos são essenciais em praticamente todas as linguagens de programação, pois permitem realizar cálculos básicos como adição, subtração, multiplicação, divisão, módulo e potenciação. Em Perl, esses operadores podem ser utilizados de maneira simples.

```

1 my $A = 2;
2 my $B = 3;
3 my $OPERACAO;
4
5 # Realiza a soma
6 $OPERACAO = $A + $B;
7 print "Soma: ".$OPERACAO."\n";
8
9 # Realiza subtração
10 $OPERACAO = 5 - 3;
11 print "Subtracao: ".$OPERACAO."\n";
12
13 # Realiza multiplicacao
14 $OPERACAO = 2 * 3;
15 print "Multiplicacao: ".$OPERACAO."\n";
16
17 # Realiza divisao
18 $OPERACAO = 4 / 2;
19 print "Divisao: ".$OPERACAO."\n";
20
21 # Realiza calculo de expressão
22 $OPERACAO = (5 - 3) * (2 * 3) / (4 / 2);
23 print "Expressao: ".$OPERACAO."\n";

```

Além dos operadores básicos, existem outros operadores que são de grande utilidade, como o operador de módulo (%), que retorna o resto da divisão de um número. Esse operador é especialmente útil na identificação de números pares ou ímpares.

```
1 my $num = 10;
2 my $resto = num % 2;
3
4 if ($resto == 0) {
5     print "Par";
6 }else{
7     print "Impar";
8 }
```

Operações de incremento e decremento também é algo muito usual em Perl, nessas operações é possível incrementar ou decrementar variáveis numéricas através de operadores como “++” ou “--”, conforme exemplificado no quadro 2.

Tabela 2 - Incremento/Decremento de variável

Operações	Característica
++\$valor	Soma um a \$valor e depois retorna o valor
\$valor++	Retorna o valor de \$valor e depois soma um
--\$valor	Diminui um de \$valor e depois retorna valor
\$valor--	Retorna o valor de \$valor e depois diminui um

Fonte: Elaborado pelo autor.

2.1.7. Operadores Relacionais e Lógicos

Assim como os operadores aritméticos, existem também os operadores lógicos e relacionais, que auxiliam na realização de comparações e na definição da veracidade de uma determinada expressão. O quadro 3 mostra todos os operadores lógicos e relacionais disponibilizados pelo Perl.

Tabela 3 - Operadores Lógicos

Operador	Significado
==	Verifica se uma variável numérica é igual a outra.
>	Verifica se uma variável numérica é maior que outra.
<	Verifica se uma variável numérica é menor que outra.
>=	Verifica se uma variável numérica é maior ou igual a outra.
<=	Verifica se uma variável numérica é menor ou igual a outra.
!=	Verifica se uma variável numérica é diferente de outra.
<=>	Realiza comparação numérica.
eq	Verifica se uma variável textual (string) é igual a outra.
ne	Verifica se uma variável textual (string) é diferente de outra.
lt	Verifica se a ordenação de uma variável textual (string) é inferior a outra.
gt	Verifica se a ordenação de uma variável textual (string) é superior a outra.
le	Verifica se a ordenação de uma variável textual (string) é inferior ou igual a outra.
ge	Verifica se a ordenação de uma variável textual (string) é superior ou igual a outra.
.	Operador de concatenação de strings.
=~	Busca padrões em strings.
!~	Similar a "=~", entretanto verifica se o padrão não for encontrado.
=	Operador de atribuição. Válido para dados de quaisquer tipos.

Fonte: Elaborado pelo autor.

2.1.8. Input de Dados

Ao desenvolver uma aplicação, é importante realizar testes de forma mais dinâmica, permitindo a entrada de dados durante a execução do programa, sem a necessidade de acessar o código fonte e inserir valores de variáveis de forma fixa. Isso pode ser alcançado utilizando o comando STDIN, que é delimitado

pelos sinais de maior e menor (" $<$ " " $>$ "). Em Perl, esses sinais são chamados de operadores diamante e indicam que o script deve receber dados naquele ponto.

```
1 # Recebe um numero digitado pelo usuario
2 print "Digite um numero: ";
3 my $numero = <STDIN>;
4 my $resto = $num % 2;
5 print "Seu numero eh: ";
6 print $numero;
7 print "\nO resto da divisao por 2 eh: ";
8 print $resto."\n\n\n";
```

Ao realizar a entrada de dados utilizando o comando STDIN, é necessário pressionar a tecla ENTER para que o script continue sua execução. No entanto, ao fazer isso, o Perl registra uma quebra de linha "\n" quando o botão ENTER é clicado. Em alguns contextos, isso pode não ser desejado. Por exemplo, quando queremos que o conteúdo digitado não tenha uma quebra de linha e seja exibido na mesma linha de execução. Uma maneira de resolver esse problema é utilizando a função *chomp*, que remove o caractere especial "\n" do final da entrada de dados.

```
1 # Funcao chomp
2 print "Digite seu nome: ";
3 my $nome = <STDIN>;
4
5 print "\nOla $nome, tudo bem com voce?\n\n";
6
7 # Corrigindo o problema
8 chomp($nome);
9 print "Ola $nome, tudo bem com voce?\n\n";
```

2.1.9. Comando Use

Ao desenvolver uma aplicação em Perl, é interessante utilizar módulos já desenvolvidos e publicados no repositório CPAN, pois já foram testados e cada um executa uma determinada função. Isso pode economizar muito tempo de desenvolvimento. No entanto, para fazer uso dos módulos, é necessário carregá-los no código, o que é possível utilizando o comando "use". Esse comando também pode ser usado para carregar os chamados *pragmas*, que são meios de alterar o comportamento padrão do código em Perl. É possível ativar esses *pragmas* usando os comandos "use strict" e "use warnings".

O comando `"use strict"` tem a função de interromper a execução do código se algum erro for identificado, enquanto o comando `"use warnings"` envia avisos ao usuário durante a execução do código. Seguindo as boas práticas, é sempre interessante utilizar ambos.

2.1.10. Utilização da Linguagem Perl

Assim como nas demais linguagens de programação, para uma boa utilização, é necessário estudar a sintaxe e aplicar os conceitos da lógica. Com base nisso, foram realizados alguns testes com o objetivo de identificar o comportamento da linguagem para a finalidade do trabalho.

No contexto de sintaxe, as estruturas condicionais em Perl não se distanciam das demais linguagens de programação. É possível fazer uso da estrutura `if` e `else` para validar uma variável ou um bloco de dados, e o mesmo pode ser aplicado com o uso da estrutura `"switch case"`, obtendo o mesmo resultado. O uso de cada estrutura dependerá do contexto da aplicação.

```
1  my $num = 0.3; #rand();
2
3  #Utilizando IF ELSE
4  if ($num > 0.7) {
5      print("$num e maior que 0.7");
6  } elsif($num > 0.4) {
7      print("$num e maior que 0.4");
8  }else{
9      print("$num e menor que ambos acima");
10 }
11
12 #Utilizando Swicth Case
13 given ($num) {
14     when ($num > 0.7) {
15         say "$num e maior que 0.7";
16     }
17     when ($num > 0.4) {
18         say "$num e maior que 0.4";
19     }
20     default {
21         say "$num e menor que ambos acima";
22     }
23 }
```

A utilização de *arrays* é muito importante em diversos tipos de aplicações, principalmente quando há a necessidade de armazenar um conjunto de dados em uma variável. A declaração de um *array* é precedida pelo símbolo '@'. O

código exibe a declaração de um *array* de cores em Perl e demonstra algumas das formas possíveis de iterar e exibir os dados armazenados no *array*, utilizando estruturas de repetição como *'for'* ou *'foreach'*.

```
1 #Definindo variaveis do tipo ARRAY
2 my @cores = (
3     "Azul",
4     "Amarelo",
5     "Ciano",
6     "Magenta"
7 );
8
9 #Forma de iterar sem indice, por convenção da linguagem PERL
10 for (@cores) {
11     print $_ . "\n";
12 }
13
14 #Outra forma de iterar, com indice
15 foreach $i (@cores){
16     print "$i "
17 }
18
19 #Exibindo um range
20 print @cores[0..3];
```

A aplicação a ser desenvolvida requer a manipulação de vários arquivos de texto. Para isso, é necessário que a linguagem Perl possibilite o acesso a diretórios, a leitura e a modificação desses arquivos. O código exibe como é feito o acesso a um diretório, a abertura de um arquivo e a exibição do seu conteúdo.

```
1 # Abrindo o arquivo
2 open(fh, "teste.txt") or die "File '$filename' can't be opened";
3
4 # Lê apenas primeira linha do arquivo
5 $firstline = <fh>;
6 print "$firstline\n";
7
8 #Lê todas as linhas do arquivo
9 while(<fh>) {
10     print "$_";
11 }
```

É possível verificar o tipo de arquivo que será manipulado, possibilitando a filtragem, a fim de evitar erros. Ao buscar por um arquivo em um diretório, é possível que existam arquivos de tipos diferentes com o mesmo nome. O código exibe como são utilizados alguns argumentos com *"-e"*, que é utilizado no *"if"* para verificar se o arquivo realmente existe. Antes de exibir na tela, é utilizado o comando *"join"*, que tem a função de unir as *strings*.

```

1 #Verificando o tipo de arquivo
2 my $file = "teste.txt";
3 my (@description, $size);
4 if (-e $file) {
5     push @description, 'um arquivo binario' if (-B _);
6     push @description, 'um socket' if (-S _);
7     push @description, 'um arquivo de texto' if (-T _);
8     push @description, 'um arquivo bloqueado' if (-b _);
9     push @description, 'um arquivo com caractere especial' if (-c _);
10    push @description, 'um diretorio' if (-d _);
11    push @description, 'um executavel' if (-x _);
12    push @description, (($size = -s _) ? "$size bytes" : 'Vazio');
13    print "$file e ", join(' ', @description), "\n";
14 }
15
16 print "\n-----\n";
17

```

Com base nesses testes é possível afirmar que a linguagem oferece recursos suficientes para o início do desenvolvimento da aplicação.

2.2. API

API é a sigla para *Application Programming Interface* ou, em português interface de programação de aplicativos. API é um conjunto de regras e protocolos que permitem que vários *softwares* se comuniquem entre si. Ela define como os componentes do *software* devem interagir e se comunicar, o que permite que outros aplicativos usem suas funcionalidades. Os principais métodos implementados pelas APIs são:

GET: Envia uma solicitação para obter informações sobre um recurso específico. Geralmente é usado para buscar e coletar dados.

POST: Envia dados para serem processados por um recurso específico. É utilizado para criar ou adicionar novos dados.

PUT: É um método utilizado para atualizar um recurso específico com os dados fornecidos. É utilizado para modificar ou substituir dados existentes.

