INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIAS E TECNOLOGIA DO AMAZONAS

CAMPUS MANAUS CENTRO

Bruno Sevalho Wesen

DESENVOLVIMENTO DE UMA APLICAÇÃO PARA ANÁLISE DE DADOS SOCIOECONÔMICOS SOBRE COMPETITIVIDADE ENTRE CIDADES BRASILEIRAS

Manaus, Amazonas – Brasil
2025

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIAS E TECNOLOGIA DO AMAZONAS

CAMPUS MANAUS CENTRO

Bruno Sevalho Wesen

DESENVOLVIMENTO DE UMA APLICAÇÃO PARA ANÁLISE DE DADOS SOCIOECONÔMICOS SOBRE COMPETITIVIDADE ENTRE CIDADES BRASILEIRAS

Trabalho de Conclusão de Curso apresentado à banca examinadora Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistema do Instituto Federal de Educação, Ciências e Tecnologia do Amazonas – IFAM Campus Manaus - Centro, como requisito para o cumprimento da disciplina TCC II – Desenvolvimento de Software

Prof. Emmerson Santa Rita - Orientador

Fevereiro / 2025
Manaus, Am

SOLICITAÇÃO ELETRÔNICA nº 2942/2025 PROT/CMC (1649851) 23042.002662/2025-90 pg. 5

Biblioteca do IFAM – Campus Manaus Centro

W512d Wesen, Bruno Sevalho.
Desenvolvimento de uma aplicação para análise de dados socioeconômicos sobre competitividade entre cidades brasileiras / Bruno Sevalho Wesen. – Manaus, 2025.
90 p. : il. color.

Monografia (Tecnologia em Análise e Desenvolvimento de Sistema). – Instituto Federal de Educação, Ciência e Tecnologia do Amazonas, *Campus* Manaus Centro, 2025.
Orientador: Prof. Me. Emmerson Santa Rita.

1. Tecnologia urbana. 2. Cidades inteligentes. 3. Inovação urbana. 4. Infraestrutura digital. I. Santa Rita, Emmerson. (Orient.). II. Instituto Federal de Educação, Ciência e Tecnologia do Amazonas. III. Título.

CDD 005.3

Elaborada por Márcia Auzier CRB 11/597

MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO MÉDIA E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA - AM
DEPARTAMENTO ACADÊMICO DE INFORMAÇÃO E COMUNICAÇÃO
CURSO DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

TERMO DE APROVAÇÃO

A monografia, que tem como título: **Desenvolvimento de uma aplicação para análise de dados socioeconômicos sobre competitividade entre cidades brasileiras** foi submetida à defesa pública, sob a avaliação de banca examinadora, como parte dos requisitos necessários para a obtenção do título de graduação do curso superior de Tecnologia em Análise e Desenvolvimento de Sistemas.

AUTOR (A): BRUNO SEVALHO WESEN

Monografia aprovada em: 31 / 01 / 2025

Orientador (a):

Primeiro (a) examinador (a):

Segundo (a) examinador (a):

RESUMO

Este trabalho apresenta o desenvolvimento e a análise de um sistema voltado para a avaliação de dados socioeconômicos das cidades brasileiras, com o objetivo de auxiliar na compreensão das disparidades regionais e na formulação de políticas públicas mais eficientes. Foram coletados dados de fontes confiáveis, como IBGE, DATASUS e INEP, abrangendo indicadores de educação, saúde e renda, os quais foram tratados, organizados e analisados por meio de ferramentas como **Dask**, **Pandas** e **Cython**, que otimizaram o processamento de grandes volumes de informações.

O sistema foi estruturado em uma arquitetura em camadas, com um backend desenvolvido em **Flask** e compilado em **C** para melhorar o desempenho, e um frontend interativo e responsivo, implementado com **Tailwind CSS**, **Semantic UI** e bibliotecas de visualização como **Plotly**. Os resultados foram apresentados em gráficos dinâmicos que permitem explorar dados como a evolução do PIB per capita, a média salarial e indicadores de saúde, evidenciando disparidades regionais e padrões de desenvolvimento.

Entre as análises realizadas, destacam-se a redução no número de matrículas escolares após a pandemia, as diferenças crescentes na média salarial entre regiões e a prevalência de problemas de saúde específicos em cidades como Manaus. O sistema demonstrou ser uma ferramenta prática e eficaz para transformar dados complexos em ideias acessíveis, contribuindo para o planejamento urbano e a competitividade das cidades brasileiras. Conclui-se que a integração de dados socioeconômicos com tecnologias modernas pode promover uma gestão pública mais informada, equitativa e sustentável.

Palavras-chave: Cidades inteligentes, Tecnologia urbana, Inovação urbana, Infraestrutura digital, Sustentabilidade urbana, Governança participativa, Inclusão digital, Privacidade dos dados, Equidade urbana.

ABSTRACT

This paper presents the development and analysis of a system aimed at evaluating socioeconomic data from Brazilian cities, with the aim of helping to understand regional disparities and formulate more efficient public policies. Data were collected from reliable sources, such as IBGE, DATASUS and INEP, covering education, health and income indicators, which were processed, organized and analyzed using tools such as **Dask**, **Pandas** and **Cython**, which optimized the processing of large volumes of information.

The system was structured in a layered architecture, with a backend developed in **Flask** and compiled in **C** to improve performance, and an interactive and responsive frontend, implemented with **Tailwind CSS**, **Semantic UI** and visualization libraries such as **Plotly**. The results were presented in dynamic graphs that allow the exploration of data such as the evolution of GDP per capita, average wage and health indicators, highlighting regional disparities and development patterns.

The analyses carried out highlighted the reduction in the number of school enrollments after the pandemic, the growing differences in average salaries between regions, and the prevalence of specific health problems in cities such as Manaus. The system proved to be a practical and effective tool for transforming complex data into accessible insights, contributing to urban planning and the competitiveness of Brazilian cities. It is concluded that the integration of socioeconomic data with modern technologies can promote more informed, equitable, and sustainable public management.

Keywords: Smart cities, Urban technology, Urban innovation, Digital infrastructure, Urban sustainability, Participatory governance, Digital inclusion, Data privacy, Urban equity.

LISTA DE FIGURAS

SOLICITAÇÃO ELETRÔNICA nº 2942/2025 PROT/CMC (1649851) 23042.002662/2025-90 pg. 10

LISTA DE TABELAS

LISTA DE SIGLAS

JS – JavaScript

REST – Representational State Transfer

PNUD – Programa das Nações Unidas para o Desenvolvimento

PNAD – Pesquisa Nacional por Amostra de Domicílios

IA – Inteligência Artificial

ML – Machine Learning

TIC – Tecnologias de Informação e Comunicação

PIB – Produto Interno Bruto

IDH – Índice de Desenvolvimento Humano

ANN – Redes neurais artificiais

IBGE – Instituto Brasileiro de Geografia e Estatisticas

MAE – Erro médio absoluto

CSV – Comma Separated Values

API - Application Programming Interface

URL - Uniform Resource Locator

HTML – HyperText Markup Language

CSS – Cascading Style Sheets

CRUD – criar, ler, atualizar e deletar

PIP – Package installer for Python

APT – Advanced Packaging Tool

DOM – Document Object Model

EJA – Ensino Médio e Educação de Jovens e Adultos

INEP – Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira

Sumário

1 INTRODUÇÃO

No cenário contemporâneo de crescente urbanização, as cidades brasileiras enfrentam desafios significativos para melhorar sua competitividade e qualidade de vida. As cidades que conseguem gerir com eficácia seus recursos e dados socioeconômicos possuem maiores chances de promover o desenvolvimento sustentável, atrair investimentos e melhorar as condições de vida de seus habitantes. A análise de dados socioeconômicos emerge como uma ferramenta essencial nesse contexto, oferecendo boas percepções sobre áreas como saúde, educação, segurança e infraestrutura, que afetam diretamente a competitividade entre as cidades (Caragliu et al., 2011).

Estudos anteriores indicam que fatores como inovação, qualidade de vida e desenvolvimento urbano estão diretamente relacionados à competitividade urbana (Giffinger et al., 2007). Diante disso, este trabalho tem como objetivo analisar dados socioeconômicos das cidades brasileiras, utilizando técnicas avançadas de inteligência artificial e aprendizado de máquina, para identificar padrões e propor estratégias que possam melhorar a competitividade dessas cidades. A integração desses dados nos processos de planejamento urbano pode fornecer informações fundamentais para a formulação de políticas públicas mais eficazes e inclusivas.

Indicadores socioeconômicos, como os disponibilizados pelo Instituto Brasileiro de Geografia e Estatística (IBGE), permitem uma visão detalhada sobre aspectos críticos do desenvolvimento urbano, incluindo educação, saúde, renda e infraestrutura. No entanto, apesar da ampla disponibilidade desses dados, sua análise e aplicação estratégica muitas vezes não são exploradas de maneira a maximizar seus benefícios no planejamento urbano e na promoção da competitividade entre as cidades.

Este trabalho busca investigar como os dados socioeconômicos podem ser utilizados para impulsionar a competitividade das cidades brasileiras, reconhecendo que a análise e a interpretação desses dados desempenham um papel estratégico no planejamento e na gestão urbana. A competitividade entre as cidades não se limita à adoção de tecnologias modernas ou à atração de

investimentos, mas envolve também a promoção de um ambiente inclusivo, sustentável e resiliente para todos os seus habitantes.

Ao explorar as possíveis correlações entre os indicadores socioeconômicos e os fatores que impulsionam a competitividade urbana, este estudo visa oferecer subsídios valiosos para gestores públicos, urbanistas e demais tomadores de decisão. Através da aplicação de técnicas avançadas de análise de dados, espera-se contribuir para a construção de cidades mais equilibradas e inovadoras, capazes de enfrentar os desafios do século XXI e oferecer oportunidades para todos os seus habitantes.

1.1 PROBLEMATIZAÇÃO

Apesar do crescente interesse e investimento em iniciativas voltadas para o desenvolvimento de cidades inteligentes, observa-se uma lacuna significativa na compreensão das verdadeiras necessidades e desafios enfrentados pelas cidades brasileiras. Como destacado por Angelidou (2017), muitos projetos de cidades inteligentes falham ao ignorar as realidades socioeconômicas locais e as demandas específicas da população. Nesse contexto, surge a seguinte questão central: até que ponto a análise de índices socioeconômicos pode contribuir para a melhoria da competitividade entre as cidades brasileiras?

Essa problemática envolve desafios cruciais. Primeiramente, é necessário avaliar se os indicadores tradicionalmente utilizados, como os fornecidos pelo Instituto Brasileiro de Geografia e Estatística (IBGE), refletem adequadamente as demandas e aspirações das cidades no século XXI. Em um mundo cada vez mais digital e interconectado, fatores como inclusão digital, acesso à informação e participação cidadã, fundamentais para o desenvolvimento urbano sustentável, muitas vezes não estão contemplados nas métricas tradicionais (Caragliu et al., 2011).

Outro desafio é entender como os dados socioeconômicos podem ser integrados de forma estratégica nas políticas públicas e no planejamento urbano. Isso requer a coleta e análise criteriosa de indicadores, bem como a identificação de padrões e tendências que possam orientar investimentos em

SOLICITAÇÃO ELETRÔNICA nº 2942/2025 PROT/CMC (1649851) 23042.002662/2025-90 pg. 15

áreas prioritárias, como infraestrutura, educação, saúde e segurança (Batty et al., 2012). A aplicação estratégica desses dados pode ajudar gestores públicos a priorizar iniciativas que impactem positivamente na competitividade das cidades brasileiras.

Além disso, é fundamental investigar como as cidades inteligentes podem utilizar os dados de maneira contextualizada e centrada nas necessidades humanas. Como observado por Hollands (2008), a tecnologia por si só não é suficiente para garantir melhorias significativas na qualidade de vida; é essencial que os dados sejam interpretados e aplicados de forma a atender às especificidades locais e promover um desenvolvimento urbano mais equilibrado.

Diante desses desafios, este estudo busca explorar como a análise de índices socioeconômicos pode contribuir para impulsionar a competitividade das cidades brasileiras. Através da integração de dados nas estratégias de desenvolvimento urbano, espera-se identificar oportunidades de intervenção que promovam um desenvolvimento mais equitativo, sustentável e inclusivo, além de fornecer subsídios para a formulação de políticas públicas inovadoras, como sugerido por Giffinger et al. (2007).

1.2 JUSTIFICATIVA

O desenvolvimento das cidades brasileiras é um fator fundamental para o crescimento econômico do país. Contudo, as disparidades socioeconômicas entre as cidades brasileiras limitam seu potencial competitivo no cenário nacional e internacional. Como destacado por Angelidou (2017), a análise de dados socioeconômicos oferece uma abordagem mais precisa para entender as necessidades específicas de cada cidade, permitindo que gestores públicos tomem decisões mais informadas e direcionem investimentos de forma estratégica.

Com a crescente disponibilidade de dados e o avanço das tecnologias de análise, como machine learning e big data, há uma oportunidade sem precedentes de otimizar a alocação de recursos e melhorar a eficiência dos serviços públicos (Batty et al., 2012). Este estudo é justificado pela

necessidade de identificar as principais barreiras e oportunidades para a melhoria da competitividade das cidades brasileiras, com base na análise de dados que possam revelar padrões e correlações entre indicadores socioeconômicos. Ao aplicar ferramentas de análise avançada, espera-se contribuir para o desenvolvimento urbano sustentável e inclusivo das cidades do Brasil, promovendo a igualdade de oportunidades e a redução das disparidades regionais (Caragliu et al., 2011).

Ao integrar a análise de dados socioeconômicos nas estratégias de desenvolvimento das cidades brasileiras, é possível promover uma abordagem mais equitativa e inclusiva, que considere as necessidades e aspirações de todos os segmentos da sociedade. Segundo Angelidou (2017), a utilização de dados socioeconômicos pode fornecer uma base sólida para a formulação de políticas públicas mais eficazes, ao identificar áreas prioritárias de intervenção e orientar a alocação de recursos de forma estratégica. Além disso, Caragliu et al. (2011) ressaltam que o uso desses indicadores permite criar cidades mais competitivas, ao atrair investimentos e melhorar a qualidade de vida dos cidadãos.

Outra justificativa para este estudo reside na crescente disponibilidade de dados e nas tecnologias de análise avançada, como inteligência artificial e aprendizado de máquina, que tornam possível explorar de forma mais detalhada as relações entre indicadores socioeconômicos e a competitividade das cidades. Conforme apontado por Batty et al. (2012), a análise em larga escala de dados urbanos, utilizando técnicas de **machine learning**, pode identificar padrões e tendências que oferecem informações valiosas para gestores públicos, urbanistas e demais atores envolvidos no desenvolvimento urbano. Essas ferramentas tecnológicas permitem uma gestão mais eficiente e direcionada, facilitando a formulação de estratégias que promovam o desenvolvimento sustentável e a inclusão social.

1.3 OBJETIVOS

1.3.1 Objetivo Geral:

Criar um sistema para análise dos dados socioeconômicos. Investigar a relação entre os dados socioeconômicos e a competitividade das cidades, visando contribuir para uma compreensão mais abrangente e integrada do desenvolvimento urbano.

1.3.2 Objetivos Específicos:

- Coletar e integrar dados socioeconômicos de fontes confiáveis, como o Programa das Nações Unidas para o Desenvolvimento (PNUD) e instituições governamentais.
- Avaliar o estado atual da competitividade entre as cidades, considerando indicadores como educação, renda e saúde..
- Investigar como os dados socioeconômicos podem ser integrados às estratégias de desenvolvimento urbano e à promoção da competitividade das cidades inteligentes.

Esses objetivos direcionarão o desenvolvimento da pesquisa, permitindo uma abordagem sistemática e abrangente para investigar a relação entre os dados socioeconômicos e a competitividade dos municípios brasileiros, bem como para propor recomendações práticas para aprimorar o desenvolvimento urbano.

2 FUNDAMENTAÇÃO TEÓRICA

O propósito deste capítulo é estabelecer um embasamento teórico sólido para a construção de um sistema de análise de dados socioeconômicos das capitais brasileiras, explorando como as Tecnologias da Informação e Comunicação (TICs) e metodologias de ciência de dados podem ser aplicadas para monitoramento e avaliação da competitividade entre cidades inteligentes.

2.1 TEORIA DAS CIDADES INTELIGENTES E COMPETITIVIDADE URBANA

A teoria das cidades inteligentes tem sido amplamente discutida no campo do urbanismo e da tecnologia da informação, enfatizando o papel da digitalização na gestão e otimização dos serviços urbanos. Caragliu et al. (2011) definem uma cidade inteligente como aquela que utiliza TICs para potencializar a eficiência dos serviços públicos, melhorar a integração de dados e promover um desenvolvimento sustentável baseado em informações concretas. Nesse contexto, um sistema que centralize, analise e visualize indicadores urbanos pode fornecer uma visão estruturada e facilitar a tomada de decisões.