DELETE: É um método para remover um dado ou um conjunto de dados em particular.

As aplicações que utilizam todas essas técnicas são comumente chamadas de APIs REST, que significa "Transferência Representacional de Estado" (*Representational State Transfer*, em inglês). Não é uma linguagem de programação, mas sim um modelo de arquitetura que fornece diretrizes para que os sistemas distribuídos se comuniquem diretamente, utilizando os princípios e protocolos existentes da Web (COSTA, 2022).

2.2.1. Para que serve API

As APIs possuem diversas funcionalidades e desempenham um papel estratégico nas operações diárias. Elas são usadas para integrar vários sistemas, a fim de facilitar seu uso. Atualmente, é muito comum sua utilização em aplicativos e sistemas empresariais com o objetivo de efetuar integrações com outros *softwares*.

Um exemplo simples de uso pode ser encontrado no cenário em que um colaborador precisa emitir uma nota fiscal para o cliente, a fim de finalizar uma venda. Nesse sentido, o sistema próprio pode fazer uso de uma API que conecta o sistema de gestão da empresa ao sistema de emissão de notas da prefeitura. Dessa forma, o colaborador precisa apenas inserir os dados necessários para a emissão da nota, e o tráfego de informações fica a critério das requisições feitas pela API, resultando em um processo finalizado em pouco tempo.

Esse é apenas um exemplo de utilidade de uma API, mas existem diversos outros cenários de uso de APIs que se enquadram no cotidiano (COSTA, 2022).

2.2.2. Por que usar API?

No quesito empresa, sua utilização é fundamental, pois otimiza processos, agregando agilidade, automação e simplicidade em diversos fluxos do sistema. Existem várias vantagens em seu uso, como a automatização de atendimento ao cliente por meio de *chatbots*, possibilitando um atendimento 24 horas por dia, todos os dias da semana, a automatização de processos, campanhas de *marketing*, publicidade, envio de e-mails e mensagens em poucos segundos. Além disso, sua utilização não se restringe apenas a uma área, como financeiro ou Tecnologia da Informação (T.I), ela se estende para todas as áreas de uma empresa, abrangendo tanto os processos internos quanto os externos.

2.2.3. API em Perl

A implementação de uma API em Perl pode ser realizada através do uso de *frameworks*, que consistem em ferramentas com componentes prontos para reutilização. Um exemplo disso é o *Dancer*, que se destaca por ser leve, flexível

e permitir o desenvolvimento rápido e preciso de aplicativos Web em Perl. O Dancer oferece recursos como roteamento, gerenciamento de requisições e respostas em métodos *Hypertext Transfer Protocol* (HTTP), além de suporte à serialização em formatos *JavaScript Object Notation* (JSON) e *eXtensible Markup Language* (XML). Além disso, também oferece configurações de segurança, como plugins de autenticação e autorização (SUKRIEH, 2021).

Outro *framework* para implementações de API em Perl é o Mojolicious, que é moderno e robusto na criação de APIs, possui alto desempenho e uma ampla gama de recursos para o desenvolvimento de aplicações web e APIs.

Assim como o Dancer, o Mojolicious possui um roteamento flexível, fornece suporte a *Websocket*, realiza o processamento de requisições e respostas HTTP, a renderização de *templates*, possui suporte à autenticação e autorização, além de diversos outros recursos avançados (RIEDEL, 2008).

Existem outros *frameworks* que possibilitam a implementação de APIs em Perl. Este projeto utilizará o *framework* Mojolicious para o desenvolvimento de uma API, bem como a implementação de rotas (*endpoints*), a fim de auxiliar a comunicação da interface com o *back-end*.

2.2.4. Rotas

As rotas em uma API são os caminhos ou URLs que os clientes utilizam para acessar recursos e funcionalidades específicas. Elas representam os diferentes pontos de entrada para as requisições realizadas. Cada rota geralmente corresponde a uma ação ou operação específica que o usuário deseja executar. Por exemplo, uma rota pode ser `/escolas` para lidar com operações relacionadas ao retorno de unidades escolares no sistema.

2.2.5. Endpoints

Os endpoints assim como o nome sugere são os pontos finais ou destinos específicos dentro de uma rota em uma API. Eles indicam a operação que deve ser realizada em relação a um recurso específico. Os endpoints são definidos pela combinação da rota base e um identificador exclusivo para o recurso em questão. Por exemplo, em uma rota `/escolas`, os endpoints podem ser

"/escolas/52012345" para obter informações de uma escola específica ou
"/escolas" para criar um registro de escola.

3. DESCRIÇÃO GLOBAL

Este capítulo apresenta as fases de desenvolvimento, descrição das interfaces da ferramenta suas funcionalidades e limitações.

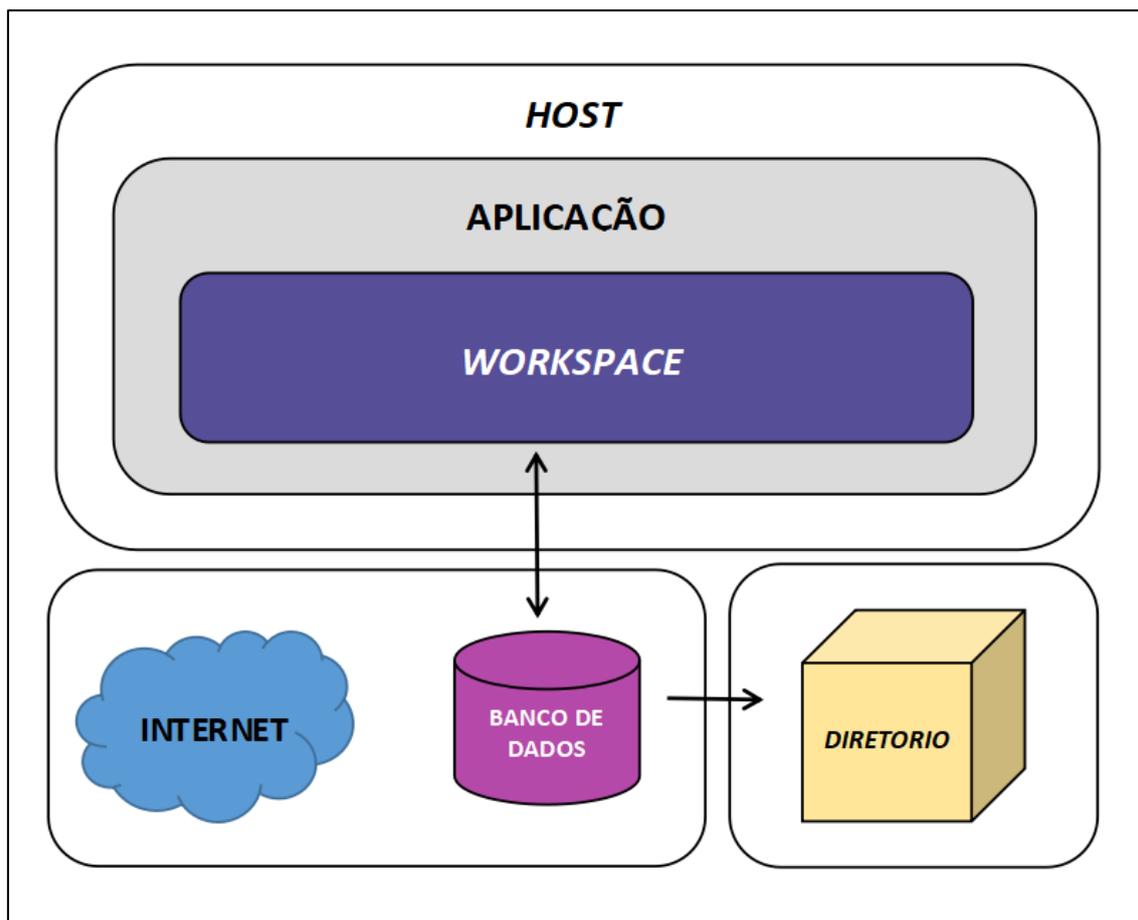
3.1. Aspecto Geral do Produto

Os usuários da ferramenta poderão acessar a aplicação pela internet por meio de um link de acesso. Ao se conectar, o usuário visualizará a tela inicial, que contém dois botões principais: um para acesso à matrícula inicial e outro para acesso à situação do aluno. Além dos botões, o usuário terá acesso a um acordeão com quatro informações sobre conteúdos relacionados à ferramenta.

Os botões principais direcionam o usuário à *Workspace*, onde será realizado todo o tratamento dos dados. A tela permite que o usuário digite o código de uma unidade escolar e faça uma pesquisa. Se a unidade for válida e houver registro no banco de dados, os dados serão exibidos em uma tabela. Nesse ponto, será possível acessar uma célula específica e alterar seu conteúdo. O registro alterado é salvo assim que o foco é removido da célula.

Após concluir todas as alterações, o usuário deve clicar no botão "exportar". Ao pressioná-lo, será gerado um arquivo de texto no formato padrão solicitado pelo INEP, em um diretório fixo. O diagrama de blocos que ilustra essa representação pode ser verificado na Figura 5.

Figura 5 - Diagrama em blocos representando a aplicação



Fonte: Elaborado pelo autor.

Após isso, já é possível realizar o envio dos arquivos corrigidos ao educacenso.

3.2. Interface do Sistema

A ferramenta possui as seguintes interfaces:

- **Tela Home:** É a tela inicial da aplicação que apresenta os botões "Matrícula Inicial" e "Situação do Aluno". Esses botões direcionam o usuário para a tela em que será realizado o tratamento dos dados da primeira ou segunda etapa do Censo Escolar. Além disso, a tela inicial disponibiliza a opção "Saiba mais", que exibe informações sobre como utilizar a ferramenta e detalhes relacionados à aplicação.
- **Tela Matrícula Inicial:** É a tela de tratamento dos dados da primeira etapa do Censo Escolar, que possui o campo "Código MEC" responsável por

receber o código da unidade escolar. Além disso, apresenta um componente de navegação com as categorias correspondentes a cada tipo de registro da primeira fase (00, 10, 20, 30, 40, 50, 60). O botão "Pesquisar" é utilizado para buscar os dados de uma escola e exibi-los na tabela de categorias. Também há o botão "Consultar Escola", que lista todas as escolas validadas, podendo conter erros a serem corrigidos. Por fim, o botão "Exportar" gera o arquivo de texto e salva em um diretório específico.

- **Tela Situação do Aluno:** É a tela de tratamento dos dados da segunda etapa do Censo Escolar. Possui o campo "Código MEC", que recebe o código da unidade escolar. Além disso, conta com um componente de navegação com as categorias de cada tipo de registro da segunda fase (89, 90, 91). Na interface, encontram-se os botões "Pesquisar" — utilizado para buscar os dados de uma escola e exibi-los na tabela de categorias —, "Consultar Escola" — que lista todas as escolas já validadas, podendo conter erros a serem corrigidos —, e "Exportar" — responsável por gerar o arquivo de texto e salvá-lo em um diretório.

3.3. Interface do Usuário

O usuário precisar possuir um computador com qualquer sistema operacional com acesso à Internet.

3.4. Funções da Ferramenta

A ferramenta possui as seguintes funções:

- Pesquisar unidades escolares
- Listar unidades disponíveis para correção
- Alterar dados das unidades escolares
- Exportar arquivo em formato TXT

3.5. Características do Usuário

A ferramenta é orientada para usuários que já possuem ciência de como funciona o processo de coleta estatística do censo escolar em ambas as fases, pois visa facilitar a correção de erros em arquivos de texto.

3.6. Suposições e Dependência

A ferramenta é uma aplicação Web e faz comunicação com o servidor de banco de dados SQL Server, portanto é necessário a conexão com a internet.

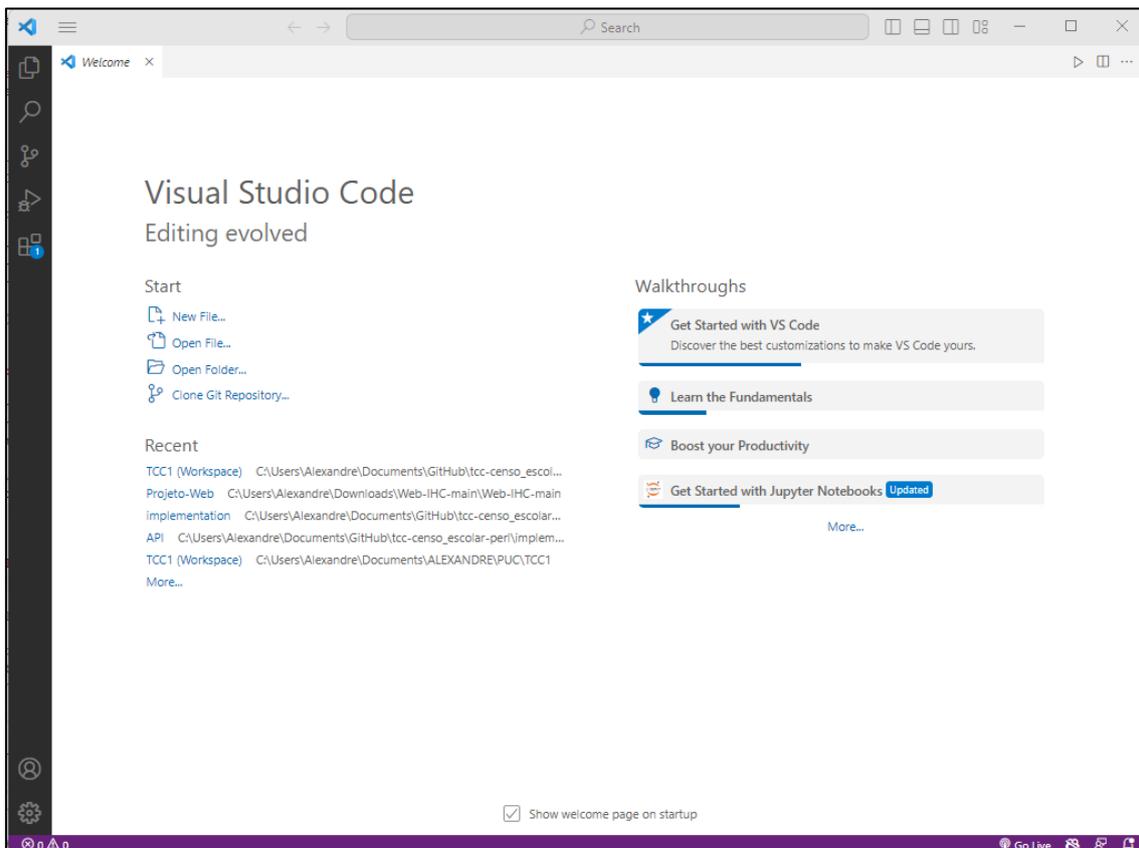
4. DESENVOLVIMENTO DA FERRAMENTA

Este capítulo apresenta as ferramentas utilizadas para o desenvolvimento da aplicação. O ambiente escolhido para a implementação foi a IDE Visual Studio Code. A ferramenta utilizada para controle de versão foi o GitHub Desktop e o banco de dados utilizado foi o SQL Server, juntamente com sua ferramenta SQL Server Management Studio (SSMS).

4.1. Visual Studio Code

Para a implementação da aplicação foi utilizado a IDE Visual Studio Code para a escrita do código fonte. A ferramenta possui controle de versionamento Git incorporado, realce de sintaxe, complementação inteligente de código (MICROSOFT, 2023). A Figura 6 exhibe a interface da ferramenta Visual Studio Code.

Figura 6 - Interface do Visual Studio Code

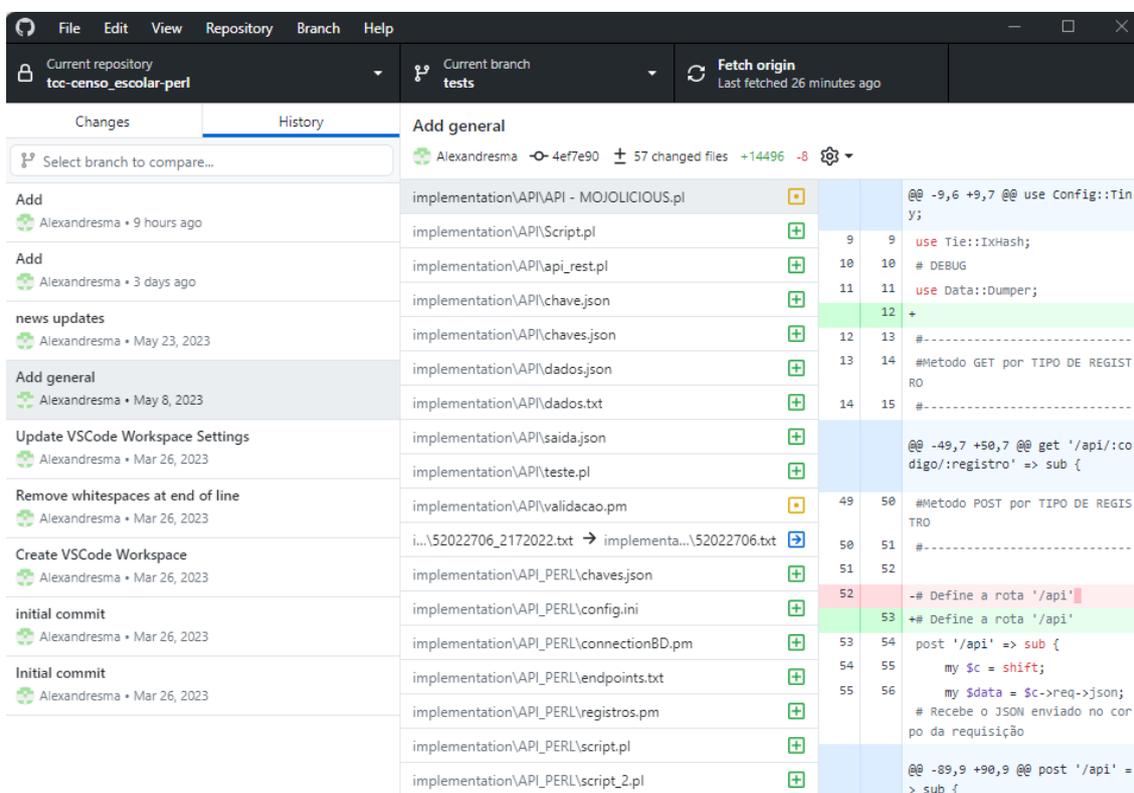


Fonte: Elaborado pelo autor.

4.2. GitHub

GitHub é uma ferramenta de hospedagem de código-fonte e arquivos, que utiliza o sistema de controle de versão distribuído Git. Para gerenciar as alterações dos scripts da aplicação, ele registra o histórico de edições dos arquivos de desenvolvimento do projeto, permite a criação de *branches* (ramificações de versões), comparações de código e rastreabilidade das mudanças ao longo do tempo (GITHUB, 2023), como exemplificado na Figura 7.

Figura 7 - Interface do GitHub Desktop



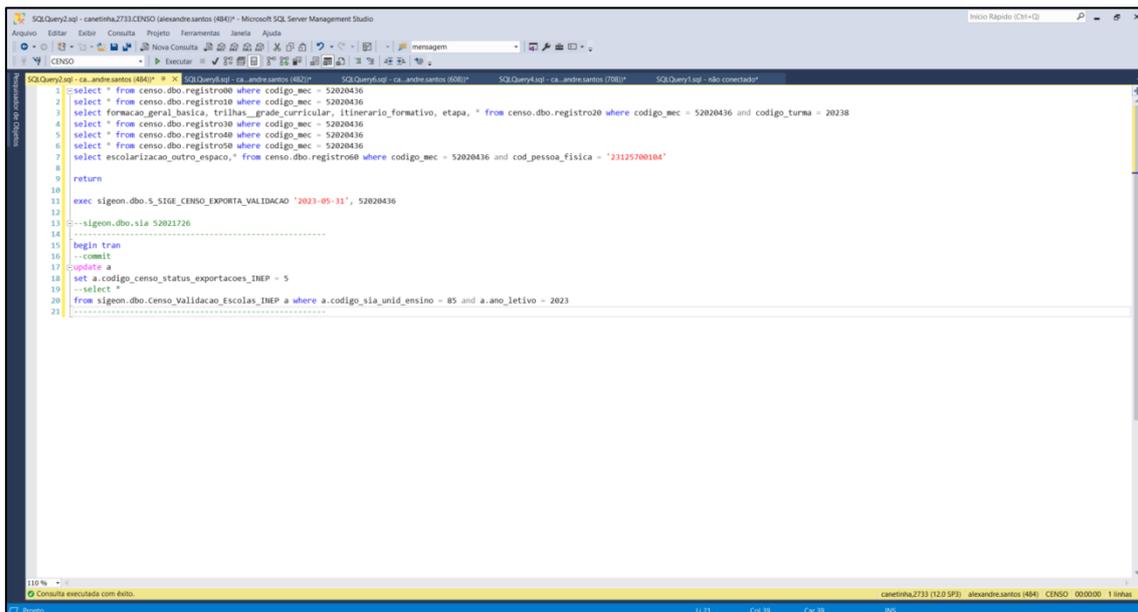
Fonte: Elaborado pelo autor.