Angelidou (2017) destaca a importância da conectividade na gestão de cidades inteligentes, pois permite que os sistemas urbanos sejam monitorados em tempo real e que decisões sejam baseadas em análises preditivas. Isso justifica o uso de um sistema computacional capaz de coletar e processar grandes volumes de dados socioeconômicos para identificar padrões relevantes na competitividade entre cidades. O uso de bancos de dados eficientes, aliados a algoritmos de aprendizado de máquina e análise estatística, permite criar um ambiente digital de suporte à gestão urbana.

Além disso, Giffinger et al. (2007) propõem um modelo de competitividade baseado em seis dimensões principais: economia, mobilidade, meio ambiente, governança, qualidade de vida e capital humano. Esses fatores servem como base para estruturar os dados analisados pelo sistema, permitindo que diferentes áreas do desenvolvimento urbano sejam avaliadas em conjunto. Com a implementação de algoritmos avançados para análise de

séries temporais e predição de tendências, o sistema pode se tornar um mecanismo essencial para o planejamento estratégico das cidades.

Batty et al. (2012) reforçam a importância do uso de big data para gestão urbana, permitindo a criação de modelos que transformam dados brutos em informações úteis. Dessa forma, a aplicação proposta neste estudo visa integrar dados provenientes de diferentes fontes públicas e consolidá-los em uma plataforma interativa, fornecendo uma base robusta para a análise da competitividade das cidades brasileiras.

2.2 ÍNDICE DE DESENVOLVIMENTO HUMANO (IDH) E DESENVOLVIMENTO URBANO

O Índice de Desenvolvimento Humano (IDH) é amplamente utilizado para mensurar o progresso socioeconômico das cidades, combinando indicadores de educação, saúde e renda. Desenvolvido pelo Programa das Nações Unidas para o Desenvolvimento (PNUD) em 1990, ele oferece um panorama mais abrangente da qualidade de vida nas regiões analisadas (PNUD, 1990).

Para fins de implementação no sistema, o IDH serve como um dos principais insumos para a modelagem de competitividade urbana, permitindo que diferentes municípios sejam comparados e avaliados conforme sua evolução ao longo do tempo. A estrutura do sistema deve permitir a importação, tratamento e análise desses dados de forma automatizada, garantindo que a visualização de tendências seja acessível e interativa.

A relação entre IDH e desenvolvimento urbano pode ser explorada por meio de três componentes principais:

- Educação: O sistema analisará taxas de alfabetização e escolarização, correlacionando esses dados com a competitividade urbana. Silva e Cardoso (2012) indicam que um maior nível educacional está diretamente relacionado ao crescimento sustentável das cidades.
- Saúde: Indicadores como expectativa de vida e mortalidade infantil serão processados pelo sistema para avaliar o impacto dos serviços de saúde na competitividade das cidades (Sachs, 2015).

- Renda: O cruzamento de dados econômicos, como PIB per capita e média salarial, permitirá identificar padrões de crescimento econômico e desigualdade entre as cidades analisadas (Diener e Seligman, 2004).

Com a implementação de técnicas de análise exploratória e preditiva, o sistema pode evidenciar padrões que auxiliem na formulação de políticas públicas. Além disso, a integração de um backend otimizado para manipulação de grandes volumes de dados, aliado a um frontend interativo, possibilita que gestores e pesquisadores acessem as informações de maneira intuitiva, favorecendo a aplicabilidade prática do sistema para análises comparativas entre diferentes capitais brasileiras.

2.3 ANÁLISE DE DADOS E INTELIGÊNCIA ARTIFICIAL

As técnicas de análise de dados, **inteligência artificial** (**IA**) e aprendizado de máquina (**Machine Learning - ML**) desempenham um papel crucial na compreensão dos padrões e tendências dentro dos dados socioeconômicos, como o Índice de Desenvolvimento Humano (IDH). Estas técnicas possibilitam a modelagem preditiva, identificação de padrões ocultos e automação de processos analíticos que são essenciais para melhorar a competitividade urbana das cidades inteligentes.

Modelos de Regressão Linear:

O modelo de regressão linear simples é um dos métodos mais fundamentais para modelagem preditiva. Segundo Draper e Smith (1998), a regressão linear estabelece uma relação linear entre uma variável dependente *Y* (como o IDH) e uma ou mais variáveis indepen dentes *X* (como indicadores de saúde, educação e renda). A equação do modelo é dada por:

$$Y = \beta 0 + \beta 1X1 + \beta 2X2 + ... + \beta nXn + \epsilon$$

Onde $\beta 0$ é o intercepto, $\beta 1,\beta 2,...,\beta n$ são os coeficientes das variáveis independentes e ϵ é o termo de erro. Este modelo é utilizado para prever o IDH com base em fatores socioeconômicos e ajuda a identificar quais variáveis têm o maior impacto no desenvolvimento humano.

Árvores de Decisão

As árvores de decisão são um método de aprendizado supervisionado que pode ser aplicado tanto para classificação quanto para regressão. No contexto da análise do IDH, as árvores de decisão podem ajudar a modelar o impacto de diferentes políticas públicas no desenvolvimento humano. O algoritmo ID3, introduzido por Quinlan (1986), utiliza a entropia e o ganho de informação para selecionar os atributos mais significativos na divisão dos nós da árvore:

Entropia: Mede o grau de incerteza ou impureza dos dados:

$$H(S) = -\sum_{i=1}^{n} pilog2pi$$

Onde *pi* é a probabilidade de ocorrência da classe *i* no conjunto de dados *S*.

Ganho de Informação: Calcula a redução na entropia após a divisão de um conjunto de dados:

$$IG(S, A) = H(S) - \sum_{v \in Values(A)} \frac{|Sv|}{|S|} H(Sv)$$

Onde *Sv* é o subconjunto de *S* dividido com base no atributo *A*.

Modelos de *Clustering* (Agrupamento):

Os algoritmos de ***clustering***, como **k-means** e DBSCAN, são utilizados para segmentar cidades em grupos com características socioeconômicas semelhantes. De acordo com Xu e Wunsch (2005), o algoritmo **k-means** minimiza a soma das distâncias quadradas entre os pontos de dados e os centróides dos **clusters**:

$$J = \sum_{i=1}^{k} \sum_{x \in Ci} \vee x - \mu i \vee^{2}$$

Onde *Ci* são os **clusters** e *μi* os centróides. Isso permite identificar padrões entre cidades que compartilham níveis similares de IDH, ajudando os gestores a planejar intervenções direcionadas.

Redes Neurais Artificiais

Redes neurais artificiais (RNAs) são modelos inspirados no cérebro humano e são compostos por camadas de neurônios artificiais. No contexto do IDH e Segundo Hecht-Nielsen (1992), as redes neurais podem ser usadas para modelar relações não lineares complexas entre os indicadores socioeconômicos e o desenvolvimento humano. O algoritmo de retropropagação é frequentemente utilizado para treinar essas redes, ajustando os pesos das conexões para minimizar o erro da previsão:

$$\Delta wij = -\eta \frac{\partial E}{\partial wij}$$

Onde *Δwij* é o ajuste do peso entre os neurônios *i* e *j*, *η* é a taxa de aprendizado, e *E* é a função de erro.

Aplicações Práticas

O uso de ***machine learning*** e análise de dados permite não apenas a previsão do IDH, mas também Batty et al. (2012) examinam como técnicas de **machine learning**, como redes neurais artificiais, podem ser utilizadas para prever indicadores socioeconômicos, incluindo o IDH, e simular cenários de políticas públicas. Por exemplo, ao usar redes neurais artificiais, pode-se prever como o aumento de 10% nos investimentos em saúde afetará a expectativa de vida e, consequentemente, o IDH.

2.4 TRABALHOS CORRELATOS

A seguir, são apresentados alguns trabalhos e estudos correlatos que abordam temas similares ao proposto neste projeto, focando na análise de dados urbanos, competitividade de cidades inteligentes e o uso de indicadores como o IDH.

***Smart Cities in Europe: Analyzing the Competitiveness of Cities* (Caragliu et al., 2011)**

Este estudo explora como diferentes cidades europeias adotam tecnologias inteligentes para melhorar sua competitividade. Caragliu et al. (2011) utilizam um conjunto de dados composto por indicadores socioeconômicos, como educação, saúde, renda e infraestrutura tecnológica, para medir o progresso e a competitividade das cidades. A relevância deste trabalho reside no fato de que ele demonstra como o uso estratégico de Tecnologias de Informação e Comunicação (TICs) pode impactar positivamente a competitividade urbana, promovendo o crescimento econômico e a inclusão social.

Os dados utilizados no estudo foram extraídos de diversas bases públicas, incluindo o Eurostat, que oferece uma ampla gama de estatísticas socioeconômicas sobre a União Europeia. As conclusões indicam que cidades com altos investimentos em infraestrutura digital e educação tendem a ser mais competitivas e atraentes para empresas e talentos. Comparando com a realidade brasileira, esse estudo sugere que a análise de dados e o investimento em setores chave, como educação e saúde, podem melhorar a competitividade das cidades brasileiras de forma semelhante.

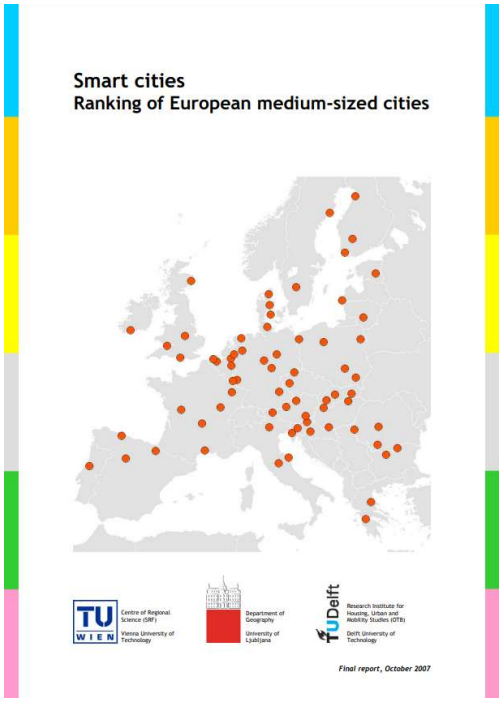
***Smart Cities: Ranking of European Medium-Sized Cities* (Giffinger et al., 2007)**

O trabalho de Giffinger et al. (2007) desenvolve um ranking de cidades inteligentes de médio porte na Europa, combinando indicadores sociais, econômicos e ambientais. Este estudo utiliza um conjunto de dados abrangente que inclui variáveis como taxa de desemprego, acesso à internet, educação, saúde e infraestrutura. A metodologia empregada envolve modelos estatísticos e técnicas de análise multivariada para identificar padrões no desenvolvimento urbano. Os resultados indicam que cidades com forte investimento em inovação e políticas públicas sustentáveis tendem a se destacar em competitividade.

Embora o estudo tenha como foco cidades europeias, sua metodologia e conclusões podem ser utilizadas como referência para avaliar a competitividade das cidades brasileiras. Ao adaptar os indicadores utilizados

para a realidade nacional, é possível identificar gargalos específicos, como desigualdade no acesso à infraestrutura e variações na qualidade dos serviços públicos. O modelo comparativo proposto por Giffinger et al. (2007) auxilia na identificação de padrões que favorecem o crescimento sustentável e o planejamento urbano eficiente, permitindo uma abordagem mais estruturada para compreender e aprimorar o desenvolvimento das cidades brasileiras.

Figura 1 – Capa de *Smart cities: Ranking of European medium-sized cities*



Fonte: (Giffinger et al, 2007)

***Smart City Policies: A Spatial Approach* (Angelidou, 2017)**

Este trabalho de Angelidou (2017) explora a interseção entre o Índice de Desenvolvimento Humano (IDH) e o desenvolvimento sustentável no contexto das cidades inteligentes. O estudo foca principalmente na relação entre políticas de educação e saúde e a sustentabilidade urbana, utilizando dados espaciais para entender as dinâmicas do desenvolvimento urbano. Angelidou argumenta que a competitividade urbana deve ser vista de uma forma holística, considerando tanto o desenvolvimento humano quanto a infraestrutura tecnológica.

Os conjuntos de dados utilizados incluem indicadores de desenvolvimento humano, extraídos do PNUD, além de dados sobre sustentabilidade e políticas públicas, fornecidos por órgãos governamentais. O estudo conclui que a adoção de políticas que incentivem o desenvolvimento humano, como a melhoria na educação e saúde, resulta em cidades mais competitivas e resilientes. Este trabalho é relevante porque ilustra como o IDH pode ser utilizado como uma métrica central para avaliar o progresso das cidades inteligentes brasileiras, além de demonstrar a importância de uma abordagem centrada no ser humano.

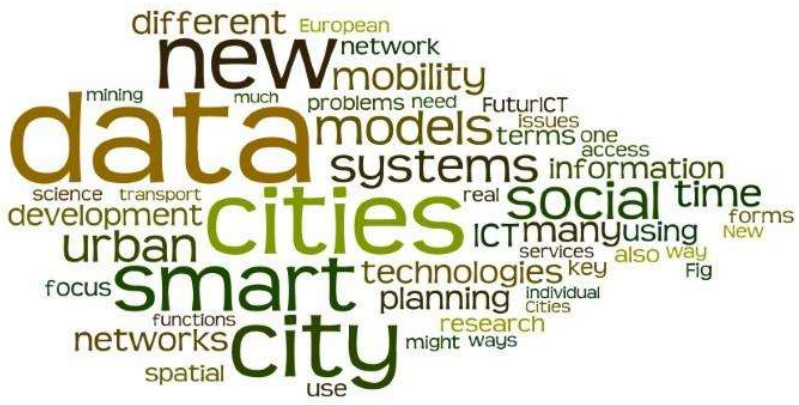
***Smart Cities of the Future: Predicting Socioeconomic Indicators with Machine Learning* (Batty et al., 2012)**

O estudo de Batty et al. (2012) examina como técnicas de **machine learning** podem ser aplicadas para prever indicadores socioeconômicos, incluindo o IDH, em cidades ao redor do mundo. Utilizando um extenso conjunto de dados extraídos de bases como o World Bank e o PNUD, este trabalho aplica algoritmos de aprendizado de máquina, como regressão linear e redes neurais, para prever o impacto de políticas públicas em áreas como educação, saúde e infraestrutura.

A relevância deste estudo está em sua aplicação prática de ferramentas de inteligência artificial para prever como mudanças nas políticas urbanas podem afetar o desenvolvimento humano e a competitividade das cidades. As conclusões indicam que modelos preditivos baseados em **machine learning** podem fornecer informações valiosas para gestores públicos ao permitir simulações de cenários futuros e a avaliação do impacto de políticas de longo prazo. Esse trabalho é diretamente aplicável ao contexto brasileiro, pois sugere que o uso de dados e tecnologias preditivas pode ser um diferencial para melhorar a competitividade das cidades.

Figura 2 – Resumo gráfico de *Smart cities of the future*

SOLICITAÇÃO ELETRÔNICA nº 2942/2025 PROT/CMC (1649851) 23042.002662/2025-90 pg. 26



Fonte: (Springer, 2012)

Comparações e Relevância para o Presente Estudo

Os trabalhos descritos acima compartilham a premissa de que a análise de dados socioeconômicos, como o IDH, é fundamental para entender e promover a competitividade urbana. Cada estudo oferece uma perspectiva única sobre como diferentes fatores, como educação, saúde e infraestrutura tecnológica, influenciam o desenvolvimento humano e o crescimento econômico das cidades.

No entanto, enquanto **Caragliu et al. (2011)** e **Giffinger et al. (2007)** focam mais na infraestrutura tecnológica e nas políticas públicas como motores de competitividade, **Angelidou (2017)** e **Batty et al. (2012)** destacam a importância de políticas centradas no ser humano e do uso de tecnologias preditivas, como **machine learning**, para antecipar os desafios urbanos e promover um desenvolvimento mais sustentável e equitativo.

Fatores de Competitividade dos Municípios Brasileiros: Proposta de Modelo de Análise (Canuto e Cherobin, 2021)

O estudo de Canuto e Cherobin (2021) propõe um modelo para analisar a competitividade dos municípios brasileiros a partir de múltiplos fatores socioeconômicos, incluindo infraestrutura, capital humano, inovação e governança. Os autores utilizam uma abordagem quantitativa baseada em

técnicas estatísticas para identificar as variáveis que mais influenciam o desempenho econômico e social dos municípios.

A pesquisa destaca que a competitividade municipal não se restringe apenas ao crescimento econômico, mas também envolve a capacidade de atrair investimentos, promover inovação e oferecer qualidade de vida à população. O modelo proposto pelos autores sugere que fatores como acesso à educação, qualidade da infraestrutura e eficiência da gestão pública desempenham papéis fundamentais na diferenciação entre municípios mais e menos competitivos.

Esse estudo contribui para o presente trabalho ao oferecer um referencial teórico e metodológico sobre como a análise de dados pode ser aplicada para avaliar a competitividade entre cidades. Além disso, reforça a importância da integração de diferentes indicadores socioeconômicos para uma compreensão mais ampla do desenvolvimento urbano. O modelo de análise proposto por Canuto e Cherobin pode ser útil para aprimorar a estrutura do sistema desenvolvido neste trabalho, permitindo uma comparação mais precisa entre as capitais brasileiras.