4.3. SQL Server Management Studio (SSMS)

O SQL Server é um sistema de gerenciamento de banco de dados relacional (SGBDR) desenvolvido pela Microsoft, que proporciona um ambiente seguro, eficiente e robusto para o armazenamento de dados (MICROSOFT, 2022). Entre seus recursos, destaca-se o SQL Server Management Studio

(SSMS), uma interface gráfica criada para gerenciar e administrar o banco de dados SQL Server. Essa ferramenta oferece diversos recursos, como a linguagem Transact-SQL (T-SQL), que facilita o desenvolvimento e a criação de objetos SQL. A Figura 8 exibe a interface do SSMS.

Figura 8 - Interface do SQL Server Management Studio (SSMS)



Fonte: Elaborado pelo autor.

5. IMPLEMENTAÇÃO E RESULTADOS

Este capítulo apresenta as telas da aplicação, descrevendo suas funções, as ações que o usuário pode realizar e demonstra como é feita a busca dos dados, correção de dados de cada registro e, por fim, a exportação do arquivo de texto.

5.1. Telas da Ferramenta

A ferramenta desenvolvida possui uma tela principal, onde é possível selecionar a fase do censo no qual se desejar corrigir os dados. Possui telas secundárias que se remetem as tratativas dos dados, onde é possível efetuar o carregamento de dados de um arquivo de texto. A interface exibe todos os campos separados por categorias. As funcionalidades principais será a alteração dos dados e a exportação do arquivo em um repositório predefinido.

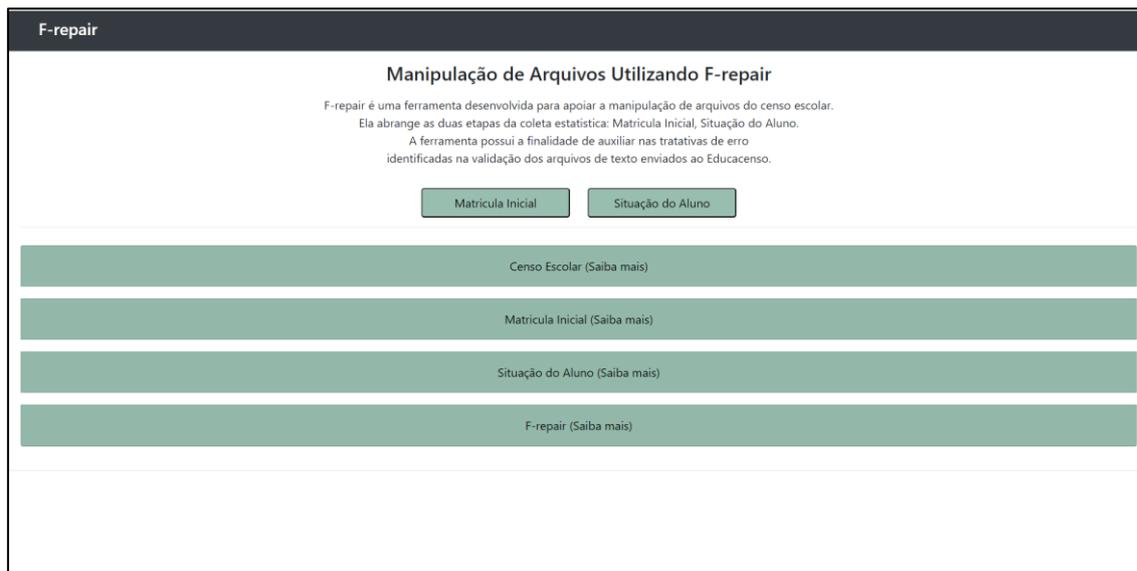
Esta seção apresenta e descreve as telas da ferramenta desenvolvida.

5.1.1. Tela Inicial

A Figura 9 apresenta a Tela Inicial, onde os usuários fazem acesso a ferramenta. Essa tela apresenta os seguintes itens:

- **Botão Matrícula Inicial:** Botão que direciona o usuário para a Tela Matrícula Inicial.
- **Botão Situação do Aluno:** Botão que direciona o usuário para a tela Situação do Aluno.
- **Botão Censo Escolar (Saiba mais):** Botão que exibe informações sobre o censo escolar.
- **Botão Matrícula Inicial (Saiba mais):** Botão que exibe informações sobre a 1ª fase do censo escolar.
- **Botão Situação do Aluno (Saiba mais):** Botão que exibe informações sobre a 2ª fase do censo escolar.
- **Botão F-Repar (Saiba mais):** Botão que exibe informações sobre a ferramenta e seu funcionamento.

Figura 9 - Tela Inicial



Fonte: Elaborado pelo autor.

5.1.2. Tela Matrícula Inicial

A Figura 10 apresenta a Tela Matrícula Inicial, onde os usuários fazem acesso a ferramenta. Essa tela apresenta os seguintes itens:

- **Código MEC:** Campo onde o usuário informa o código da unidade escolar que deseja pesquisar.
- **Botão Pesquisar:** Botão que busca as informações da escola e envia os dados para tabela
- **Botão Consultar Escola:** Botão que lista todas as unidades escolares disponíveis para correção.
- **Botão Exportar:** Botão que gera o arquivo de texto em um diretório fixo.
- **Componente Base de Navegação:** Esse componente faz a categorização dos registros (00, 10, 20, 30, 40, 50, 60) em abas.
- **Tabela:** Tabela onde é carregado os registros da unidade escolar pesquisada.

Figura 10 - Tela Matrícula Inicial

F-repair

1ª Etapa: Registros - Matrícula Inicial

Digite o código da unidade escolar.

Código MEC

Selecione o tipo de registro desejado.

REGISTRO 00

#	TIPO REGISTRO	CODIGO MEC	SITUAÇÃO FUNCIONAMENTO	DATA INICIO ANO	DATA FIM ANO	NOME ESCOLA	CEP	ENDEREÇO_COD_MUNICIPIO	ENDEREÇO_COD_DISTrito	ENDEREÇO	ENDEREÇO_NUMERO
	<input type="button" value="Filtrar"/>	<input type="button" value="1"/>	<input type="button" value="Filtrar"/>	<input type="button" value="Filtrar"/>	<input type="button" value="Filtrar"/>	<input type="button" value="Filtrar"/>					

Fonte: Elaborado pelo autor.

5.1.1.3. Tela Situação do Aluno

A Figura 11 apresenta a Tela Situação do Aluno, onde os usuários fazem acesso a ferramenta. Essa tela apresenta os seguintes itens:

- **Código MEC:** Campo onde o usuário informa o código da unidade escolar que deseja pesquisar.
- **Botão Pesquisar:** Botão que busca as informações da escola e envia os dados para tabela
- **Botão Consultar Escola:** Botão que lista todas as unidades escolares disponíveis para correção.
- **Botão Exportar:** Botão que gera o arquivo de texto em um diretório fixo.
- **Componente Base de Navegação:** Esse componente faz a categorização dos registros (89, 90, 91) em abas.
- **Tabela:** Tabela onde é carregado os registros da unidade escolar pesquisada.

Figura 11 - Tela Situação do Aluno

F-repair

2º Etapa: Registros - Situação do Aluno

Digite o código da unidade escolar.

Código MEC

Selecione o tipo de registro desejado.

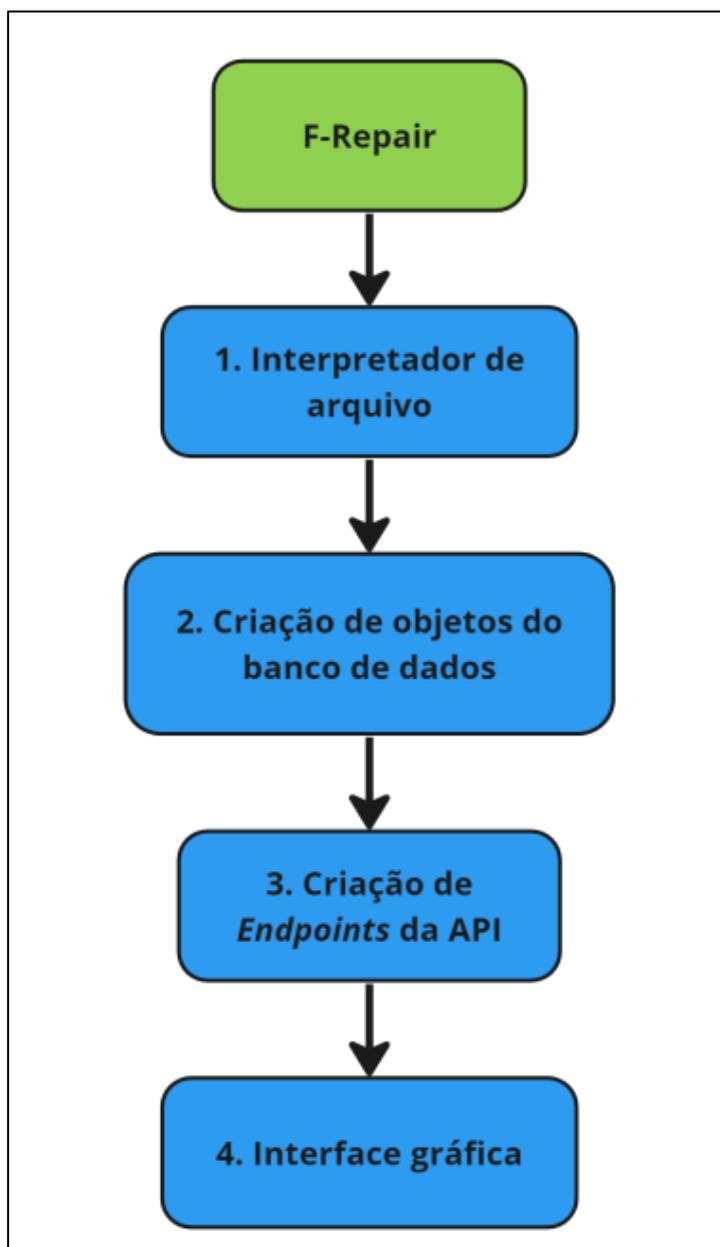
#	CODIGO MEC	CPF GESTOR	NOME GESTOR	CARGO GESTOR	EMAIL GESTOR
<input type="text"/>					

Fonte: Elaborado pelo autor.

5.2. Fluxo

A fim de organizar melhor as ideias e os conceitos teóricos descritos no projeto, foram criados fluxos e separado as principais etapas a serem desenvolvidas, conforme a Figura 12.

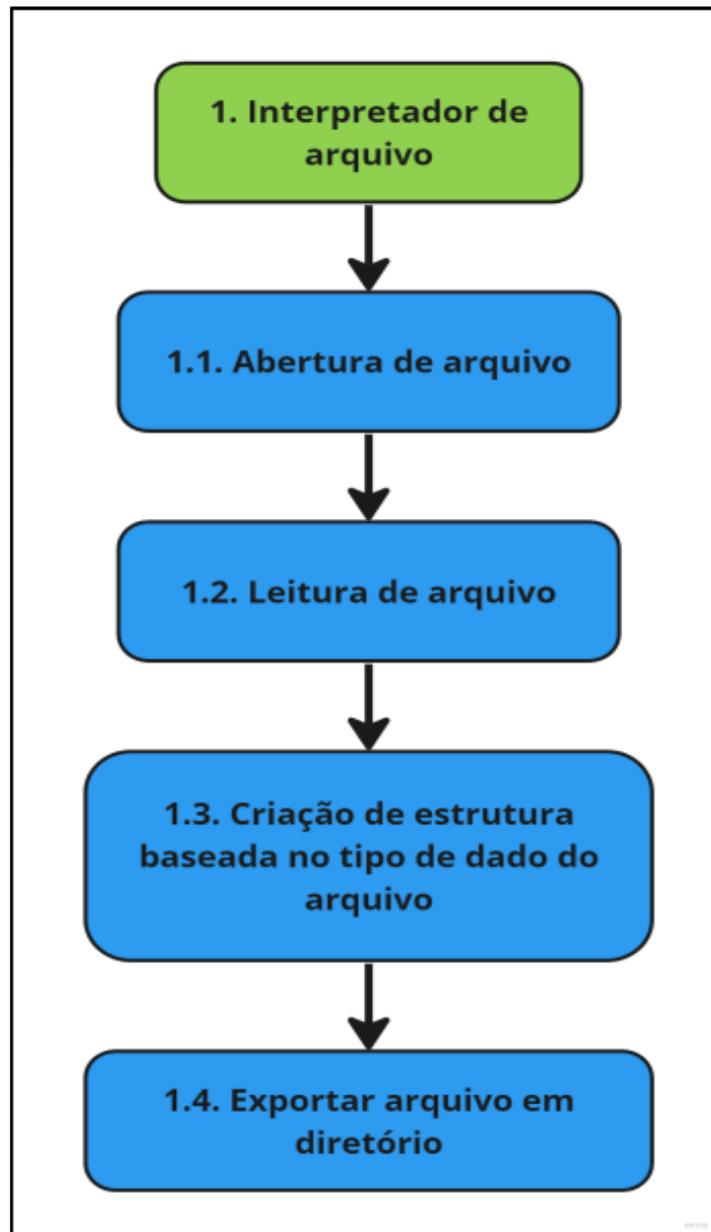
Figura 12 - Etapas do Projeto



Fonte: Elaborado pelo autor.

A primeira etapa do projeto foi concebida inicialmente como a seção de interpretação de arquivo de texto, conforme ilustrado na Figura 13. Nessa etapa, seria realizada a abertura e leitura do arquivo armazenado em um diretório, seguida pelo tratamento dos dados contidos no arquivo. No entanto, surgiram muitos erros, o que exigiu uma mudança de abordagem para o uso de um banco de dados.

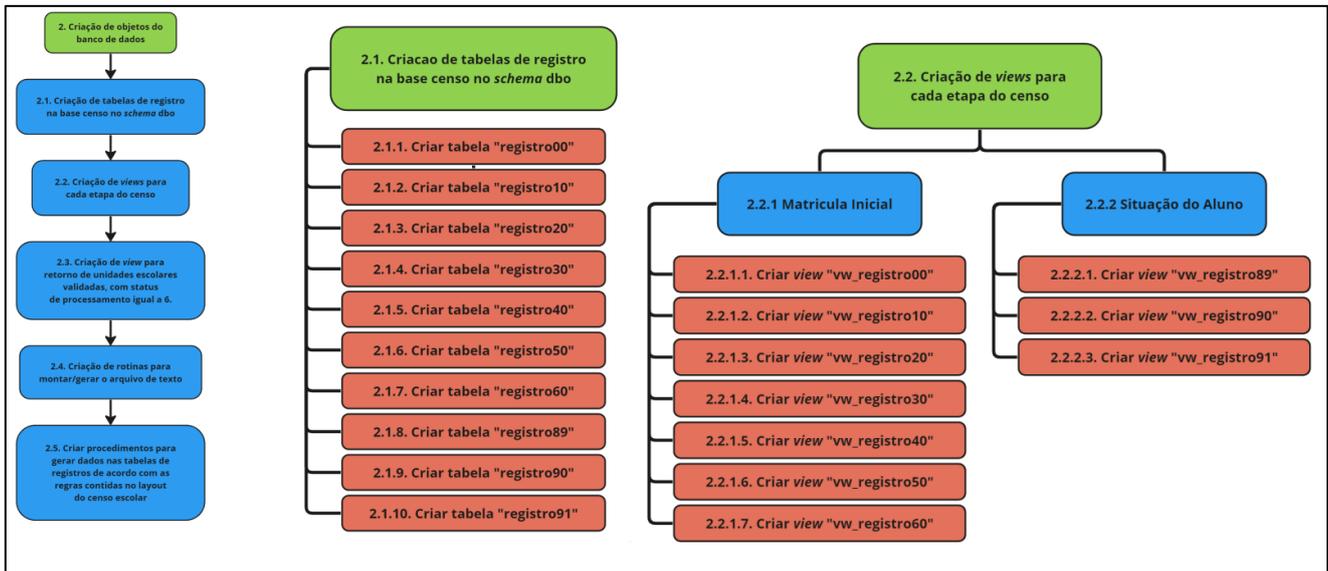
Figura 13 - Fluxo de Interpretação de Arquivos



Fonte: Elaborado pelo autor.

A partir dessa nova abordagem foi necessário criar objetos de banco como tabelas, views e procedimentos, para atender a essa nova perspectiva, conforme o fluxo exibido na Figura 14.

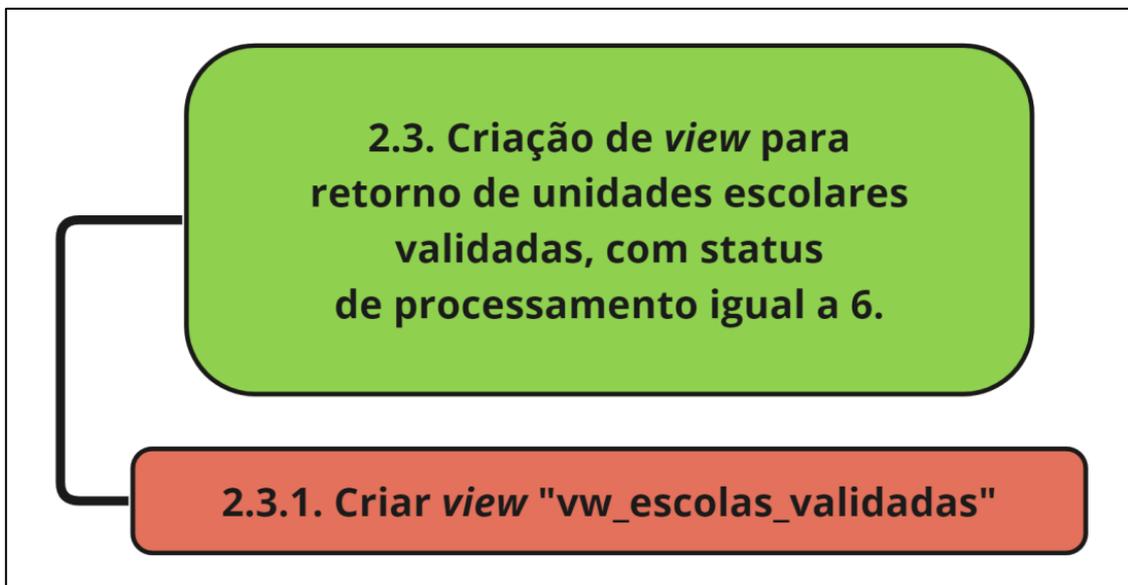
Figura 14 - Fluxo do Banco de Dados



Fonte: Elaborado pelo autor.

A Figura 15 exibe a view a ser utilizada para consulta de unidades escolares disponíveis para correção.

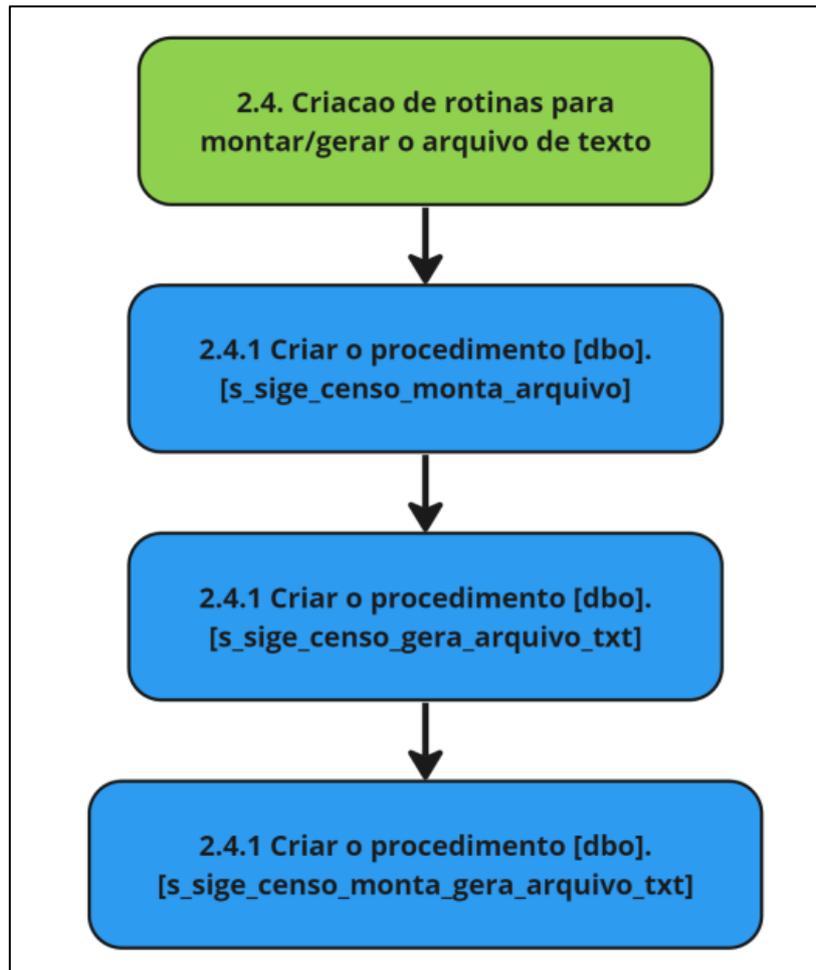
Figura 15 - Fluxo do Banco de Dados II



Fonte: Elaborado pelo autor.