Diferenças em Relação aos Trabalhos Correlatos

Embora os trabalhos correlatos apresentem contribuições relevantes para o entendimento da competitividade urbana e das cidades inteligentes, este estudo se diferencia em diversos aspectos. Primeiramente, ao contrário de estudos como Giffinger et al. (2007) e Caragliu et al. (2011), que analisam cidades europeias utilizando rankings baseados em indicadores socioeconômicos, este trabalho foca exclusivamente nas ciades brasileiras, permitindo uma análise mais contextualizada da realidade nacional.

Além disso, enquanto pesquisas como Canuto e Cherobin (2021) propõem modelos teóricos para avaliar a competitividade dos municípios brasileiros, este trabalho desenvolve um sistema interativo que permite a visualização e análise dinâmica dos dados, integrando técnicas de **aprendizado de máquina** e **big data** para a identificação de padrões e tendências.

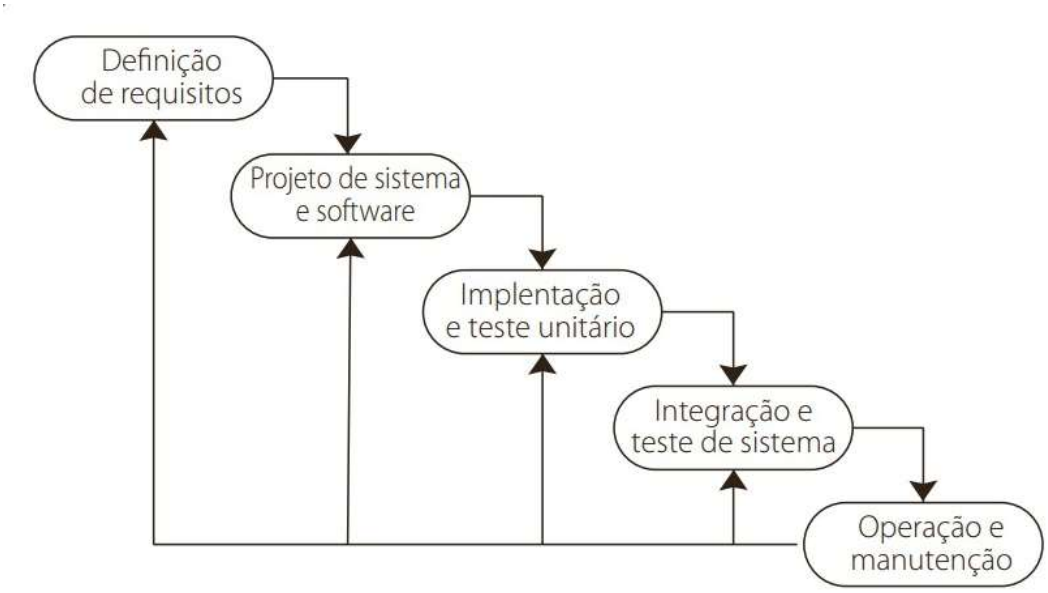
Outro diferencial significativo é o uso de um **backend** otimizado com **Cython** e **Dask**, permitindo a manipulação eficiente de grandes volumes de dados, algo não abordado nos trabalhos anteriores. Além disso, o **frontend** do sistema, desenvolvido com **Flask**, **Tailwind CSS** e **Semantic UI**, oferece uma interface amigável para que gestores públicos e pesquisadores possam explorar os dados de maneira interativa e tomar decisões mais embasadas.

Dessa forma, este trabalho não apenas contribui para a compreensão dos fatores que influenciam a competitividade entre as capitais brasileiras, mas também propõe uma ferramenta prática e aplicável para análise de cenários, tornando-se um diferencial em relação às abordagens predominantemente teóricas encontradas na literatura.

3 METODOLOGIA

O desenvolvimento deste trabalho segue o **Modelo Cascata**, conforme ilustrado na **Figura 3**, que organiza as etapas de construção do projeto em um fluxo sequencial estruturado. Este modelo é amplamente utilizado em projetos de engenharia de software, sendo adaptado aqui para estruturar o processo de análise de dados socioeconômicos das cidades brasileiras, com o objetivo de identificar padrões e propor estratégias para melhorar a competitividade entre as cidades.

Figura 3 – Modelo Cascata



Fonte: (Sommerville, 2011)

Etapas da Metodologia:

3.1 DEFINIÇÃO DE PROBLEMA E ESCOPO

Nesta etapa inicial, foram estabelecidos os objetivos gerais do trabalho e as diretrizes para a construção do sistema. O foco esteve na definição do escopo da análise, identificando quais **indicadores socioeconômicos** seriam relevantes para medir a competitividade entre as capitais brasileiras. Isso envolveu a seleção de fontes de dados confiáveis, como **IBGE, PNUD e bases**

governamentais abertas, bem como a delimitação das variáveis analisadas (educação, saúde, infraestrutura, renda, etc.). Além disso, foram identificadas as **necessidades funcionais** do sistema, garantindo que a análise pudesse ser conduzida de forma escalável e reproduzível.

3.2 ESTRUTURA DO SISTEMA

Nesta fase, foi projetada a arquitetura geral do sistema, contemplando o fluxo de dados e as etapas de processamento. Para isso, foi definido um **pipeline de dados**, estruturado em cinco etapas principais: **coleta, pré-processamento, análise exploratória, modelagem preditiva e visualização dos resultados**. O design do sistema priorizou a modularidade e a eficiência computacional, garantindo que os dados pudessem ser analisados de forma otimizada, independente do volume. Além disso, foi estabelecido um modelo de organização do código para facilitar futuras expansões e adaptações da solução.

3.3 IMPLEMENTAÇÃO E TESTE UNITÁRIO

A implementação do sistema foi conduzida utilizando linguagens e bibliotecas específicas para análise de dados e aprendizado de máquina. Os algoritmos foram testados individualmente (teste unitário) para garantir que cada componente do sistema (extração de dados, limpeza, modelagem e geração de relatórios) funcionasse conforme esperado.

3.4 INTEGRAÇÃO E TESTE DE SISTEMA

Após a validação dos componentes individuais, as diferentes partes do sistema foram integradas em um único fluxo. Nesta etapa, foi realizada uma análise detalhada do desempenho do pipeline como um todo, avaliando-se a eficácia das técnicas de análise para identificar padrões e correlações entre os índices socioeconômicos e a competitividade urbana. Métricas como acurácia, coeficiente de determinação (R^2) e erro médio absoluto (MAE) foram utilizadas para avaliar os modelos.

SOLICITAÇÃO ELETRÔNICA nº 2942/2025 PROT/CMC (1649851) 23042.002662/2025-90 pg. 31

3.5 OPERAÇÃO E MANUTENÇÃO

Com o sistema em operação, os resultados foram documentados e interpretados para fornecer pensamentos práticos. Além disso, foram definidos planos para a manutenção do sistema, incluindo a atualização de dados e a possibilidade de adaptação para novas variáveis ou contextos. A proposta inclui a entrega de um modelo replicável para gestores públicos e urbanistas, permitindo que as análises sejam facilmente atualizadas à medida que novos dados se tornem disponíveis.

SOLICITAÇÃO ELETRÔNICA nº 2942/2025 PROT/CMC (1649851) 23042.002662/2025-90 pg. 32

4 FERRAMENTAS UTILIZADAS

Foram utilizadas as seguintes ferramentas durante o desenvolvimento do estudo:

4.1 LINGUAGEM PYTHON

Python é uma linguagem de programação de alto nível, amplamente utilizada em análise de dados, aprendizado de máquina e desenvolvimento de sistemas devido à sua simplicidade, flexibilidade e vasta biblioteca de ferramentas (VAN ROSSUM, Guido; DRAKE, Fred L, 2001). Com uma sintaxe intuitiva, **Python** facilita o desenvolvimento rápido de soluções complexas, tornando-o ideal para projetos de pesquisa acadêmica e aplicações práticas.

No contexto deste trabalho, **Python** foi escolhido por sua robustez e ampla adoção na comunidade de ciência de dados. Diante disso a versatilidade do **Python** permitiu integrar todas as etapas do projeto, desde a coleta de dados até a modelagem preditiva e a geração de percepção, consolidando-se como uma ferramenta essencial para o sucesso deste trabalho.

4.2 PANDAS

Pandas é uma biblioteca de **Python** especializada na manipulação e análise de dados tabulares e estruturados (THE PANDAS DEVELOPMENT TEAM, 2012). É amplamente utilizada devido à sua eficiência e flexibilidade para lidar com grandes volumes de dados. A biblioteca fornece estruturas como **DataFrames**, que facilitam a organização, limpeza e transformação de dados, permitindo operações como filtragem, agregação, e fusão de tabelas de forma simples e intuitiva.

No contexto deste trabalho, o Pandas foi utilizado para:

- Importar dados de fontes como arquivos **CSV** ou bancos de dados.
- Realizar a limpeza e organização de índices socioeconômicos, incluindo tratamento de valores ausentes.
- Aplicar análises descritivas para identificar tendências e estatísticas básicas dos dados.

- Integrar dados de diferentes fontes para formar um conjunto coeso e pronto para análise.

A funcionalidade do **Pandas** em manipular dados estruturados foi essencial para preparar as informações necessárias para as etapas de modelagem e visualização.

4.3 NUMPY

NumPy é uma biblioteca fundamental para computação numérica em **Python**, fornecendo suporte para a criação e manipulação de **arrays multidimensionais**. Seu foco em eficiência permite a execução de operações matemáticas e estatísticas de alto desempenho, sendo amplamente utilizada em projetos que demandam cálculos intensivos (HARRIS, Charles R. et al, 2020).

No contexto deste trabalho, o **NumPy** foi utilizado para:

- Realizar cálculos matemáticos e estatísticos avançados, como normalização de dados e métricas de correlação.
- Manipular **arrays numéricos** grandes, otimizando o desempenho nas operações sobre os índices socioeconômicos.
- Servir como base para outras bibliotecas (como **Pandas** e **Scikit-learn**), garantindo que as operações fossem realizadas de forma eficiente.

4.4 SCIKIT-LEARN

O **Scikit-learn** é uma biblioteca **Python** amplamente utilizada em projetos de aprendizado de máquina, conhecida por sua simplicidade e eficiência na implementação de algoritmos (PEDREGOSA, Fabian et al, 2011). Ela oferece uma vasta gama de ferramentas para tarefas como classificação, regressão, **clustering** e redução de dimensionalidade, tornando-se indispensável para análises preditivas e exploratórias.

No contexto deste trabalho, o **Scikit-learn** foi utilizado para:

- **Pré-processamento de dados:** Aplicação de técnicas como normalização e padronização dos índices socioeconômicos para garantir que os algoritmos de aprendizado de máquina funcionem de forma ideal.
- **Modelagem preditiva:** Implementação de algoritmos como regressão linear para analisar relações entre variáveis e predizer com base nisso.

4.5 DASK

Dask é uma biblioteca Python projetada para computação paralela e processamento eficiente de dados em larga escala (DASK DEVELOPMENT TEAM, 2016). Ele estende as capacidades de bibliotecas como **Pandas**, **NumPy** e **Scikit-learn**, permitindo que as mesmas operações sejam realizadas em grandes volumes de dados que não cabem na memória, além de acelerar o processamento distribuindo tarefas entre múltiplos núcleos ou máquinas.

Principais Funcionalidades

1. Escalabilidade: **Dask** permite trabalhar com **datasets** maiores que a memória do sistema, dividindo-os em "**chunks**" (partições menores) que são processados de forma paralela ou sequencial, conforme os recursos disponíveis.
2. Integração com outras bibliotecas: Ele se integra com bibliotecas populares, como **Pandas**, **NumPy** e **Scikit-learn**, permitindo a migração de código existente para **Dask** com poucas modificações.
3. Execução paralela: Utiliza múltiplos **threads** ou **clusters** para distribuir a carga de trabalho, reduzindo significativamente o tempo de processamento de tarefas complexas.
4. Gráficos de tarefas dinâmicos: **Dask** constrói gráficos de tarefas dinâmicos que otimizam a execução das operações, garantindo que os recursos sejam utilizados de forma eficiente.

No contexto deste projeto, que analisa dados socioeconômicos das cidades brasileiras, o **Dask** foi utilizado para lidar com o grande volume de informações provenientes de fontes como IBGE e PNUD. Suas funcionalidades foram aplicadas nas seguintes etapas:

- Processamento de grandes volumes de dados: Dados socioeconômicos que envolvem múltiplas cidades e diversas variáveis, como educação, saúde e infraestrutura, foram carregados e processados de forma eficiente com **Dask DataFrames**, uma extensão do **Pandas** para grandes **datasets**.
- Agregação e análise paralela: O **Dask** permitiu realizar cálculos complexos, como a média ou correlação entre variáveis, utilizando múltiplos núcleos do processador, otimizando o tempo de execução.
- Pipeline escalável de aprendizado de máquina: Em combinação com o **Scikit-learn**, o **Dask** foi utilizado para aplicar algoritmos de aprendizado de máquina em grandes volumes de dados, distribuindo as tarefas e permitindo análises mais rápidas e eficazes.

Benefícios do **Dask**

- Redução do tempo de processamento para grandes **datasets**.
- Capacidade de manipular dados maiores do que a memória do sistema.
- Fácil integração com bibliotecas conhecidas e código já existente.
- Compatível tanto com máquinas individuais quanto com **clusters** distribuídos.

Dask foi uma ferramenta essencial para lidar com os desafios de escala e eficiência neste projeto, possibilitando análises mais rápidas e abrangentes dos dados socioeconômicos das cidades brasileiras.

4.6 JUPYTER

O **Jupyter Notebook** é uma ferramenta de código aberto amplamente utilizada para criação e compartilhamento de documentos interativos que integram código, texto explicativo, visualizações e equações matemáticas. Ele é muito popular na comunidade de ciência de dados devido à sua flexibilidade e facilidade de uso, permitindo que usuários combinem análise de dados e documentação em um único ambiente (KLINGE, Thomas et al, 2015).

No contexto deste projeto, que analisa os dados socioeconômicos das cidades brasileiras para identificar padrões e estratégias de competitividade urbana, o **Jupyter Notebook** foi utilizado como a principal interface para o

desenvolvimento das análises. Suas funcionalidades foram exploradas da seguinte forma:

- Exploração de dados: Foi utilizado para carregar, limpar e explorar os dados socioeconômicos provenientes de fontes como o IBGE e PNUD, oferecendo uma interface prática para visualizar amostras de dados e estatísticas descritivas.
- Iteração rápida com algoritmos: O **Jupyter** permitiu testar diferentes algoritmos de aprendizado de máquina (como regressão linear, **clustering** e árvores de decisão), ajustando os parâmetros em tempo real e observando os resultados imediatamente.
- **Visualização de ideias:** Gráficos interativos e mapas foram criados diretamente no **notebook**, facilitando a análise e interpretação dos resultados obtidos a partir dos dados das cidades brasileiras.
- **Documentação do processo:** O uso de texto em **Markdown** e explicações no mesmo ambiente de execução do código ajudou a organizar as análises, garantindo que cada etapa do processo fosse documentada de forma clara e compreensível.

4.7 LINUX

O **Linux** é um sistema operacional de código aberto amplamente utilizado em diversos campos, desde servidores até ambientes de desenvolvimento (TORVALDS, Linus; DIAMOND, David, 2001). Reconhecido por sua estabilidade, segurança e personalização, o **Linux** é uma escolha popular entre desenvolvedores, pesquisadores e profissionais que trabalham com análise de dados devido à sua capacidade de lidar eficientemente com tarefas complexas e grandes volumes de dados.

- **Código aberto e gratuito:** Como um sistema operacional de código aberto, o **Linux** oferece liberdade para personalização e uso sem custos de licenciamento, além de permitir que a comunidade contribua para melhorias constantes.

- **Estabilidade e desempenho:** O **Linux** é altamente confiável e otimizado para desempenho em tarefas intensivas, tornando-o ideal para análises de dados e execução de algoritmos computacionalmente exigentes.
- **Ambiente amigável para desenvolvimento:** Possui suporte nativo a linguagens como **Python**, **R**, **C**, **C++** e outras, além de ferramentas para automação e controle de versões, como **Bash** e **Git**.
- **Gerenciamento de recursos:** Sua arquitetura eficiente permite melhor uso de memória e **CPU**, especialmente em servidores e máquinas com múltiplos núcleos, otimizando o desempenho em tarefas paralelas.
- **Ecossistema robusto de pacotes:** Ferramentas como o gerenciador de pacotes **APT** (**Debian**/**Ubuntu**) ou **YUM** (**Fedora**) permitem instalar bibliotecas e softwares de análise de dados com facilidade.

No desenvolvimento deste projeto, o **Linux** foi utilizado como o ambiente operacional principal devido às suas vantagens em estabilidade e suporte para ferramentas de análise de dados e aprendizado de máquina. As principais aplicações incluem:

- **Execução de scripts e automação:** Scripts de processamento de dados e treinamento de modelos em **Python** foram executados de maneira eficiente no **Linux**, aproveitando o terminal **zsh** para automação de tarefas repetitivas, como limpeza de dados ou execução de pipelines.
- **Paralelismo e escalabilidade:** O suporte nativo do **Linux** para computação paralela foi utilizado em conjunto com bibliotecas como **Dask** otimizando operações em múltiplos núcleos.
- **Facilidade de instalação de ferramentas:** Ferramentas como **Jupyter Notebook**, **Pandas**, **Scikit-learn** e **Dask** foram configuradas rapidamente por meio de gerenciadores de pacotes como **pip** e **APT**, garantindo um ambiente de trabalho funcional e ágil.

4.8 CYTHON

SOLICITAÇÃO ELETRÔNICA nº 2942/2025 PROT/CMC (1649851) 23042.002662/2025-90 pg. 38

Cython é uma linguagem de programação e uma ferramenta de extensão do Python que combina a simplicidade e flexibilidade do Python com o desempenho do código nativo **C/C++** (BEHZAD, Ehsan et al,2023). Ele permite que desenvolvedores otimizem partes do código **Python**, traduzindo-o para **C/C++**, o que resulta em execuções significativamente mais rápidas.

Características Principais

1. Desempenho aprimorado: Ao compilar o código **Python** em código nativo **C**, o **Cython** pode acelerar a execução de funções intensivas em processamento, reduzindo significativamente o tempo de execução.
2. Compatibilidade com **Python**: **Cython** é totalmente compatível com **Python**, permitindo que desenvolvedores utilizem bibliotecas **Python** existentes, como **NumPy**, ao mesmo tempo em que otimizam partes críticas do código.
3. Interface com **C/C++**: Ele permite integrar facilmente bibliotecas ou funções escritas em **C** ou **C++**, expandindo as capacidades do **Python** para projetos que requerem alto desempenho.
4. Facilidade de uso: O código **Cython** é muito semelhante ao **Python**, com a adição opcional de anotações de tipos para maior otimização. Isso facilita a curva de aprendizado e o uso por desenvolvedores que já dominam **Python**.

No contexto deste projeto, o **Cython** foi usado para otimizar partes críticas do **pipeline** de análise de dados socioeconômicos das cidades brasileiras, especialmente em tarefas que envolviam grande volume de cálculos matemáticos e operações repetitivas. As principais aplicações incluem:

- **Aceleração de cálculos numéricos:** Funções matemáticas que manipulavam grandes matrizes e **arrays** de dados foram otimizadas com **Cython**, reduzindo o tempo de execução e permitindo a análise de dados em larga escala de maneira mais eficiente.

- **Integração com NumPy:** Ao combinar o poder do **Cython** com as estruturas de dados do **NumPy**, foi possível melhorar o desempenho de operações vetorizadas em índices socioeconômicos.
- **Desempenho em loops intensivos:** Laços iterativos, comuns em tarefas como cálculo de estatísticas avançadas ou execução de algoritmos de **clustering**, foram acelerados usando **Cython**, resultando em menor tempo de processamento.
- **Aumento de desempenho:** Código otimizado com **Cython** pode ser até 100 vezes mais rápido do que **Python puro**, dependendo da tarefa.
- **Integração nativa com Python:** Permite combinar o desempenho de **C/C++** com a facilidade de desenvolvimento de **Python**.
- **Flexibilidade:** Pode ser usado para otimizar apenas as partes do código que são críticas em termos de desempenho.
- **Código reutilizável:** Código escrito em **Cython** pode ser compilado e usado como módulos **Python** comuns.

4.9 FLASK

O **Flask** é um **microframework** de **Python** projetado para o desenvolvimento de aplicações **web**. Simples, leve e altamente flexível, o **Flask** permite criar desde protótipos rápidos até sistemas web robustos, oferecendo liberdade na escolha de bibliotecas e ferramentas adicionais (GRINBERG, Miguel, 2018). Por ser modular e extensível, ele é amplamente utilizado em projetos que integram análises de dados, como portais interativos e dashboards de gestão.

No desenvolvimento deste projeto, o **Flask** foi utilizado para criar o site de gestão dos dados socioeconômicos das cidades brasileiras, servindo como uma interface interativa para visualização, consulta e análise de informações. Suas funcionalidades foram exploradas da seguinte forma:

- **Frontend para análise de dados:** O Flask foi utilizado para construir um **frontend** dinâmico que permite aos usuários visualizar dados socioeconômicos em tabelas, gráficos e mapas interativos. Ferramentas

SOLICITAÇÃO ELETRÔNICA nº 2942/2025 PROT/CMC (1649851) 23042.002662/2025-90 pg. 40

como **Jinja2** (motor de **templates** do **Flask**) foram usadas para renderizar as páginas **web** dinamicamente com base nos dados.

- **Integração com o backend de análise de dados:** A aplicação foi integrada a bibliotecas como **Pandas** e **Scikit-learn** para processar os dados diretamente no servidor, que eram exibidos ao usuário em tempo real.
- **Roteamento para funcionalidades específicas:** Rotas foram implementadas para funcionalidades como:
    - **Upload** e processamento de novos **datasets**.
    - Consulta de dados específicos (ex.: índices de saúde, educação e infraestrutura por capital).
    - Geração de relatórios baseados nos resultados da análise.
- **Criação de APIs RESTful:** Uma **API** foi implementada para permitir que outros sistemas acessem os dados gerados pela aplicação, facilitando a integração com futuras ferramentas de gestão urbana.
- **Visualização de gráficos e dashboards:** Extensões como **Flask-Cors** e bibliotecas como **Plotly** e **Matplotlib** foram integradas para criar dashboards com visualizações ricas e interativas, permitindo aos gestores públicos explorar os dados de forma intuitiva.

4.10 TAILWIND CSS

O **Tailwind CSS** é um **framework** de utilitários para estilização que permite criar interfaces modernas e responsivas de maneira ágil e flexível. Diferente de **frameworks** tradicionais como **Bootstrap**, o **Tailwind** não fornece componentes pré-estilizados, mas sim uma coleção de classes utilitárias que podem ser combinadas para personalizar o design de forma granular. Ele é amplamente adotado devido à sua simplicidade, reutilização de código e suporte integrado à responsividade (TAILWIND LABS, 2023).

- **Baseado em classes utilitárias:** O **Tailwind CSS** utiliza classes pré-definidas que representam propriedades **CSS** específicas, como “text-

SOLICITAÇÃO ELETRÔNICA nº 2942/2025 PROT/CMC (1649851) 23042.002662/2025-90 pg. 41

center” (alinhamento de texto centralizado), “bg-blue-500” (fundo azul), ou “p-4” (padding de 1rem). Isso elimina a necessidade de escrever folhas de estilo extensas.

- **Responsividade integrada:** O **Tailwind** facilita a criação de layouts responsivos com **breakpoints** simples, como “sm”: (pequeno), “md”: (médio), “lg”: (grande), e “xl”: (extra grande), permitindo que os desenvolvedores adaptem o design para diferentes tamanhos de tela.
- **Comunidade e documentação rica:** A extensa documentação e a comunidade ativa facilitam o aprendizado e a resolução de problemas durante o desenvolvimento.

No desenvolvimento do site de gestão para este projeto, o **Tailwind CSS** foi utilizado para criar uma interface moderna, intuitiva e responsiva que facilita a interação dos usuários com os dados socioeconômicos das cidades brasileiras. Suas funcionalidades foram aplicadas da seguinte forma:

- **Estilização rápida de componentes:** As classes utilitárias do **Tailwind** permitiram estilizar botões, tabelas, gráficos e outros elementos do site de forma eficiente, garantindo uma aparência visualmente atraente e consistente.
- **Design responsivo: Breakpoints** do **Tailwind** foram usados para ajustar automaticamente o layout em dispositivos móveis e **desktops**, garantindo uma experiência de usuário fluida em qualquer resolução.

4.11 JAVASCRIPT

O **JavaScript** é uma linguagem de programação versátil e essencial para o desenvolvimento **web**, sendo amplamente utilizada para criar interfaces interativas e dinâmicas (FLANAGAN, David, 2020). Ele é executado diretamente no navegador, permitindo que os usuários interajam com elementos da página em tempo real.

SOLICITAÇÃO ELETRÔNICA nº 2942/2025 PROT/CMC (1649851) 23042.002662/2025-90 pg. 42

- **Execução no cliente:** O **JavaScript** é executado diretamente no navegador do usuário, reduzindo a carga no servidor e melhorando a experiência do usuário com interações rápidas.
- **Manipulação do DOM:** Ele permite acessar e modificar o **DOM** (Document Object Model), possibilitando atualizações dinâmicas de conteúdo sem recarregar a página.
- **Suporte para eventos:** O **JavaScript** gerencia eventos como cliques, pressionamento de teclas e movimentos do **mouse**, permitindo interações intuitivas.
- **Compatibilidade com APIs:** É usado para integrar **APIs REST**, **WebSockets** e outras interfaces que permitem a comunicação entre o **frontend** e o **backend**.

No desenvolvimento do site de gestão para os dados socioeconômicos das cidades brasileiras, o **JavaScript** foi utilizado para criar funcionalidades dinâmicas e interativas que melhoraram a experiência do usuário e facilitaram a visualização dos dados. Suas aplicações específicas incluem:

- **Atualização dinâmica de dados:** O **JavaScript** foi utilizado para exibir gráficos e tabelas que se atualizam em tempo real com os dados provenientes do **backend** desenvolvido em **Flask**. **APIs REST** foram integradas com funções **JavaScript** para buscar e exibir informações dinâmicas, como dados de educação, saúde e infraestrutura.
- **Validação de formulários:** O **JavaScript** foi aplicado para validar **inputs** de usuário em tempo real, garantindo que as informações inseridas fossem corretas antes de serem enviadas ao servidor.
- **Interação com o DOM:** Funções **JavaScript** foram utilizadas para manipular o **layout** da página, como exibir ou ocultar seções baseadas em cliques do usuário, criando uma navegação mais fluida e intuitiva.

4.12 SEMANTIC UI

O **Semantic UI** é um **framework CSS** que facilita a criação de interfaces modernas, responsivas e esteticamente agradáveis (Semantic UI, 2025).

Ele utiliza uma abordagem baseada em classes que seguem uma nomenclatura semântica, permitindo que o **HTML** seja mais legível e intuitivo. Por oferecer uma ampla gama de componentes pré-estilizados, o **Semantic UI** é uma escolha popular para o desenvolvimento de aplicações **web** rápidas e consistentes.

- **Nomenclatura semântica:** As classes do **Semantic UI** são intuitivas e próximas da linguagem natural tornando o código **HTML** mais legível e fácil de entender.
- **Amplo conjunto de componentes:** Inclui elementos prontos, como botões, tabelas, **cards**, **menus** e formulários, que podem ser personalizados para atender às necessidades do projeto.
- **Responsividade embutida:** Oferece suporte nativo para **layouts** responsivos, permitindo que as interfaces se ajustem automaticamente a diferentes tamanhos de tela.
- **Integração com JavaScript:** Inclui módulos integrados com **JavaScript** para criar elementos interativos, como **modais**, **dropdowns**, **tooltips** e **carrosséis**, com mínima configuração.

No desenvolvimento do site de gestão dos dados socioeconômicos das cidades brasileiras, o **Semantic UI** foi utilizado para criar uma interface visual consistente, funcional e intuitiva. Suas funcionalidades específicas incluem:

- **Criação de layouts responsivos:** A tabela dessa ferramenta foi usada para estruturar a página com colunas e linhas adaptáveis a diferentes dispositivos. Isso garantiu que a aplicação funcionasse bem tanto em desktops quanto em dispositivos móveis.
- **Elementos estilizados:** Componentes pré-construídos do **Semantic UI** aceleraram o desenvolvimento de botões, tabelas, **cards**, **menus** e outros.

4.13 PLOTLY

O **Plotly** é uma biblioteca interativa de visualização de dados para **Python, R, JavaScript e outras linguagens**. Amplamente utilizada em

ciência de dados e desenvolvimento **web**, ela permite criar gráficos de alta qualidade, interativos e responsivos (PLOTLY TECHNOLOGIES INC, 2015). É ideal para análises avançadas e **dashboards dinâmicos**, oferecendo suporte para uma ampla gama de visualizações, como gráficos de linha, barras, dispersão, mapas e visualizações 3D.

- **Interatividade integrada:** Os gráficos criados com **Plotly** possuem interatividade nativa, permitindo ações como **zoom**, **hover**, seleção de dados e atualização dinâmica.
- **Compatibilidade com várias linguagens:** Suporte para **Python** (**Plotly Express**), **R**, **JavaScript** (via **Plotly.js**) e **Julia**, facilitando sua integração em diferentes projetos e plataformas.
- **Personalização avançada:** Permite ajustar detalhes como cores, títulos, legendas, eixos e anotações, tornando os gráficos altamente configuráveis para atender às necessidades do projeto.
- **Suporte para gráficos complexos:** Oferece suporte a visualizações avançadas, como mapas geográficos, gráficos de rede, gráficos de calor e visualizações 3D, essenciais para dados multidimensionais.

No desenvolvimento deste projeto, o **Plotly** foi utilizado para criar visualizações interativas que destacam os padrões e tendências nos dados socioeconômicos das cidades brasileiras. Suas aplicações específicas incluem:

- **Criação de gráficos interativos:** Os gráficos de linha e barra foram usados para exibir séries temporais de indicadores, como variações de saúde, educação e infraestrutura ao longo dos anos.

- **Visualização de clusters e correlações:** Gráficos de dispersão interativos foram usados para ilustrar correlações entre variáveis, como a relação entre educação e competitividade urbana, com **clusters** destacados por região ou classificação.
- **Dashboards dinâmicos:** A integração com o **Flask** permitiu que os gráficos do **Plotly** fossem incorporados diretamente no site de gestão,

SOLICITAÇÃO ELETRÔNICA nº 2942/2025 PROT/CMC (1649851) 23042.002662/2025-90 pg. 45

possibilitando aos gestores públicos explorar os dados de forma interativa.

5 DESENVOLVIMENTO DO SISTEMA

5.1 ANÁLISE E ESPECIFICAÇÃO DE REQUISITOS

Foram definidos os seguintes requisitos funcionais e não funcionais para o sistema, que permitirá aos gestores públicos interagirem com um painel interativo para simulação e análise de cenários.

Requisitos Funcionais (RF)

1. **Seleção de Indicadores Personalizável:** O sistema deve permitir que os gestores públicos escolham múltiplos indicadores socioeconômicos (educação, saúde, renda, infraestrutura, entre outros) para análise comparativa. Além disso, deve possibilitar a seleção de faixas temporais para visualização da evolução histórica dos dados.

2. **Análises Estatísticas e Modelos Preditivos:** O sistema deve oferecer análises estatísticas detalhadas dos indicadores selecionados, permitindo a aplicação de técnicas de aprendizado de máquina para prever tendências futuras e gerar insights para a competitividade das cidades.

3. **Comparação entre Cidades:** O sistema deve possibilitar a comparação direta entre diferentes cidades, apresentando métricas padronizadas para facilitar a análise da competitividade urbana e permitindo que gestores compreendam a relação entre fatores socioeconômicos e desempenho das cidades.

4. **Visualização Interativa de Dados:** O sistema deve apresentar dashboards interativos que exibam gráficos dinâmicos, mapas geoespaciais e tabelas comparativas, facilitando a interpretação dos dados e permitindo ajustes na exibição conforme as preferências do usuário.

5. **Geração e Exportação de Relatórios:** O sistema deve permitir a geração automática de relatórios analíticos em formatos como **PDF**, **CSV** e **Excel**, contendo gráficos, insights e recomendações baseadas na análise dos dados.

Requisitos Não Funcionais (RNF)

1. **Desempenho e Eficiência:** O sistema deve ser otimizado para processar grandes volumes de dados sem comprometer a experiência do usuário. Deve utilizar armazenamento em arquivos **Parquet** para otimizar a leitura e escrita, além de processar cálculos estatísticos de forma eficiente com bibliotecas especializadas como **Dask** e **NumPy**.

2. **Escalabilidade Modular:** A arquitetura do sistema deve ser modular, permitindo sua expansão para suportar um número crescente de usuários e volume de dados. O uso de **APIs REST** facilita a integração com novas funcionalidades e fontes de dados futuras.

3. **Acessibilidade e Interface Intuitiva:** O sistema deve seguir princípios de design acessível, garantindo que usuários com diferentes níveis de experiência em análise de dados possam interpretar as informações corretamente. Deve-se utilizar **Semantic UI** e **Tailwind CSS** para garantir um layout responsivo e intuitivo.