A Figura 16 exibe os procedimentos que serão utilizadas para a montagem, geração e exportação do arquivo de texto.

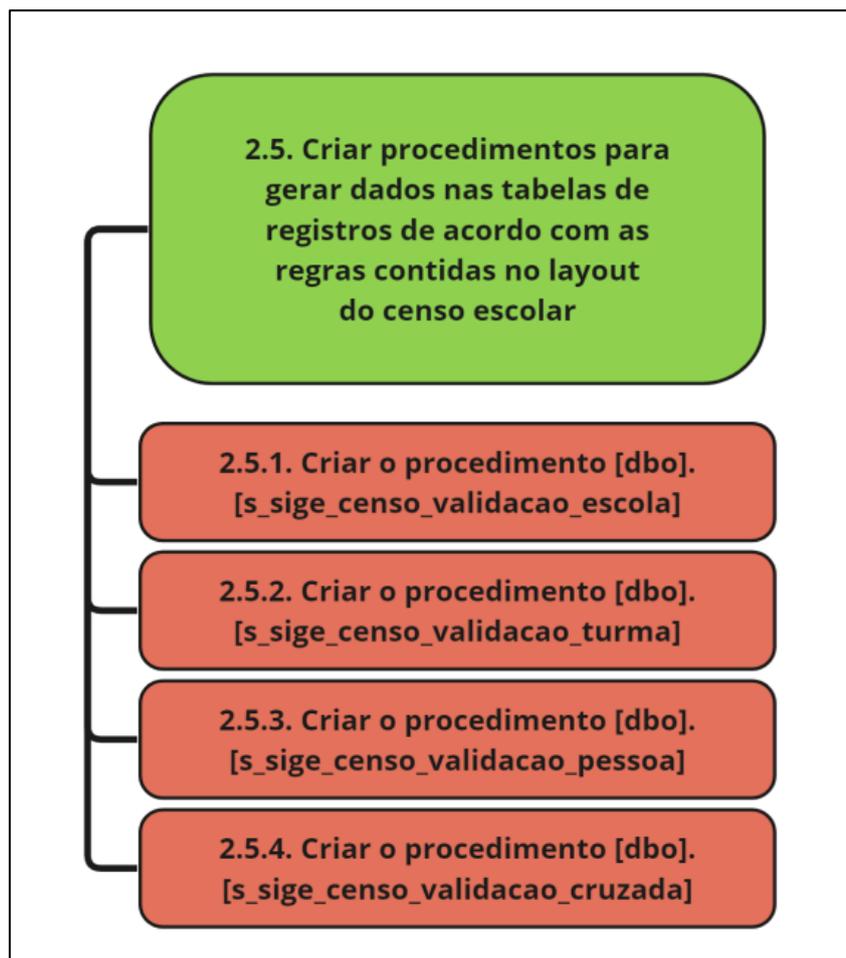
Figura 16 - Fluxo do Banco de Dados III



Fonte: Elaborado pelo autor.

A Figura 17 exibe os procedimentos utilizados para validação dos dados que irão formar as tabelas de registro.

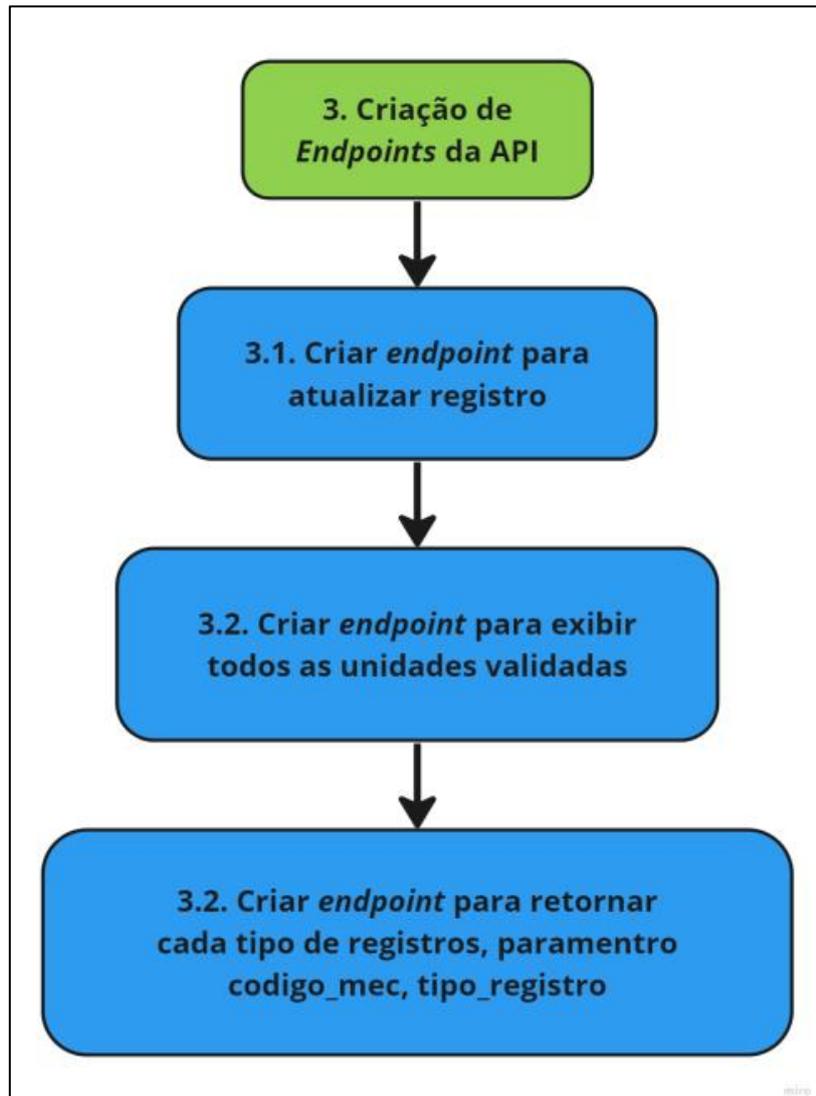
Figura 17 - Fluxo do Banco de Dados IV



Fonte: Elaborado pelo autor.

Com os objetos de banco criados, o próximo fluxo a seguir é o desenvolvimento da API e a implementação das rotas (*endpoints*) para cada funcionalidade da ferramenta, conforme o fluxo definido na Figura 18.

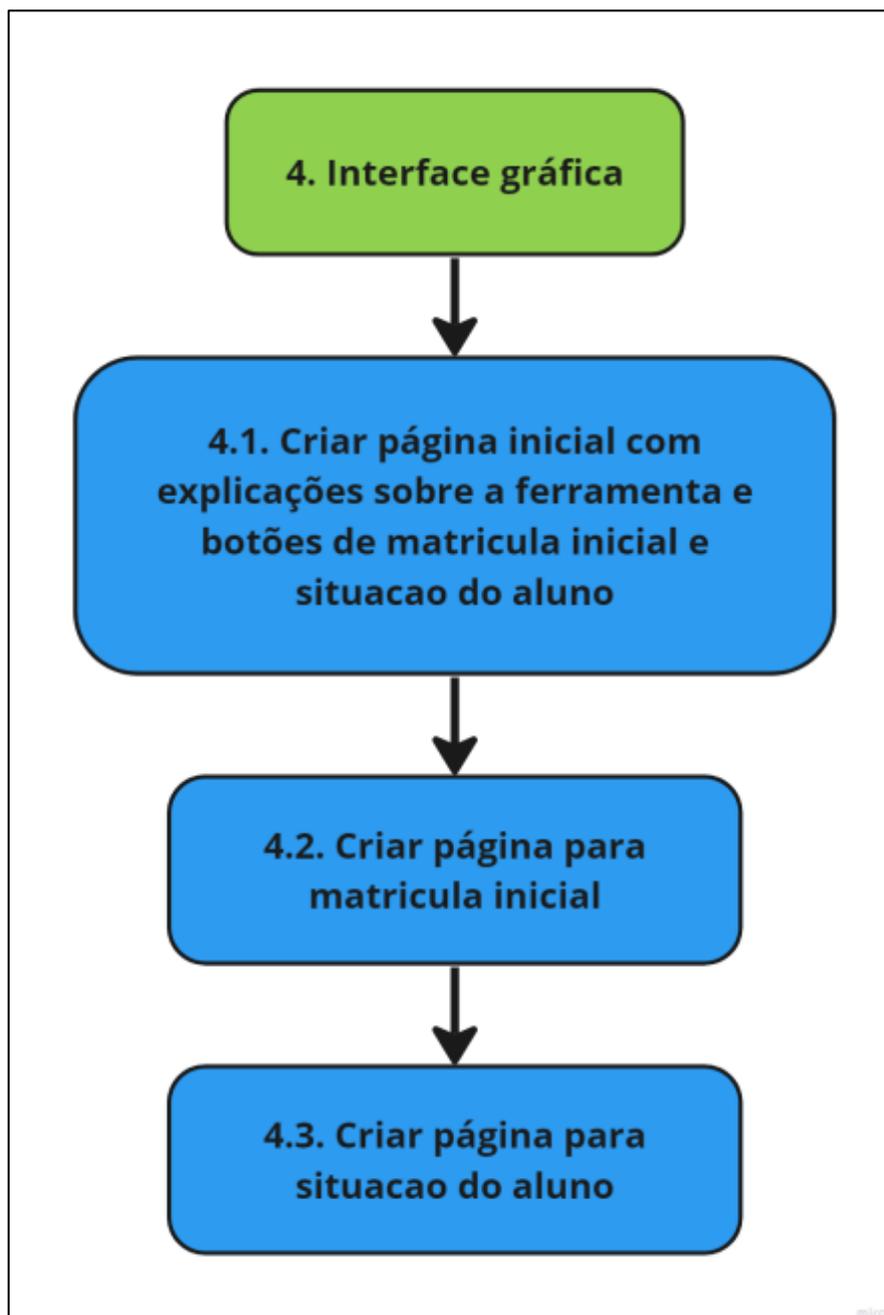
Figura 18 - Fluxo da API



Fonte: Elaborado pelo autor.

A Figura 19 exibe o fluxo da interface Web a ser criada, onde define as telas necessárias da ferramenta.

Figura 19 - Fluxo da Interface



Fonte: Elaborado pelo autor.