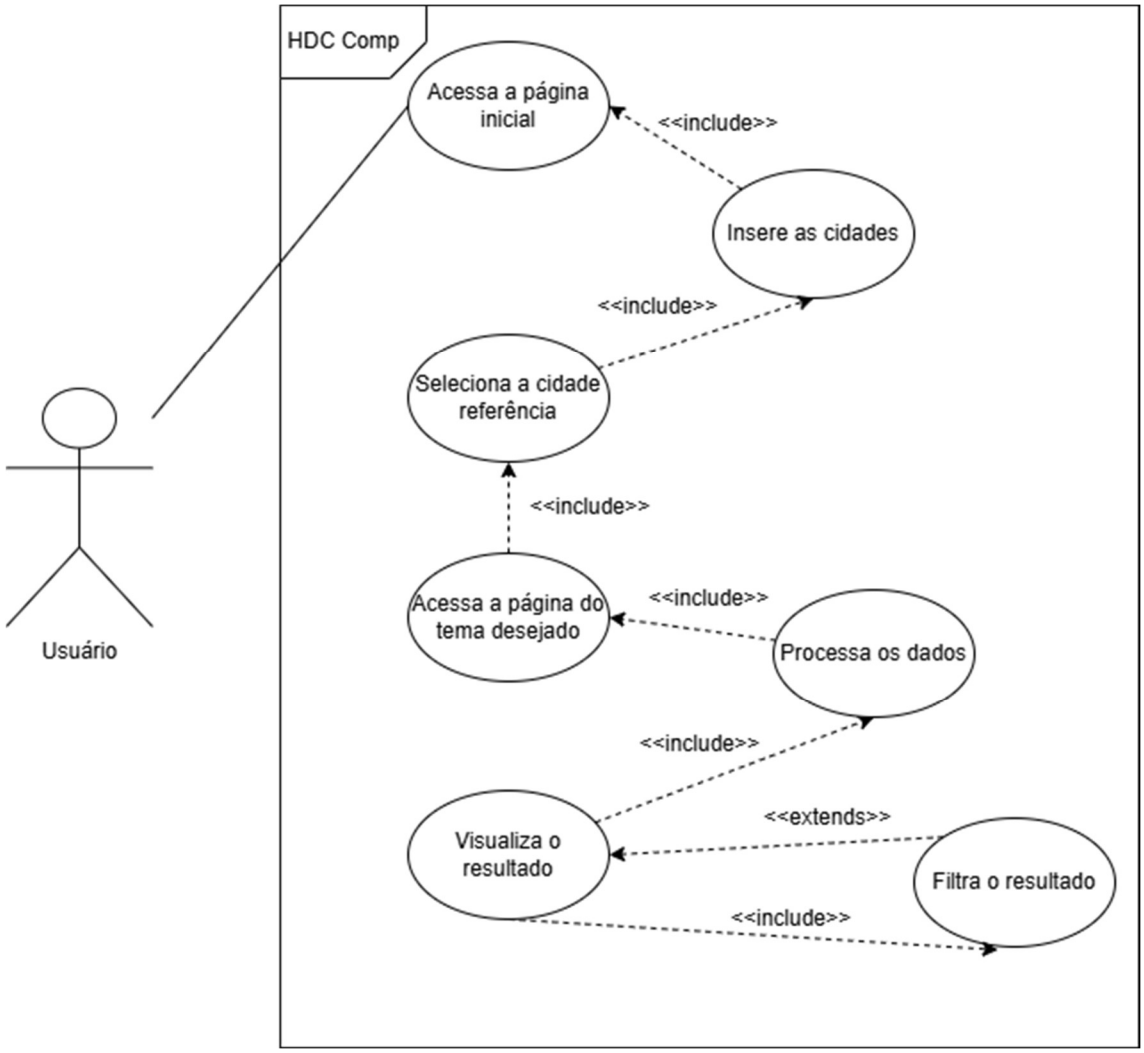
4. **Confiabilidade e Atualização dos Dados:** O sistema deve garantir que os dados apresentados estejam sempre atualizados, implementando mecanismos de atualização automática baseados em fontes oficiais, como IBGE e PNUD.

5.2 DIAGRAMA DE CASO DE USO

5.2.1 Diagrama de caso de uso

SOLICITAÇÃO ELETRÔNICA nº 2942/2025 PROT/CMC (1649851) 23042.002662/2025-90 pg. 48

Figura 4 – diagrama do painel interativo



Fonte: Próprio autor

O diagrama de caso de uso apresentado na **Figura 4** ilustra as principais interações dos usuários com o sistema HDC Comp, focado na análise e comparação de dados socioeconômicos entre cidades. A seguir, estão descritos os elementos e fluxos do sistema:

Acessa a página inicial: O usuário inicia a interação acessando a página inicial do sistema. Essa página oferece uma visão geral das funcionalidades disponíveis e permite iniciar o fluxo de análise.

Insere as cidades: Após acessar a página inicial, o usuário insere as cidades de interesse para análise, selecionando-as por nome ou através de

filtros predefinidos. Essa etapa é essencial para delimitar os dados a serem processados.

Seleciona a cidade referência: O sistema permite que o usuário escolha uma cidade como referência para análise comparativa. Esse processo orienta os cálculos e visualizações gerados posteriormente.

Acessa a página do tema desejado: Nesta etapa, o usuário seleciona o tema específico que deseja analisar, como educação, saúde ou infraestrutura. Cada tema possui dados específicos que serão processados pelo sistema.

Processa os dados: O sistema realiza o processamento dos dados inseridos, utilizando algoritmos para organizar, calcular e preparar as informações relacionadas ao tema escolhido. Essa etapa é automatizada e gera os resultados para visualização.

Visualiza o resultado: Após o processamento, o usuário pode visualizar os resultados em formatos como tabelas ou gráficos interativos. Essa visualização facilita a interpretação e análise das informações.

Filtra o resultado (opcional): Caso esteja disponível e se necessário, o usuário pode aplicar filtros adicionais para refinar os resultados exibidos, como selecionar um intervalo de tempo ou destacar determinadas variáveis. Esse caso de uso é indicado no diagrama como uma extensão opcional (<<extends>>).

5.2.2 Descrição dos Casos de Uso

Tabela 1 - Descrição do caso de uso “Acessa a página inicial”

Caso de uso	UC01 – Acessa a página inicial
Descrição	O usuário acessa a página inicial do sistema em seu navegador.
Atores	Usuário
Pré-condição	O sistema deve estar online e acessível.

SOLICITAÇÃO ELETRÔNICA nº 2942/2025 PROT/CMC (1649851) 23042.002662/2025-90 pg. 50

Pós-condições	• O sistema deve apresentar a página inicial em alguns segundos.
Fluxo principal	1. Abre o navegador de preferência. 2. Digita o endereço para acessar o sistema. 3. Acessa a página inicial.
Fluxo alternativo	1. Abre o navegador de preferência. 2. Digita erroneamente o endereço do sistema. 3. Tenta novamente até acertar o endereço.

Fonte: Próprio autor

Tabela 2 - Descrição do caso de uso "Insere as cidades"

Caso de uso	UC02 – Insere as cidades
Descrição	O usuário insere as cidades que deseja analisar no sistema.
Atores	Usuário
Pré-condição	O usuário deve ter acessado a página inicial
Pós-condições	• As cidades inseridas são armazenadas e exibidas na interface para seleção e análise.
Fluxo principal	1. O sistema exibe o campo de entrada para cidades. 2. O usuário digita os nomes das cidades. 3. O sistema valida e armazena as cidades inseridas.
Fluxo alternativo	1. O usuário insere um nome inválido. 2. O sistema não retorna um resultado e aguarda uma nova entrada.

Fonte: Próprio autor

SOLICITAÇÃO ELETRÔNICA nº 2942/2025 PROT/CMC (1649851) 23042.002662/2025-90 pg. 51

Tabela 3 - Descrição do caso de uso "Seleciona a cidade referência"

Caso de uso	UC03 – Seleciona a cidade referência
Descrição	O usuário seleciona uma cidade como referência para a análise comparativa.
Atores	Usuário
Pré-condição	As cidades já devem ter sido inseridas no sistema.
Pós-condições	• A cidade selecionada é marcada como referência para as análises e gráficos.
Fluxo principal	1. O sistema exibe a lista de cidades inseridas. 2. O usuário seleciona uma cidade. 3. O sistema confirma a seleção e atualiza os dados com base na cidade escolhida.
Fluxo alternativo	1. O usuário não seleciona nenhuma cidade. 2. O sistema exibe uma mensagem solicitando que uma cidade seja selecionada.

Fonte: Próprio autor

Tabela 4 - Descrição do caso de uso "Acessa a página do tema desejado"

Caso de uso	UC04 – Acessa a página do tema desejado
Descrição	O usuário acessa a página específica do tema que deseja analisar, como saúde, educação, ou renda.
Atores	Usuário
Pré-condição	O sistema deve ter os temas cadastrados e organizados para seleção.
Pós-condições	• O sistema exibe os dados e opções disponíveis relacionados ao tema escolhido.

SOLICITAÇÃO ELETRÔNICA nº 2942/2025 PROT/CMC (1649851) 23042.002662/2025-90 pg. 52

Fluxo principal	1. O sistema exibe a lista de temas disponíveis. 2. O usuário seleciona o tema desejado. 3. O sistema redireciona para a página do tema escolhido.
Fluxo alternativo	1. O usuário tenta acessar um tema inexistente. 2. O sistema exibe uma mensagem de erro e espera que o usuário retorne à lista de temas disponíveis.

Fonte: Próprio autor

Tabela 5 - Descrição do caso de uso "Processa os dados"

Caso de uso	UC05 – Processa os dados
Descrição	O sistema realiza o processamento dos dados inseridos e os organiza para análise.
Atores	Sistema
Pré-condição	Os dados devem estar inseridos e válidos.
Pós-condições	• O sistema gera os resultados processados e prontos para exibição.
Fluxo principal	1. O sistema valida os dados inseridos. 2. Realiza cálculos e agrupamentos necessários. 3. Prepara os resultados para serem exibidos.
Fluxo alternativo	-

Fonte: Próprio autor

Tabela 6 - Descrição do caso de uso "Visualiza o resultado"

Caso de uso	UC06 – Visualiza o resultado
Descrição	O usuário visualiza os resultados da análise em tabelas ou gráficos interativos.
Atores	Usuário

Pré-condição	O processamento dos dados deve estar concluído.
Pós-condições	• O usuário compreende os dados apresentados e pode tomar decisões com base neles.
Fluxo principal	1. O sistema exibe os resultados em diferentes formatos visuais. 2. O usuário interage com os gráficos ou tabelas para explorar os dados.
Fluxo alternativo	1. O usuário não encontra o resultado esperado. 2. O sistema permite ajustar os filtros ou realizar um novo processamento.

Fonte: Próprio autor

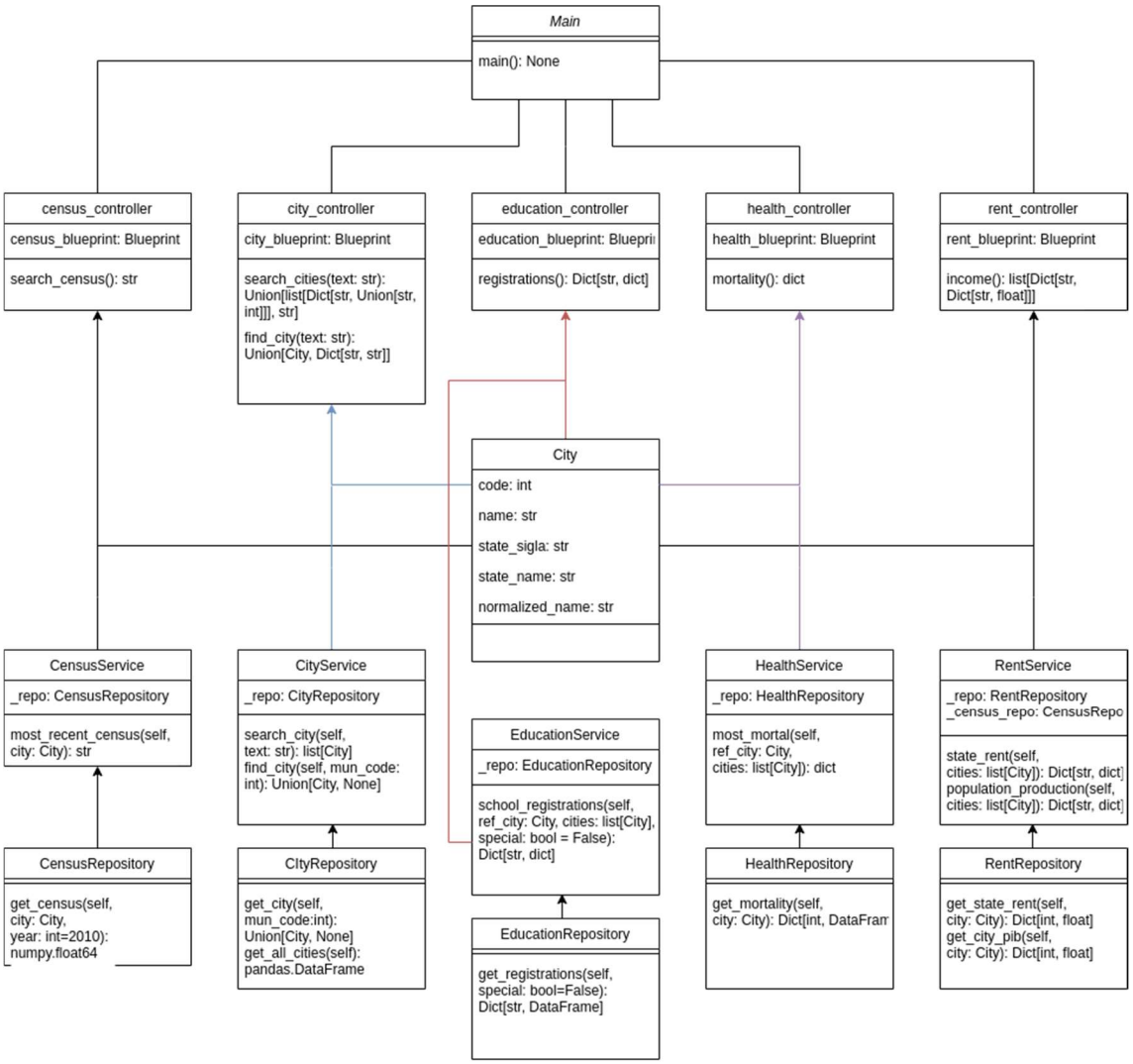
Tabela 7 - Descrição do caso de uso "Filtra o resultado"

Caso de uso	UC07 – Filtra o resultado
Descrição	O usuário aplica filtros nos resultados para refinar a análise e visualizar informações específicas.
Atores	Usuário
Pré-condição	Os resultados devem estar disponíveis para análise.
Pós-condições	• O sistema exibe os resultados filtrados com base nos critérios definidos pelo usuário.
Fluxo principal	1. O sistema exibe opções de filtros disponíveis. 2. O usuário seleciona os critérios desejados. 3. O sistema aplica os filtros e atualiza os resultados exibidos.
Fluxo alternativo	1. O usuário aplica um filtro inválido. 2. O sistema exibe uma mensagem de erro e mantém os resultados originais.

Fonte: Próprio autor

5.2 DIAGRAMA DE CLASSES

Figura 5 – Diagrama de classes



Fonte: Próprio Autor

O diagrama de classes apresentado na **Figura 5** representa a estrutura organizacional do sistema, destacando as relações entre controladores, serviços, repositórios e o modelo `City`. Este modelo atua como a base para armazenar e manipular os dados fundamentais das cidades, enquanto as demais camadas (controladores, serviços e repositórios) garantem o fluxo de informações e a execução das funcionalidades do sistema.

O sistema segue uma arquitetura em camadas bem definida, composta por **Controladores (Controllers)**, **Serviços (Services)**, **Repositórios**

SOLICITAÇÃO ELETRÔNICA nº 2942/2025 PROT/CMC (1649851) 23042.002662/2025-90 pg. 56

(Repositories) e **Modelos (Models)**. O modelo `City` é central para o funcionamento do sistema, servindo como a estrutura de dados que representa as cidades e é amplamente utilizado nas operações de busca, análise e manipulação de dados.

5.3.1 Principais Componentes

- **Camada de Apresentação (Main)**

Descrição: O main representa a camada de apresentação do sistema, sendo responsável por inicializar a aplicação **Flask** e gerenciar as rotas principais da **API**. Ele atua como ponto de entrada do **backend**, configurando os controladores e registrando os **blueprints** para cada módulo.

Funcionalidades Principais:

- Inicializa a aplicação **Flask**.
- Configura o suporte a **Cross-Origin Resource Sharing (CORS)**, permitindo o consumo da **API** por diferentes domínios.
- Define as configurações da **API**, como cabeçalhos de resposta e codificação de caracteres.
- Registra os **blueprints** dos controladores (education_blueprint, rent_blueprint, city_blueprint, health_blueprint, census_blueprint).
- Define a porta e o **host** para execução da aplicação.

Importância no Sistema: O main é responsável por orquestrar a comunicação entre a camada de controle e os serviços, garantindo que as requisições sejam devidamente processadas e encaminhadas. Ele permite a modularização da aplicação ao utilizar **blueprints**, facilitando a manutenção e a escalabilidade do sistema.

- **Modelo (`City`)**

Descrição: O modelo `City` representa as informações essenciais de cada cidade no sistema, sendo utilizado como base para operações de armazenamento, consulta e manipulação de dados.

Atributos:

`code`: Código único da cidade, utilizado como identificador principal.

`name`: Nome da cidade.

`state_sigla`: Sigla do estado ao qual a cidade pertence.

`state_name`: Nome completo do estado.

`normalized_name`: Nome normalizado da cidade para facilitar comparações e buscas consistentes.

Importância no Sistema: É utilizado em todos os módulos para garantir que os dados sejam estruturados de maneira consistente e possam ser compartilhados entre diferentes componentes, como controladores, serviços e repositórios.

- **Controladores (Controllers)**

Descrição: Responsáveis por gerenciar as requisições externas e encaminhá-las para os serviços correspondentes. Também retornam as respostas ao usuário ou à interface do sistema.

Classes:

`census_controller`: Gerencia operações relacionadas ao censo.

`city_controller`: Gerencia buscas e consultas relacionadas às cidades.

`education_controller`: Gerencia dados de matrículas escolares.

`health_controller`: Gerencia informações de saúde, como taxas de mortalidade.

`rent_controller`: Gerencia dados econômicos, como renda e PIB.

- **Serviços (Services)**

Descrição: Contêm a lógica de negócio do sistema, coordenando as operações entre controladores e repositórios.

Classes:

`CensusService`: Processa dados relacionados ao censo.

`CityService`: Realiza operações de busca e consulta sobre as cidades.

`EducationService`: Lida com a análise de matrículas escolares.

`HealthService`: Processa dados sobre saúde, como causas de mortalidade.

`RentService`: Gerencia informações econômicas, como PIB e renda per capita.

- **Repositórios (Repositories)**

Descrição: Interagem diretamente com as fontes de dados, realizando operações de armazenamento, recuperação e manipulação.

Classes:

`CensusRepository`: Gerencia o acesso aos dados do censo.

`CityRepository`: Gerencia os dados das cidades.

`EducationRepository`: Lida com os dados relacionados à educação.

`HealthRepository`: Gerencia informações de saúde pública.

`RentRepository`: Lida com dados econômicos, como renda e PIB.

Fluxo de Dados

Interação com o usuário: As requisições do usuário são capturadas pelos **Controladores** (ex.: consulta de dados sobre saúde ou educação).

Processamento de lógica de negócio: Os **Controladores** encaminham as requisições para os **Serviços**, que aplicam a lógica de negócio e coordenam as operações necessárias.

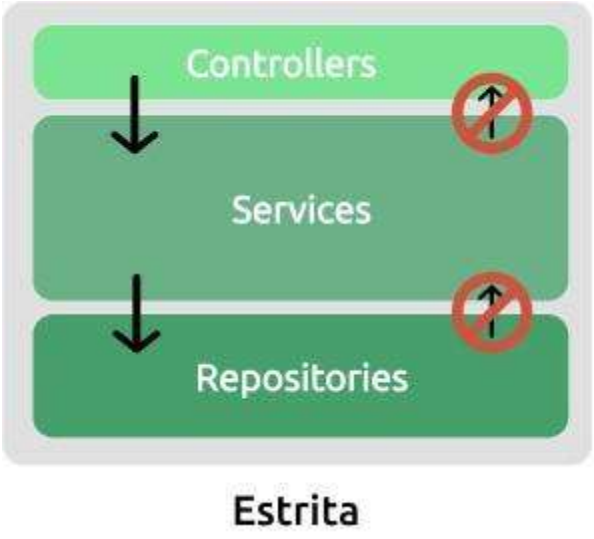
Interação com o banco de dados: Os **Serviços** acessam os **Repositórios** para buscar ou manipular dados no banco. O modelo `City` é usado como base para estruturar as informações durante essas operações.

Resposta ao usuário: Após o processamento, os **Controladores** retornam os resultados ao usuário de forma estruturada, como gráficos ou tabelas.

5.3 ARQUITETURA DO SISTEMA

A arquitetura utilizada no sistema segue o modelo em camadas estritas. Nesse modelo, cada camada desempenha um papel específico e comunica-se apenas com a camada imediatamente inferior, garantindo uma separação clara de responsabilidades. Essa abordagem promove organização, escalabilidade e manutenibilidade no desenvolvimento do sistema. (MARTIN, Robert C., 2019)

Figura 6 – Arquitetura em camadas estritas



Fonte: (GuiaDev, 2023)

Descrição da Arquitetura

SOLICITAÇÃO ELETRÔNICA nº 2942/2025 PROT/CMC (1649851) 23042.002662/2025-90 pg. 60

1. **Camada de Apresentação:**

A **camada de apresentação** é responsável por fornecer a interface gráfica e interativa para o usuário final. No sistema, essa camada consiste em páginas **HTML** renderizadas pelo **Flask**, com suporte de **CSS**, **JavaScript** e bibliotecas como **Semantic UI** e **Font Awesome** para componentes visuais.

Principais responsabilidades:

- Exibir tabelas, gráficos e relatórios gerados pelo sistema.
- Permitir que o usuário selecione indicadores e insira dados para análise.
- Interagir com o **backend** via requisições **HTTP** (**fetch API** no **JavaScript**).
- Renderizar os templates **HTML** com dados dinâmicos enviados pelo **Flask**.

Exemplo: O usuário seleciona um conjunto de cidades e indicadores para análise. A interface envia a solicitação ao **backend** e exibe os gráficos gerados.

2. **Controllers:**

A camada de **Controllers** é responsável por interagir diretamente com o usuário ou com solicitações externas (como requisições HTTP). No contexto do sistema, esta camada atua como a entrada principal para as funcionalidades. Por exemplo:

- Recebe solicitações para acessar dados socioeconômicos ou visualizar análises específicas.
- Valida os dados de entrada do usuário antes de encaminhá-los para os **Serviços**.
- Retorna os resultados processados ao usuário, exibindo-os na interface, seja em tabelas, gráficos ou mapas interativos.

- **Exemplo:** O usuário solicita uma análise de uma cidade específica; o **Controller** valida o pedido e encaminha-o à camada de **Services**.

3. **Services (Camada de Negócio):**

A camada de **Services** implementa a lógica de negócio do sistema. É o núcleo funcional que processa os dados e aplica regras específicas. No presente sistema, essa camada é responsável por:

- Coordenar o fluxo entre os **Controllers** e os **Repositories**.
- Processar os dados socioeconômicos, realizando cálculos ou análises específicas, como o agrupamento de cidades por indicadores ou a comparação de dados de referência.
- Garantir que a lógica de negócio seja executada corretamente, independentemente do tipo de entrada.
- **Exemplo:** O **Service** recebe o pedido para comparar indicadores de educação entre cidades, obtém os dados da camada de **Repositories**, processa-os e retorna os resultados ao **Controller**.

4. **Repositories (Camada de Acesso a Dados):**

A camada de **Repositories** é responsável por interagir com a base de dados, abstraindo os detalhes de armazenamento e consulta. No sistema descrito, essa camada realiza:

- Consultas às bases de dados já tratados, provenientes das fontes externas (ex.: IBGE, PNUD).
- Operações CRUD (criar, ler, atualizar e deletar) para garantir que os dados socioeconômicos estejam atualizados e acessíveis.
- Fornecimento de dados brutos para a camada de **Services**.
- **Exemplo:** O **Repository** busca os dados de saúde e educação de uma cidade solicitada, retornando-os para que sejam processados pela camada de **Services**.

A arquitetura em camadas estritas implementada no sistema descrito garante que o fluxo de dados seja bem definido, desde a interação inicial com o usuário até o processamento e exibição dos resultados. Esse modelo promove um sistema robusto, escalável e alinhado aos objetivos do trabalho, que envolvem a análise e gestão de dados socioeconômicos das cidades brasileiras para melhorar a competitividade urbana.

SOLICITAÇÃO ELETRÔNICA nº 2942/2025 PROT/CMC (1649851) 23042.002662/2025-90 pg. 63

6 SISTEMA

6.1 INTRODUÇÃO DO SISTEMA

O presente capítulo tem como objetivo apresentar o sistema desenvolvido, detalhando sua estrutura, funcionamento e funcionalidades principais. Serão descritas a arquitetura do sistema, os componentes que o compõem, incluindo suas classes, atributos e métodos, bem como as interações entre eles. Além disso, serão explorados os principais casos de uso do sistema, destacando como ele organiza, processa e apresenta dados socioeconômicos de forma acessível, visando auxiliar na análise e gestão das cidades brasileiras e contribuir para a tomada de decisão estratégica.

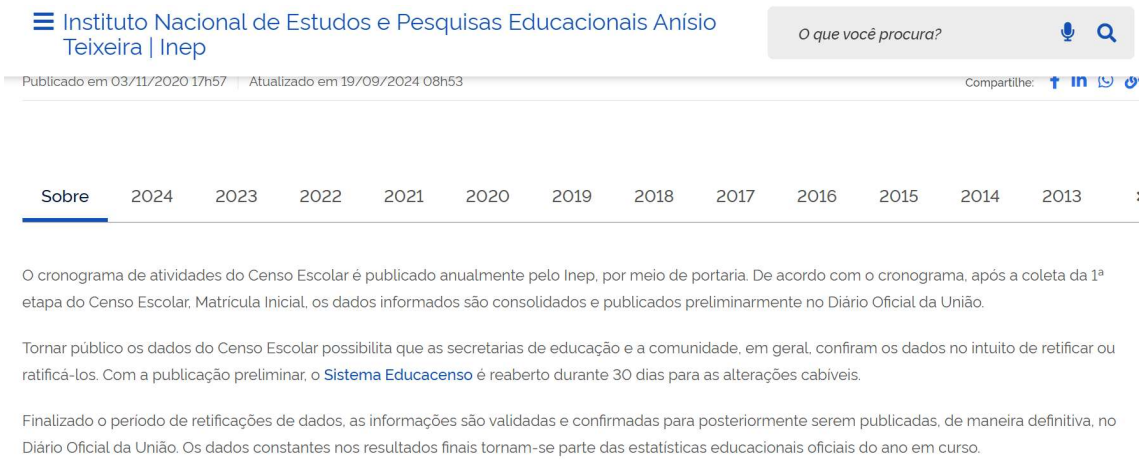
6.2 MAPEAMENTO DOS DADOS

O presente capítulo aborda o mapeamento e tratamento dos dados utilizados no sistema. Serão detalhadas as etapas de coleta, organização e pré-processamento das informações provenientes de fontes confiáveis, como IBGE e PNUD, essenciais para garantir a qualidade das análises realizadas. Além disso, serão apresentados os métodos aplicados para lidar com inconsistências, dados ausentes e normalização, bem como as ferramentas utilizadas para facilitar essas tarefas. Esse processo foi fundamental para estruturar os dados de forma eficiente e assegurar a precisão dos resultados obtidos pelo sistema.

Para a extração dos dados foi utilizada fontes confiáveis abrangendo indicadores relacionados a educação, saúde, renda e demografia, que são fundamentais para as análises socioeconômicas propostas. A seguir, cada uma das fontes é contextualizada, destacando os dados extraídos e sua aplicação no sistema. Os dados relacionados à **educação** foram obtidos por meio do Censo Escolar, realizado pelo Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira (INEP)

Figura 7 – Dados de educação

SOLICITAÇÃO ELETRÔNICA nº 2942/2025 PROT/CMC (1649851) 23042.002662/2025-90 pg. 64



Fonte: (INEP, 2020)

O Censo Escolar forneceu informações detalhadas sobre matrículas, infraestrutura das escolas e taxa de alfabetização, indicadores fundamentais para avaliar o panorama educacional das cidades brasileiras e sua relação com a competitividade urbana. Na área da **saúde**, os dados foram extraídos do Sistema de Informações sobre Mortalidade (SIM)

Figura 8 – Dados de saúde

Fonte: (gov.br - SIM)

Esses dados incluem estatísticas sobre mortalidade e causas de óbito, permitindo identificar padrões e desafios relacionados à saúde pública nas cidades. Para complementar essa análise, foi utilizada a Tabela CID-10 que categoriza detalhadamente as causas de morte, fornecendo informações essenciais para entender a situação de saúde das cidades. No que diz respeito

à **renda**, foram utilizadas duas fontes principais. A Pesquisa Nacional por Amostra de Domicílios Contínua (PNAD Contínua), disponibilizado pelo IBGE.

Figura 9 – Dados de renda

Fonte: (IBGE)

Estes trouxeram informações sobre rendimentos domiciliares médios e taxa de ocupação. Além disso, dados sobre o Produto Interno Bruto (PIB) por município foram extraídos da Base dos Dados do IBGE abrangendo setores como agricultura, indústria e serviços, o que permitiu analisar o desempenho econômico das cidades brasileiras. Por fim, informações **demográficas** foram coletadas a partir do Censo Demográfico 2022. Esses dados forneceram uma visão detalhada sobre as características populacionais, densidade demográfica e condições dos domicílios urbanos e rurais.

Figura 10 – Dados demográficos

Fonte: (IBGE, 2022)

Após a coleta, os dados passaram por um processo rigoroso de tratamento. Foram realizadas etapas de limpeza, como remoção de valores nulos e inconsistentes, normalização das variáveis para permitir comparações entre as cidades e conversão de formatos para integração no sistema. Além disso, os dados foram validados para assegurar sua integridade e alinhamento com os objetivos do estudo. Essas etapas garantiram a qualidade e consistência dos dados, proporcionando uma base sólida para as análises realizadas pelo sistema.

6.3 BACKEND

O backend do sistema foi desenvolvido utilizando o framework **Flask** em **Python**, que serviu como a base para a implementação da lógica de negócios, processamento de dados e comunicação com o frontend. O **Flask** foi escolhido por sua leveza e flexibilidade, permitindo a criação de **APIs REST** modulares e altamente eficientes. O sistema foi projetado seguindo a arquitetura em camadas, garantindo uma separação clara de responsabilidades, o que facilitou a organização e a escalabilidade do projeto.

A estrutura do backend foi dividida em **controllers**, **services** e **repositories**, cada um desempenhando funções específicas no sistema. Essa divisão não apenas melhorou a modularidade, mas também tornou o código mais legível e fácil de manter.

Os **controllers** foram responsáveis por gerenciar as requisições recebidas do frontend. Cada rota definida no **Flask** foi associada a um controlador, que cuidava de validar os parâmetros de entrada, encaminhar as requisições para os serviços e formatar as respostas para serem retornadas ao usuário no formato **JSON**. Por exemplo, ao receber uma solicitação para comparar indicadores entre duas cidades, o controlador validava os dados enviados pelo usuário, como os nomes das cidades e os indicadores selecionados, e então repassava a solicitação à camada de serviços.

A camada de **services** implementou a lógica de negócios do sistema. Aqui foram processadas as análises mais complexas, como cálculos de médias, tendências e padrões nos dados socioeconômicos. Além disso, essa

SOLICITAÇÃO ELETRÔNICA nº 2942/2025 PROT/CMC (1649851) 23042.002662/2025-90 pg. 67

camada centralizou a aplicação de regras específicas para cada funcionalidade, garantindo que a lógica fosse reutilizável e consistente em todo o sistema.

Na camada de **repositories**, foi realizada a interação direta com os arquivos de dados, que estavam armazenados no formato **.parquet**. Para lidar com esses arquivos de forma eficiente, foi utilizada a biblioteca **Dask**, que permitiu o processamento paralelo dos dados. Isso foi fundamental para garantir que o sistema pudesse trabalhar com grandes volumes de dados, realizando consultas rápidas e precisas. A escolha pelo formato **.parquet** foi feita devido à sua eficiência em termos de armazenamento e desempenho em leitura e escrita.

Após o desenvolvimento e testes em **Python**, todo o backend foi compilado para **C** utilizando o **Cython**. Essa decisão foi tomada para otimizar o desempenho do sistema, especialmente em operações que envolviam cálculos intensivos e processamento de grandes conjuntos de dados. O **Cython** converteu o código **Python** em código nativo **C**, que foi então compilado para possuir alto desempenho. O processo de transpilar o código Python para C exigiu algumas otimizações adicionais:

Foi necessário ‘**tipar’ explicitamente variáveis** frequentemente utilizadas, especialmente aquelas instanciadas dentro de loops e iterações como na **Figura 11**. Essas otimizações reduzem o tempo de execução em operações repetitivas.

Figura 11 – tipagem de variáveis em **Python**

```python
def population_production(self, cities: list[City]) -> Dict[str, dict]:
        - The keys are strings representing the city codes.
        - The values are floats representing the GDP per capita for each c
    """
    completer = Completer()
    result: dict = {}
    for i in cities:
        code: str = str(i.code)
        result[code] = self._repo.get_city_pib(i)
        census_dict: dict = {}
        for j in result[code]:
            year: int = 2010 if j <= 2019 else 2022
            census: np.float64 = self._census_repo.get_census(i, year)
            result[code][j] = (result[code][j] * 1000) / census
            census_dict[j] = census
        completer.complete_pib(census_dict, result)
    return result
```

Fonte: Próprio Autor

O modelo **City**, que era amplamente utilizado no backend, foi reescrito em um arquivo **.pyx** (próprio da linguagem **Cython**) conforme **Figura 12** para garantir maior controle sobre a compilação. Isso permitiu ajustar detalhes finos do código, otimizando operações relacionadas ao processamento e manipulação de instâncias desse modelo. Com isso, o backend ganhou as seguintes vantagens:

Redução no tempo de execução: A execução como código nativo C permitiu que o sistema processasse requisições e cálculos de forma mais rápida (conforme o **benchmark** da **Figura 13**).