5.3. Implementação de API

Para que a aplicação funcione conforme o esperado é necessário o desenvolvimento de uma API, que contenha rotas para cada funcionalidade, O código exibe como se define a estrutura básica de uma API em Perl utilizando o *framework* Mojolicious.

```
1 use strict;
2 use warnings;
3 use Mojolicious::Lite;
4
5 plugin 'CORS'=>{origins=>'*'};
6
7 options '/api.censoproject.com/escolas-validadas' => sub {
8     my $c = shift;
9     $c->res->headers->header('Access-Control-Allow-Origin' => '*');
10    $c->res->headers->header('Access-Control-Allow-Methods' => 'GET, OPTIONS');
11    $c->res->headers->header('Access-Control-Allow-Headers' => 'Content-Type');
12    $c->render(text => 'OK!');
13 };
14
15 get '/api.censoproject.com/escolas-validadas' => sub {
16     my $c = shift;
17     $c->res->headers->header('Access-Control-Allow-Origin' => '*');
18
19     #----- Initial connection -----
20     my $resultados = consulta_v_bd();
21     #----- End connection -----
22
23     if () {
24         }
25         return $c->render(json => \@output);
26     } else {
27         my $no_result = ("message" => "Nenhum registro encontrado para a unidade escolar");
28     }
29 };
30
31 #-----
32 # Inicia a aplicação
33 app->start;
```

A primeira parte define uma rota `/api.censoproject.com/exemplo` e especifica um *callback* (sub-rotina) que será executado quando uma requisição for feita nessa rota. O *callback* recebe um objeto `$c` como argumento, que representa o contexto da requisição.

Dentro do *callback*, três cabeçalhos de resposta são definidos para permitir o acesso CORS (*Cross-Origin Resource Sharing*). O cabeçalho `'Access-Control-Allow-Origin'` é configurado com o valor `'*'`, permitindo que qualquer origem acesse a API. Os cabeçalhos `'Access-Control-Allow-Methods'` e `'Access-Control-Allow-Headers'` especificam os métodos HTTP e os cabeçalhos permitidos para a requisição `OPTIONS`. Em seguida, a resposta é renderizada com o texto `'OK'`.

A segunda parte do código define uma rota GET para a mesma URL '/api.censoproject.com/exemplo', logo em seguida fica bloco de código onde deve ser definido a lógica da aplicação a ser executado nessa rota, esse exemplo é de uma requisição do tipo GET, mais utilizada para busca de informações sem muitos parâmetros.

Por fim, a aplicação é iniciada com `app->start`, o que inicia o servidor web e permite que a API comece a receber requisições.

5.3.1. Endpoint Busca de Escolas Disponíveis

O código exibe a implementação de uma rota que possui como objetivo retornar os resultados de uma consulta ao banco de dados em formato JSON quando uma requisição GET for feita para a rota '/api.censoproject.com/escolas-validadas'.

```
1 # METODO GET ESCOLAS VALIDADAS
2
3 options '/api.censoproject.com/escolas-validadas' => sub {
4   my $c = shift;
5   $c->res->headers->header('Access-Control-Allow-Origin' => '*');
6   $c->res->headers->header('Access-Control-Allow-Methods' => 'GET, OPTIONS');
7   $c->res->headers->header('Access-Control-Allow-Headers' => 'Content-Type');
8   $c->render(text => 'OK');
9 };
10
11 get '/api.censoproject.com/escolas-validadas' => sub {
12   my $c = shift;
13   $c->res->headers->header('Access-Control-Allow-Origin' => '*');
14
15   #----- Initial connection -----
16   my $resultados = consulta_v_bd();
17   #----- End connection -----
18
19   if (defined $resultados) {
20     my @output;
21     foreach my $row (@$resultados) {
22       my %data;
23       foreach my $key (keys %$row) {
24         $data{$key} = $row->{$key};
25       }
26       push @output, \%data;
27     }
28     return $c->render(json => \@output);
29   } else {
30     my $no_result = ("message" => "Nenhum registro encontrado para a unidade escolar");
31   }
32 };
```

Esse *endpoint* visa buscar os dados das unidades escolares disponíveis e retornar o resultado após uma consulta ao banco de dados para isso é necessário obter o objeto de contexto da requisição `$c` através do comando *shift*.

Em seguida, é feito a realização de uma consulta ao banco de dados através da função `consulta_v_bd()` implementada. Após a consulta, é verificado se o resultado atribuído na variável `$resultados` está definido (ou seja, se houve

resultados encontrados). Se estiver definido, é criado um *array* @output para armazenar os dados retornados.

Em seguida, um loop percorre cada linha de resultado retornada e cria um *hash* %data com cada chave e valor da linha. Esse *hash* %data é adicionado ao *array* @output. Por fim, a resposta é renderizada em formato JSON com a função *render* do objeto de contexto \$c, utilizando o *array* @output como dados de saída.

```
1 sub consulta_v_bd {
2
3     my $dbh = conectar_bd();
4     $dbh->{'odbc_enable_utf8'} = 1;
5     $dbh->{'odbc_cursortype'} = 2;
6
7     my $sth;
8     my @resultados;
9     eval {
10         $sth = $dbh->prepare("SELECT id, codigo_mec, escola FROM censo.dbo.vw_escolas_validadas;");
11         $sth->execute();
12
13         while (my $row = $sth->fetchrow_hashref()) {
14             push @resultados, $row;
15         }
16     };
17     if ($?) {
18         # Tratamento de exceção
19         warn "Erro ao executar a consulta: $@";
20         return undef;
21     }
22
23     $sth->finish() if $sth;
24     $dbh->disconnect() if $dbh;
25
26     if (@resultados) {
27         return \@resultados;
28     } else {
29         return undef;
30     }
31 }
```

A sub-rotina acima chamada *consulta_v_bd* faz uma conexão com o banco de dados através da função *conectar_bd()*, em seguida dentro de um bloco *eval*, a consulta SQL é preparada com a função *prepare* usando a declaração "SELECT id, codigo_mec, escola FROM censo.dbo.vw_escolas_validadas;". Em seguida, a consulta é executada com a função *execute()*.

5.3.2. Endpoint de Pesquisa Escola Específica

O código exibe a implementação de uma rota que possui como objetivo retornar os resultados de uma consulta ao banco de dados em formato JSON quando uma requisição POST for feita para a rota '/api.censoproject.com/codigo_mec/registro'.

```

1 get '/api.censoproject.com/codigo_mec/:registro' => sub {
2   my $c = shift;
3   $c->res->headers->header('Access-Control-Allow-Origin' => '*');
4   my $codigo_mec = $c->param('codigo_mec');
5   my $registro = $c->param('registro');
6
7   #----- Initial validation -----
8   my $codigo_error = validar_codigo($codigo_mec);
9   if ($codigo_error) {
10    return $c->render(json => $codigo_error, status => 400);
11  }
12
13  my $registro_error = validar_registro($registro);
14  if ($registro_error) {
15    return $c->render(json => $registro_error, status => 400);
16  }
17  #----- End validation -----
18
19  #----- Initial connection -----
20  my $resultados = consulta_cr_bd($codigo_mec, $registro);
21  #----- End connection -----
22
23  if (defined $resultados && @$resultados) {
24    my @output;
25    foreach my $row (@$resultados) {
26      my %data;
27      foreach my $key (keys %$row) {
28        $data{$key} = $row->{$key};
29      }
30      push @output, \%data;
31    }
32    return $c->render(json => \@output);
33  } else {
34    my %no_result = ("message" => "Nenhum registro encontrado para a unidade escolar $codigo_mec");
35    return $c->render(json => \%no_result);
36  }
37 };

```

A variável `$data` é utilizada para armazenar o conteúdo da requisição, que é obtido através da função `$c->req->json`. A requisição contém um corpo em formato JSON, e a função `json` analisa esse JSON e retorna uma referência para uma estrutura de dados Perl. Os valores são extraídos das chaves 'codigo_mec' e 'registro' do JSON recebido e os armazena nas variáveis `$codigo_mec` e `$registro`, respectivamente.

Em seguida, há uma seção de validação inicial dos dados recebidos, pois foi criada uma função para validar a quantidade de caracteres do código Mec e outra função para validar o tipo de registro. Dessa forma a função `validar_codigo` é chamada passando o valor de `$codigo_mec` como argumento. Se essa função retornar algum erro, a resposta é renderizada em formato JSON com o conteúdo do erro e o status HTTP 400 (*Bad Request*) usando a função `render` do objeto de contexto `$c`. A mesma lógica é aplicada para a validação do `$registro`.

Após a validação inicial, é feita a conexão com o banco de dados, utilizando a função `consulta_cr_bd()`, que possui a mesma lógica da função do código anterior, é passado os valores de `$codigo_mec` e `$registro` como argumentos. E essa função realiza a consulta ao banco de dados usando esses valores e retornando o resultado.

5.4. Implementação da Interface Gráfica

Para o desenvolvimento da interface web, foi utilizado o *framework* Bootstrap. Dessa forma, foi possível aproveitar uma variedade de componentes pré-prontos, como menus de navegação, barras de progresso, botões, formulários e vários outros elementos, todos com estilos consistentes e responsivos. Além disso, o Bootstrap oferece um sistema de grid flexível, o que facilita a criação de layouts adaptáveis e compatíveis.

O corpo da página inclui um cabeçalho de navegação fixo no topo, com o título "F-repair". Logo abaixo, há um cartão com uma breve descrição da ferramenta "F-repair", explicando seu propósito e funcionalidades. Também são apresentados dois botões que levam às páginas "Matrícula Inicial" e "Situação do Aluno", respectivamente.

Em seguida, há um conjunto de cartões que funcionam como acordeões. Cada cartão apresenta informações sobre diferentes aspectos do "Censo Escolar", como uma breve descrição do censo, informações sobre a "Matrícula Inicial" e a "Situação do Aluno". Cada cartão pode ser recolhido ou expandido para exibir mais detalhes.

Por fim, há um último cartão que descreve a ferramenta "F-repair" em mais detalhes, explicando como ela funciona e as opções disponíveis. A ferramenta permite a pesquisa e edição de dados das unidades escolares, separando-os em diferentes tipos de registro com base na etapa escolhida. Também é possível exportar o arquivo de texto corrigido após as alterações serem concluídas.

5.5. Implementação alternativa da ferramenta utilizando a linguagem C#

Como alternativa, durante o processo de desenvolvimento, foi implementada uma aplicação utilizando a linguagem C#, conforme mostra a Figura 20.

Essa nova abordagem apresenta funcionalidades semelhantes à implementação web, como acesso aos dados de cada tipo de registro por meio de botões específicos, possibilidade de alteração de dados, contagem de registros e exportação de arquivo de texto para um diretório definido.