Maior escalabilidade: Os aumentos no desempenho garantem que o sistema pudesse lidar com mais usuários e requisições simultâneas sem comprometer a experiência.

Figura 12 – Modelo city na linguagem **Cython**

city.pyx

```
cdef class City:
    cdef int _code
    cdef str _name
    cdef str _state_sigla
    cdef str _state_name
    cdef str _normalized_name

    def __init__(self, int code, str name, str state_sigla, str state_name, str normalized_name):
        if len(code) != 7:  # Garantindo que o código não ultrapasse 7 dígitos
            raise ValueError("O código deve ser um número de 7 dígitos.")

        # Validando o tamanho da sigla do estado (deve ser 2 caracteres)
        if len(state_sigla) != 2:
            raise ValueError("A sigla do estado deve ter exatamente 2 caracteres.")

        # Validando o nome do estado (tamanho máximo de 20 caracteres)
        if len(state_name) > 20:
            raise ValueError("O nome do estado não pode ultrapassar 20 caracteres.")

        # Atribuindo os valores
        self._code = code
        self._name = name
        self._state_sigla = state_sigla.upper()  # Garantindo que a sigla seja em maiúsculas
        self._state_name = state_name
        self._normalized_name = normalized_name

    @property
    def code(self):
        return self._code

    @property
    def name(self):
        return self._name
```

Fonte: Próprio Autor

Figura 13 – Comparativo de **C** e **Python**

```
/media/bruno-wesen/Strange-Data-S/Programas/Projects/IDH-Comp
Tempo total da requisição para url_c: 114.5703 segundos
Tempo total da requisição para url_python: 117.8567 segundos
Resposta da url_c: {
  "cid_table": {
    "A00": "A00   Colera",
    "A000": "A00.0 Colera dev Vibrio cholerae 01 biot ...
Resposta da url_python: {
  "cid_table": {
    "A00": "A00   Colera",
    "A000": "A00.0 Colera dev Vibrio cholerae 01 biot ...
```

Fonte: Próprio Autor

6.4 FRONTEND

O frontend do sistema foi projetado para oferecer uma interface interativa, responsiva e intuitiva, garantindo que os usuários possam acessar e analisar os dados socioeconômicos de forma eficiente. Para atingir esse objetivo, foram utilizadas tecnologias modernas e ferramentas que facilitaram tanto o desenvolvimento quanto a experiência do usuário.

A estrutura do frontend foi construída com o suporte do framework **Flask**, que acessa os templates **HTML** por meio de uma pasta específica e os arquivos **CSS** pela pasta **static**, garantindo uma organização clara. O **Flask** também desempenhou um papel essencial na integração entre o **Python** e o **JavaScript**, permitindo o uso de variáveis **Python** diretamente nos **scripts**. Para isso, foi necessário criar rotas específicas no **Flask**, que permitiram a importação e o uso dos arquivos **JavaScript** nos templates **HTML**.

Os componentes visuais do sistema foram enriquecidos com a utilização do **Semantic UI**, que ofereceu elementos estilizados como botões e tabelas, proporcionando uma aparência moderna e funcional. Além disso, ícones do **Font Awesome** foram incorporados para melhorar a navegação e a clareza da interface.

A interação entre o frontend e o backend foi realizada utilizando a biblioteca **python-requests**, que permitiu que o **Flask** se comunicasse com o backend do sistema, enviando as requisições necessárias e recebendo os resultados para exibição no navegador. Por outro lado, o **JavaScript** foi utilizado para interagir diretamente com o **Flask** no frontend, por meio de **fetch**, especialmente para o envio de dados do **JavaScript** para o **Flask**. Um exemplo desse fluxo pode ser observado na **Figura 14**, onde o **fetch** é utilizado para enviar dados a uma rota específica e redirecionar para outra página.

SOLICITAÇÃO ELETRÔNICA nº 2942/2025 PROT/CMC (1649851) 23042.002662/2025-90 pg. 71

Figura 14 – **js fetch** para rota do frontend

```
    for (let i = 0; i < data.cities.length; i++) {
        const response = await fetch('/census', {
        method: 'POST',
        headers: {
            'Content-Type': 'application/json',
        },
        body: JSON.stringify(data.cities[i]),
        });
        data.cities[i] = await response.json();
    }

    response = await fetch('/intermediate_route'+route, {
        method: 'POST',
        headers: {
            'Content-Type': 'application/json',
        },
        body: JSON.stringify(data),
    });

    // End
    element.className = originalClasses;
    const state = {
        selectedCities: selectedCities,
        cities: cities,
        selectedOption: selectedOption.id,
        city_components: city_components,
        data: data
    };
    sessionStorage.setItem('pageState', JSON.stringify(state))
    window.location.href=route;
}
```

Fonte: Próprio Autor

A camada de controle no Flask foi projetada para receber os dados processados pelo JavaScript, conforme mostrado na **Figura 15**. Os dados recebidos são enviados ao backend utilizando a biblioteca **python-requests** e, posteriormente, armazenados no navegador utilizando a funcionalidade de

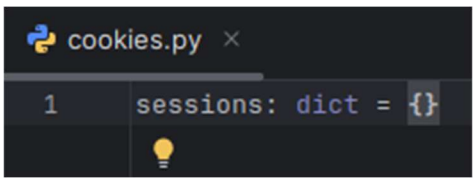
cookies e sessões. A geração de um identificador único para cada usuário foi implementada por meio de um sistema de sessões, representado pela variável `sessions` criado no arquivo de gerenciamento de cookies conforme **Figura 16**, que associa o identificador aos valores e resultados da requisição ao backend, como exemplificado na rota “intermediate_route” da **Figura 15**.

Figura 15 – rota **Flask** esperando requisição **fetch** do **js**

```python
@home_blueprint.route( rule: "/intermediate_route/<route>", methods=["POST"])
def intermediate_route(route: str):
    route = 'rent' if route == 'income' else route
    data = request.get_json()
    response = requests.get(current_app.config['BACKEND_API'] + '/' + route, json=data)
    if not session.get('id'):
        session['id'] = str(uuid.uuid4())
    sessions[session['id']] = {'data': data, 'result': response.json(), 'context': route}
    return 'Recebido com sucesso!', 200


@home_blueprint.route( rule: "/census", methods=["POST"])
def census_route():
    city = request.get_json()
    response = requests.get(current_app.config['BACKEND_API'] + '/census', json=city)
    city['census'] = response.text
    return city
```

Fonte: Próprio Autor

Figura 16 – Arquivo de gerenciamento de **cookies**



Fonte: Próprio Autor

A geração de gráficos, considerada uma das funcionalidades principais do sistema, foi delegada à camada de serviços. Essa decisão organizou o código e garantiu que a lógica de visualização fosse mantida separada do controle de fluxo principal. O controlador **Flask** foi configurado para chamar a

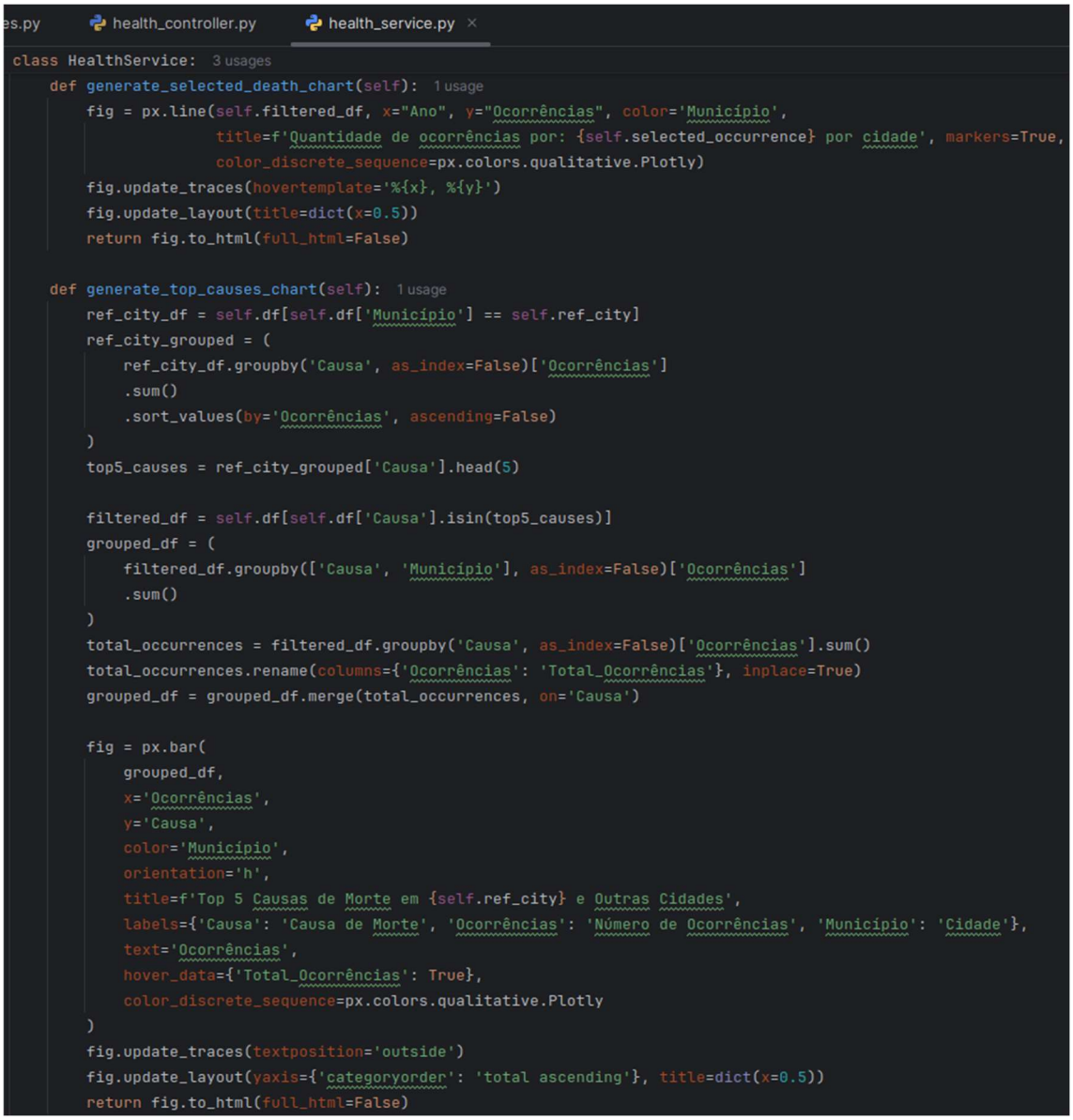
camada de serviços e, em seguida, retornar os templates **HTML** com os gráficos gerados. Esse fluxo pode ser observado na **Figura 17**, enquanto o detalhamento da geração dos gráficos pela camada de serviços é mostrado na **Figura 18**.

Figura 17 – Rota de saúde

```
@health_blueprint.route( rule: "/", methods=["GET"])
@health_blueprint.route( rule: "/<text>", methods=["GET"])
@cross_origin()
def health_page(text=None):
    data = sessions[session['id']]

    if data['context'] != 'health':
        return 'You dont have the results for this page, go to home page select the cities and choose your context.', 405

    ref_city = data['data']['ref_city']
    cities = data['data']['cities']
    result = data['result']
    data['selected_cause'] = text
    service = HealthService(ref_city, cities, result, selected_cause=text)

    return render_template( template_name_or_list: 'health.html', ref_city=ref_city, cities=cities,
                           title=service.get_all_cause_cid_table(),
                           selected_cause_death_html=service.generate_selected_death_chart(),
                           top_common_cause_html=service.generate_top_causes_chart(),
                           selected_cause_death_age_html=service.generate_selected_cause_with_age_chart(),
                           selected_cause_average_age_html=service.generate_selected_cause_average_age_chart())
```

Fonte: Próprio Autor

Figura 18 – Serviço de saúde



```
es.py    health_controller.py    health_service.py ×
class HealthService:  3 usages
    def generate_selected_death_chart(self):  1 usage
        fig = px.line(self.filtered_df, x="Ano", y="Ocorrências", color='Município',
                      title=f'Quantidade de ocorrências por: {self.selected_occurrence} por cidade', markers=True,
                      color_discrete_sequence=px.colors.qualitative.Plotly)
        fig.update_traces(hovertemplate='%{x}, %{y}')
        fig.update_layout(title=dict(x=0.5))
        return fig.to_html(full_html=False)

    def generate_top_causes_chart(self):  1 usage
        ref_city_df = self.df[self.df['Município'] == self.ref_city]
        ref_city_grouped = (
            ref_city_df.groupby('Causa', as_index=False)['Ocorrências']
            .sum()
            .sort_values(by='Ocorrências', ascending=False)
        )
        top5_causes = ref_city_grouped['Causa'].head(5)

        filtered_df = self.df[self.df['Causa'].isin(top5_causes)]
        grouped_df = (
            filtered_df.groupby(['Causa', 'Municipio'], as_index=False)['Ocorrências']
            .sum()
        )
        total_occurrences = filtered_df.groupby('Causa', as_index=False)['Ocorrências'].sum()
        total_occurrences.rename(columns={'Ocorrências': 'Total_Ocorrências'}, inplace=True)
        grouped_df = grouped_df.merge(total_occurrences, on='Causa')

        fig = px.bar(
            grouped_df,
            x='Ocorrências',
            y='Causa',
            color='Município',
            orientation='h',
            title=f'Top 5 Causas de Morte em {self.ref_city} e Outras Cidades',
            labels={'Causa': 'Causa de Morte', 'Ocorrências': 'Número de Ocorrências', 'Município': 'Cidade'},
            text='Ocorrências',
            hover_data={'Total_Ocorrências': True},
            color_discrete_sequence=px.colors.qualitative.Plotly
        )
        fig.update_traces(textposition='outside')
        fig.update_layout(yaxis={'categoryorder': 'total ascending'}, title=dict(x=0.5))
        return fig.to_html(full_html=False)
```

Fonte: Próprio Autor

A utilização de gráficos interativos foi essencial para transformar os dados socioeconômicos em informações visuais acessíveis. A biblioteca **Plotly** foi utilizada para criar gráficos dinâmicos e detalhados, integrando-os aos templates **HTML** para facilitar a análise pelo usuário. Os gráficos foram configurados para exibir comparações entre indicadores e cidades, permitindo interatividade por meio de zoom, filtros e seleções dinâmicas.

Os designs de interface do sistema foi projetado com um foco significativo na experiência do usuário, buscando não apenas funcionalidade,

mas também uma apresentação visual atraente e moderna. Para alcançar esse objetivo, foram utilizados designs gerados por inteligência artificial, especificamente por ferramentas como a **DALL-E**, que forneceram inspiração para a construção das páginas presentes no sistema.

Os designs gerados contemplaram tanto layouts para **desktop** quanto para dispositivos móveis, permitindo criar interfaces que se adaptam perfeitamente a diferentes tamanhos de tela. A **Figura 19** ilustra um dos designs **desktop** inspirados pelas imagens da **DALL-E**, enquanto a **Figura 20** apresenta um design voltado para dispositivos móveis. Esses layouts serviram como base para a construção das páginas, garantindo um design coeso e visualmente agradável em ambas as versões.