O *back-end* da ferramenta foi desenvolvido em C#, enquanto para o *front-end* foi utilizada a interface Windows *Forms*, que faz parte do framework .NET e oferece um conjunto de controles e componentes predefinidos que podem ser arrastados e soltados em uma janela de aplicativo para criar interfaces interativas.

A ferramenta foi implementada é uma aplicação desktop, implementada como alternativa para o caso da ausência de conexão com a internet.

Figura 20 - Tela F-repair C#

The screenshot shows the F-Repair C# application window. At the top, there is a menu bar with 'Arquivo', 'Editar', and 'Ajuda'. Below the menu is a search bar containing '52020436'. The main area is a table with the following columns: 'Registro OO', 'codigo', 'codigo_mec', 'codigo_turna', 'cod_turna_inep', 'inep_aluno', 'matricula', 'matricula2', 'codigo_modalidade', 'codigo_etapa', 'movimento', 'rendimento', 'concluinte', 'andamento', and 'situacaoAlunc'. The table contains 35 rows of data. On the left side, there is a sidebar with buttons for 'Registro 00', 'Registro 10', 'Registro 20', 'Registro 30', 'Registro 40', 'Registro 50', 'Registro 60', 'Registro 89', 'Registro 90', and 'Registro 91'. At the bottom left, there is an 'Exportar TXT' button. The status bar at the bottom indicates 'Count: [272]'.

Registro OO	codigo	codigo_mec	codigo_turna	cod_turna_inep	inep_aluno	matricula	matricula2	codigo_modalidade	codigo_etapa	movimento	rendimento	concluinte	andamento	situacaoAlunc
556805	52020436	20221	27168721	119125619054	21120548560	564932926	1	21	1					5
556806	52020436	20221	27168721	119400336160	13101462793	564932699	1	21			1			5
556807	52020436	20221	27168721	120432591771	13101698009	537912494	1	21			1			5
556808	52020436	20221	27168721	120436965160	11001764303	537912403	1	21			0			4
556809	52020436	20221	27168721	121169772667	15106100515	537912772	1	21			1			5
556810	52020436	20221	27168721	122217257300	11002196363	537912385	1	21			1			5
556811	52020436	20221	27168721	122302274244	17111711423	564935147	1	21		1				1
556812	52020436	20221	27168721	122426244070	14103597350	537912557	1	21			1			5
556813	52020436	20221	27168721	122558612165	22121717901	537912746	1	21			1			5
556814	52020436	20221	27168721	122948523205	13101846831	537912519	1	21			1			5
556815	52020436	20221	27168721	123157907866	11120467822	537912737	1	21			1			5
556816	52020436	20221	27168721	123505957576	17110922365	537912657	1	21			1			5
556817	52020436	20221	27168721	123777628327	21119892015	537912773	1	21			1			5
556818	52020436	20221	27168721	123891461045	15105400364	537912597	1	21			1			5
556819	52020436	20221	27168721	124004920263	16108907178	537912628	1	21			1			5
556820	52020436	20221	27168721	124038101508	13101754290	564933374	1	21		1				1
556821	52020436	20221	27168721	124314898273	17111099207	537912661	1	21			1			5
556822	52020436	20221	27168721	124327422575	19115839549	537912700	1	21			1			5
556823	52020436	20221	27168721	125004196060	15105248292	537912591	1	21			1			5
556824	52020436	20221	27168721	125257975660	14103422018	537912546	1	21		1				1
556825	52020436	20221	27168721	125260402722	14103470542	564932800	1	21			1			5
556826	52020436	20221	27168721	125260466020	14103704406	537912562	1	21			1			5
556827	52020436	20221	27168721	125260484435	14103124086	537912543	1	21			1			5
556828	52020436	20221	27168721	125260627210	14103587110	537912556	1	21			1			5
556829	52020436	20221	27168721	127829796393	15105452346	537912603	1	21			1			5
556830	52020436	20221	27168721	127832976470	15105435143	537912601	1	21		1				1
556831	52020436	20221	27168721	127888787028	15105395887	537912595	1	21			1			5
556832	52020436	20221	27168721	127888814440	1510573056	537912607	1	21			0			4
556833	52020436	20221	27168721	128132978026	15105498045	537912610	1	21			1			5
556834	52020436	20221	27168721	205139294586	22121864380	537912755	1	21			1			5
556835	52020436	202210	27168722	121035415100	12001109554	537912436	1	25			1			5

Fonte: Elaborado pelo autor.

6. CONSIDERAÇÕES FINAIS

Este trabalho descreve o desenvolvimento de uma ferramenta chamada F-repair, que incorpora a linguagem de programação Perl em seu back-end, sendo utilizada para criar uma API, em conjunto com o framework Mojolicious. Além disso, na parte visual, foi criada uma aplicação Web utilizando recursos HTML, CSS, JavaScript e Bootstrap.

A aplicação desenvolvida F-repair, otimiza o processo de coleta de dados de unidades escolares ao simplificar a correção de erros em arquivos de texto do Censo Escolar. Através de uma interface web intuitiva, os dados da unidade escolar são apresentados de forma organizada, seguindo o layout fornecido pelo INEP. Isso facilita a identificação e correção dos erros em cada registro, tornando o processo mais eficiente em comparação ao uso anterior do bloco de notas. Com o uso da ferramenta F-repair, o tempo necessário para corrigir os erros nos arquivos de texto foi reduzido significativamente. A eliminação da contagem manual dos pipes para identificar a posição do registro a ser alterado resulta em uma economia considerável de tempo. Além disso, a ferramenta possui uma interface visual agradável e menos cansativa para os usuários.

Além das vantagens mencionadas, a ferramenta F-repair também oferece a opção de exportar os dados em um arquivo de texto formatado de acordo com as especificações do INEP. Isso simplifica o carregamento do arquivo na plataforma do Educacenso, garantindo uma conclusão adequada do processo de correção. A importância dessa ferramenta no tratamento de dados reportados como incorretos pelo Educacenso é evidente, pois o método atual demanda muito tempo e apresenta alto risco de erros.

A ferramenta desempenha um papel fundamental ao agilizar o processo de correção, aumentar a precisão dos resultados e reduzir a possibilidade de equívocos.

6.1. Recomendações para Trabalhos Futuros

Na sequência do presente trabalho surgiram alguns aspectos que se revelaram relevantes para uma melhoria na abordagem. Em seguida são

referidos sumariamente aqueles que poderão vir a ser objecto de futura implementação:

- Aprimoramento da ferramenta para usuários que não possuem amplo conhecimento sobre o processo de coleta;
- Criação de botão para definição de caminho para geração do arquivo de texto;
- Criação de página para edição de dados específicos e descrição sobre o que cada dado do registro se refere;
- Integração com APIs da Receita Federal e dos Correios para correção automática de erros em nomes, endereços, CEP e CPF, entre outros.

REFERÊNCIAS BIBLIOGRÁFICAS

BARRO, Bruna B. **Perl: Entenda o Que é e Para que Serve essa Linguagem de Programação**. [S. l.], 1 fev. 2023. Disponível em:

<https://www.hostinger.com.br/tutoriais/perl>. Acesso em: 30 mai. 2023.

CAMARGO, Orson. **Censo, a contagem da população**; Brasil Escola.

Disponível em: <https://brasilecola.uol.com.br/sociologia/censo-contagem-populacao.htm>. Acesso em: out. 2022.

COSTA, Janine. **APIs: Tudo sobre o que é API, exemplos e importância!**:

Descubra o que é uma API, para que serve, qual a importância e principais exemplos de aplicações para gerar integração e desenvolvimento!. [S. l.], 27 jan. 2022. Disponível em: <https://www.zenvia.com/blog/apis-entenda-o-que-sao-e-como-funcionam/>. Acesso em: 28 maio 2023.

DINIZ, Ednar. **O Censo Escolar**. Revista Brasileira de Estudos Pedagógicos, Brasília, v. 80, nº. 194 (1999), p. 156-163, jan/abr, 1999. Disponível em:

<http://rbep.inep.gov.br/ojs3/index.php/rbep/article/view/1263>

GIL, Antônio Carlos. **Como Elaborar Projetos de Pesquisa**. 6. ed. São Paulo: Editora Atlas Ltda., 2017.

GITHUB, GitHub. **Let's build from here**: The complete developer platform to build, scale, and deliver secure softwar. [S. l.], 2023. Disponível em:

<https://github.com/about>. Acesso em: 30 maio 2023.

INEP (Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira).

Censo Escolar. Disponível em: <https://www.gov.br/inep/pt-br/areas-de-atuacao/pesquisas-estatisticas-e-indicadores/censo-escolar>. Acesso em: out. 2022.

MARIANO, DIEGO CÉSAR BATISTA; de MELO-MINARDI, R. C. . **Introdução à Programação para Bioinformática com Perl**. 1. ed. North Charleston, SC (EUA): CreateSpace Independent Publishing Platform, 2016. v. 2. 200p.

MEC (Ministério da Educação). **Apresentação: Coleta de dados do Censo Escolar**. [S. l.], 2018. Disponível em: <http://portal.mec.gov.br/component/tags/tag/educacenso>. Acesso em: 10 out. 2022.

MICROSOFT, **Code editing. Redefined**: Free. Built on open source. Runs everywhere. [S. l.], 2023. Disponível em: <https://code.visualstudio.com/>. Acesso em: 30 maio 2023.

MICROSOFT, SQL SERVER. **SQL Server**: Descubra o poder de seus dados. [S. l.], 2022. Disponível em: <https://www.microsoft.com/pt-br/sql-server/>. Acesso em: 30 maio 2023.

RIEDEL, Sebastian. **Perl Mojolicious**: Documentation - The Mojolicious Guide to the Galaxy. [S. l.], 2008. Disponível em: <https://docs.mojolicious.org/>. Acesso em: 28 maio 2023.

SIMÕES, A.; CARVALHO, N. **Desenvolvimento de Aplicações em Perl com FreeLing 3**. Linguamática, v. 4, n. 2, p. 87-92, 4 Dez. 2012.

SUKRIEH, Alexis. **Perl Dancer**: Documentation. [S. l.], 17 jun. 2021. Disponível em: <https://perldancer.org/>. Acesso em: 28 maio 2023.

WALL, Larry. Sobre Perl: **O que é Perl? Características e História**. [S. l.], 2002. Disponível em: <https://www.perl.org>. Acesso em: 18 out. 2022.

WAZLAWICK, R. S. **Metodologia da Pesquisa para Ciência da Computação**. 2ª. ed. [S.l.]: Campus, 2014

HIETANIEMI, Jarkko. CPAN: Comprehensive Perl Archive Network. [S. l.], 1995. Disponível em: <https://www.cpan.org/>. Acesso em: 25 out. 2022.