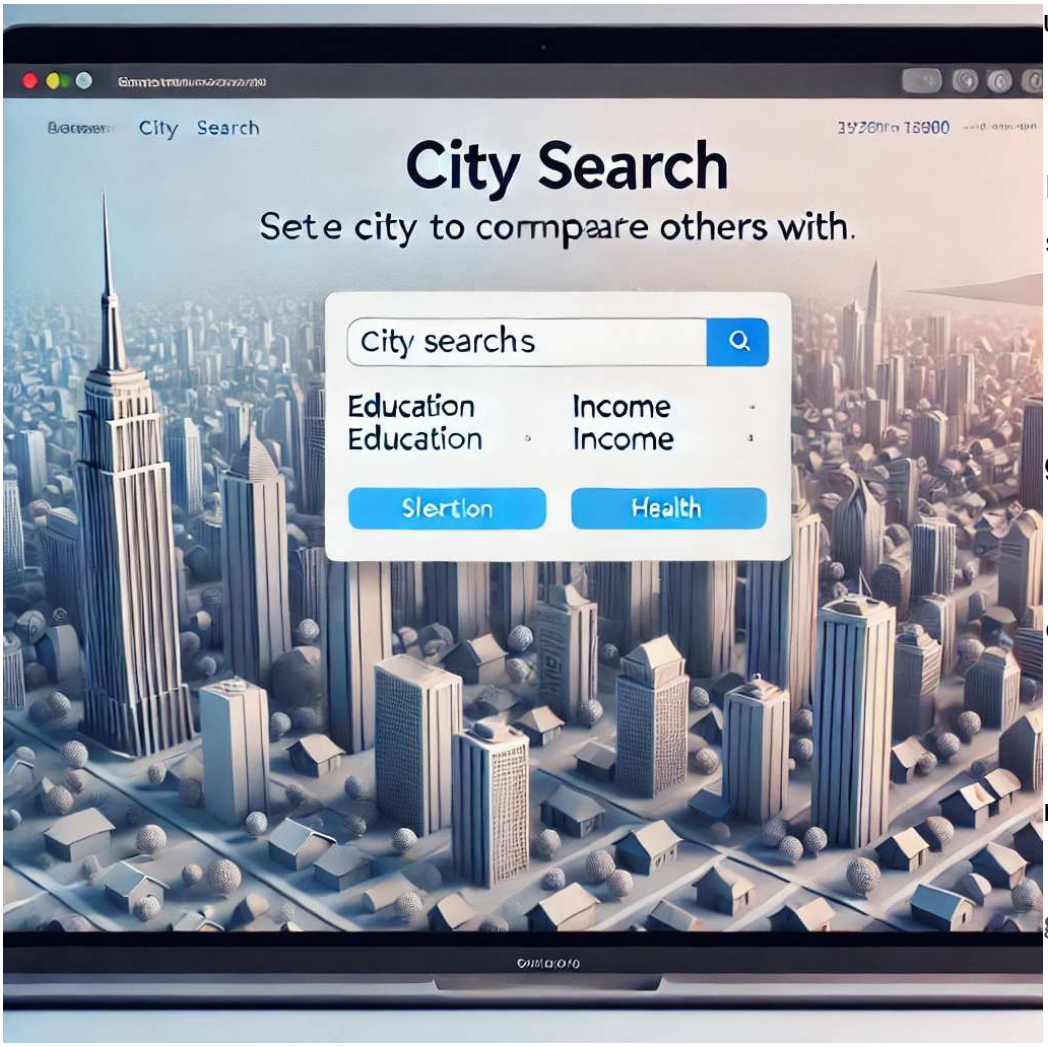


Figura 19 – Design da página inicial

Fonte: gerado

por DALL-E em 12/01/2025.

Figura 20 – Design da página inicial responsiva

Fonte: gerado por DALL-E em 12/01/2025.

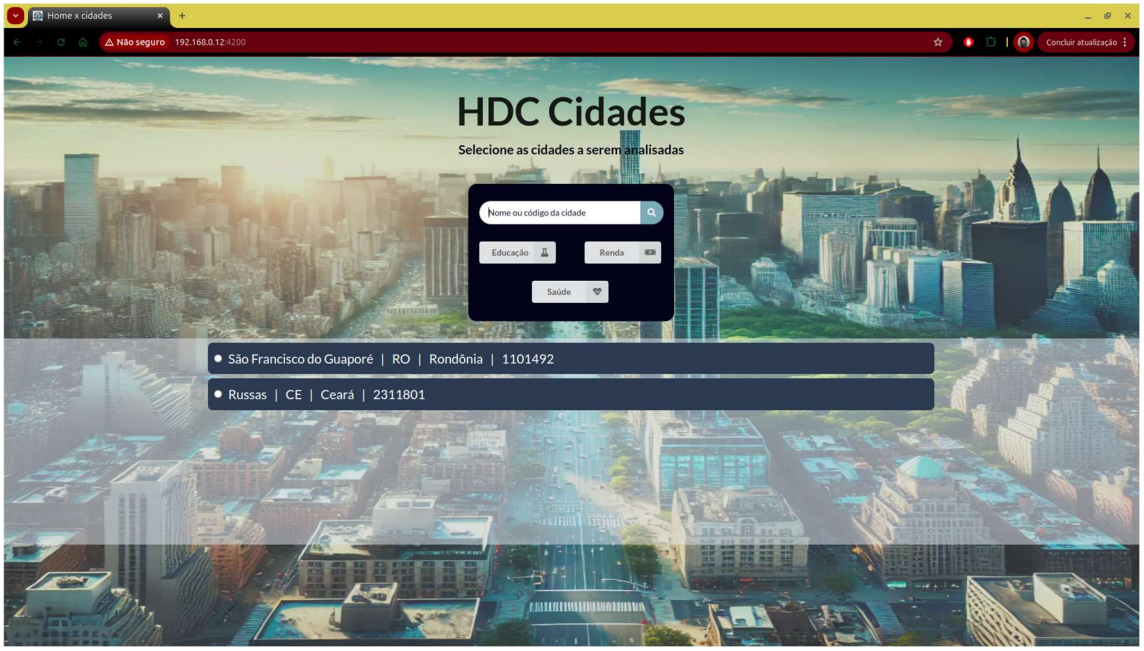
Além disso, as imagens de fundo presentes no sistema também foram geradas pela mesma ferramenta de inteligência artificial, contribuindo para um aspecto visual único e profissional. A **Figura 21** exemplifica uma dessas imagens, que foi incorporada como elemento decorativo, complementando o estilo moderno da interface.

Figura 21 – Imagem de fundo da página inicial

Fonte: gerado por DALL-E em 12/01/2025.

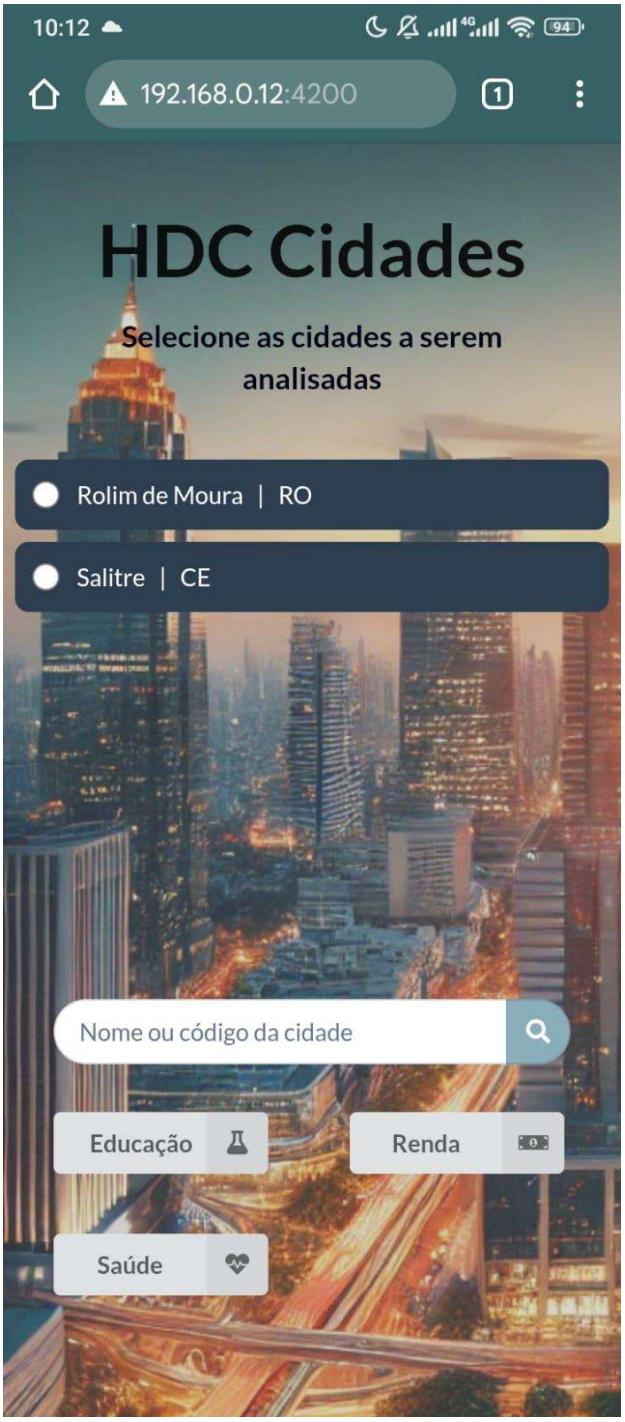
O resultado dessa integração entre designs gerados por inteligência artificial e as funcionalidades do sistema pode ser visto na página inicial, apresentada na **Figura 22** para a versão **desktop** e na **Figura 23** para a versão mobile. Essas páginas iniciais foram projetadas para serem convidativas e intuitivas, oferecendo aos usuários uma experiência de navegação fluida e responsiva, enquanto destacam as principais funcionalidades do sistema de forma clara e acessível.

Figura 22 – Página inicial **Desktop**



Fonte: gerado por DALL-E em 12/01/2025.

Figura 23 – Página inicial dispositivo móvel



Fonte: Próprio Autor

6.5.1 Temas

Neste subcapítulo, são abordados os principais temas contemplados pelo sistema, que foram organizados em eixos estratégicos para facilitar a análise e visualização dos dados socioeconômicos. Esses temas incluem educação, renda e saúde, considerados pilares essenciais para compreender a competitividade das cidades brasileiras. Cada tema foi estruturado para permitir a exploração detalhada dos indicadores associados, proporcionando pensamentos sobre os desafios e avanços em cada área. Além disso, a integração dos dados foi realizada de forma a garantir coesão e acessibilidade, facilitando a navegação e o entendimento pelos usuários do sistema. Nos próximos tópicos, cada tema será detalhado, destacando os dados utilizados, as funcionalidades associadas e os resultados obtidos.

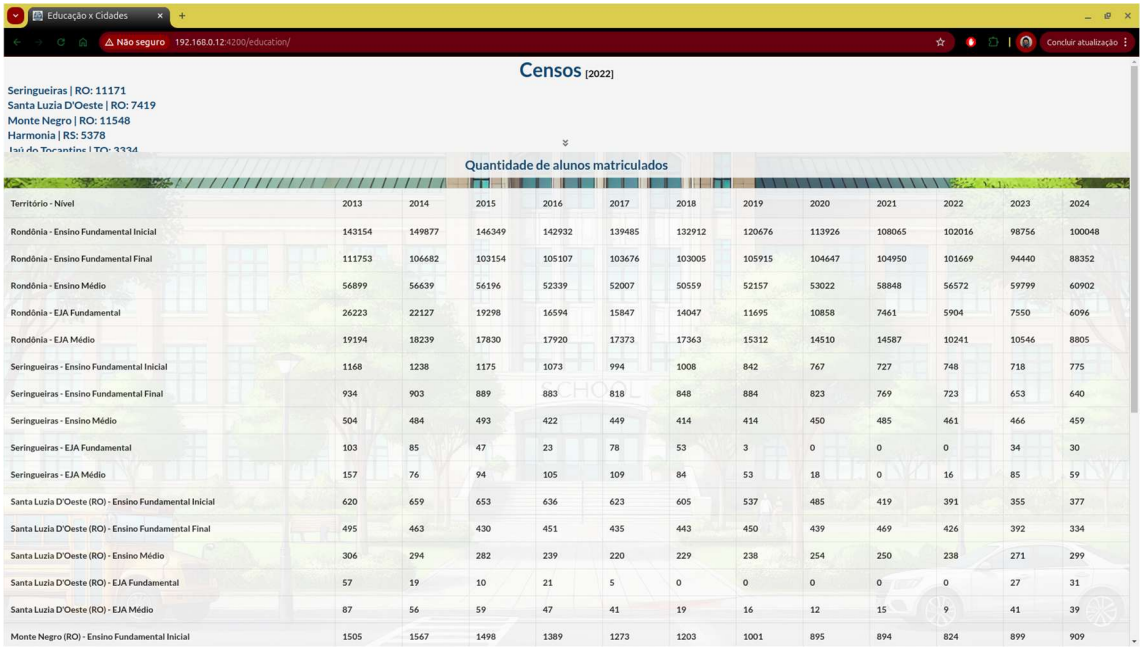
6.5.1.1 Página de Educação

A página de educação foi concebida para exibir a evolução da quantidade de alunos matriculados em diferentes níveis de ensino, como Ensino Fundamental (Inicial e Final), Ensino Médio e Educação de Jovens e Adultos (EJA). A tabela dinâmica apresentada na **Figura 24** organiza os dados por município e por ano, permitindo que o usuário visualize as mudanças nas matrículas ao longo do tempo.

Um dos destaques desta página é o uso de cores e estilizações que tornam os dados mais legíveis, enquanto os filtros permitem que o usuário selecione as cidades de interesse, focando em regiões específicas. Esse design facilita a identificação de tendências e possíveis quedas ou aumentos no acesso à educação nas capitais analisadas. Além disso, a escolha de uma imagem de fundo temática reforça o caráter educacional da página, proporcionando um visual coeso e agradável.

SOLICITAÇÃO ELETRÔNICA nº 2942/2025 PROT/CMC (1649851) 23042.002662/2025-90 pg. 81

Figura 24 – Página de educação **Desktop**



Território - Nível	2013	2014	2015	2016	2017	2018	2019	2020	2021	2022	2023	2024
Rondônia - Ensino Fundamental Inicial	143154	149877	146349	142932	139485	132912	120676	113926	108065	102016	98756	100048
Rondônia - Ensino Fundamental Final	111753	106682	103154	105107	103676	103005	105915	104647	104950	101669	94440	88352
Rondônia - Ensino Médio	56899	56639	56196	52339	52007	50559	52157	53022	58848	56572	59799	60902
Rondônia - EJA Fundamental	26223	22127	19298	16594	15847	14047	11695	10858	7461	5904	7550	6096
Rondônia - EJA Médio	19194	18239	17830	17920	17373	17363	15312	14510	14587	10241	10546	8805
Seringueiras - Ensino Fundamental Inicial	1168	1238	1175	1073	994	1008	842	767	727	748	718	775
Seringueiras - Ensino Fundamental Final	934	903	889	883	818	848	884	823	769	723	653	640
Seringueiras - Ensino Médio	504	484	493	422	449	414	414	450	485	461	466	459
Seringueiras - EJA Fundamental	103	85	47	23	78	53	3	0	0	0	34	30
Seringueiras - EJA Médio	157	76	94	105	109	84	53	18	0	16	85	59
Santa Luzia D'Oeste (RO) - Ensino Fundamental Inicial	620	659	653	636	623	605	537	485	419	391	355	377
Santa Luzia D'Oeste (RO) - Ensino Fundamental Final	495	463	430	451	435	443	450	439	469	426	392	334
Santa Luzia D'Oeste (RO) - Ensino Médio	306	294	282	239	220	229	238	254	250	238	271	299
Santa Luzia D'Oeste (RO) - EJA Fundamental	57	19	10	21	5	0	0	0	0	0	27	31
Santa Luzia D'Oeste (RO) - EJA Médio	87	56	59	47	41	19	16	12	15	9	41	39
Monte Negro (RO) - Ensino Fundamental Inicial	1505	1567	1498	1389	1273	1203	1001	895	894	824	899	909

Fonte: Próprio Autor

A **Figura 25** apresenta uma análise dos dados educacionais relacionados à média de alunos matriculados no Ensino Médio em três grandes cidades brasileiras: Manaus, Rio de Janeiro e São Paulo. O objetivo dessa análise foi avaliar o impacto da pandemia de COVID-19 sobre o número de matrículas antes e depois de 2020, destacando as variações em cada localidade.

Antes da pandemia, a média de alunos matriculados no Ensino Médio em Manaus era de **95.912**. Durante o ano de 2020, houve um ligeiro aumento para **99.028** alunos matriculados. No entanto, após 2020, essa média caiu para **88.761**, representando um delta de **-7,46%**, evidenciando uma redução significativa no número de matrículas.

No Rio de Janeiro, o número médio de matrículas antes da pandemia era de **153.044**, subindo para **172.381** em 2020. Contudo, após 2020, houve uma queda para **148.624**, resultando em uma variação de **-2,89%**. Esse decréscimo, embora menor do que em Manaus, indica um impacto negativo na continuidade das matrículas.

São Paulo apresentou o maior número absoluto de alunos matriculados no Ensino Médio antes da pandemia, com uma média de **363.194**. Em 2020, esse número caiu para **292.190**, mas recuperou-se parcialmente após 2020, alcançando uma média de **309.121**. Ainda assim, a diferença acumulada resultou em um delta de **-14,89%**, a maior variação percentual entre as cidades analisadas.

Esses resultados revelam um padrão de queda generalizada nas matrículas no Ensino Médio após a pandemia, com variações de impacto dependendo da região. Manaus e São Paulo sofreram os maiores declínios proporcionais, enquanto o Rio de Janeiro apresentou uma redução mais moderada. Esses dados destacam a necessidade de políticas públicas voltadas para a recuperação educacional, especialmente em cidades com maiores quedas, onde o impacto foi mais expressivo.

Figura 25 – Média de alunos matriculados nas cidades de Manaus, Rio de Janeiro e São Paulo

	Média Alunos matriculados antes da pandemia (2020)	Alunos matriculados em 2020	Média Alunos matriculados depois de 2020	Delta
Manaus - Ensino Médio	95.912	99.028	88.761	-7,46%
Rio de Janeiro (RJ) - Ensino Médio	153.044	172.381	148.624	-2,89%
São Paulo (SP) - Ensino Médio	363.194	292.190	309.121	-14,89%

Fonte: Próprio Autor

6.5.1.2 Página de Renda

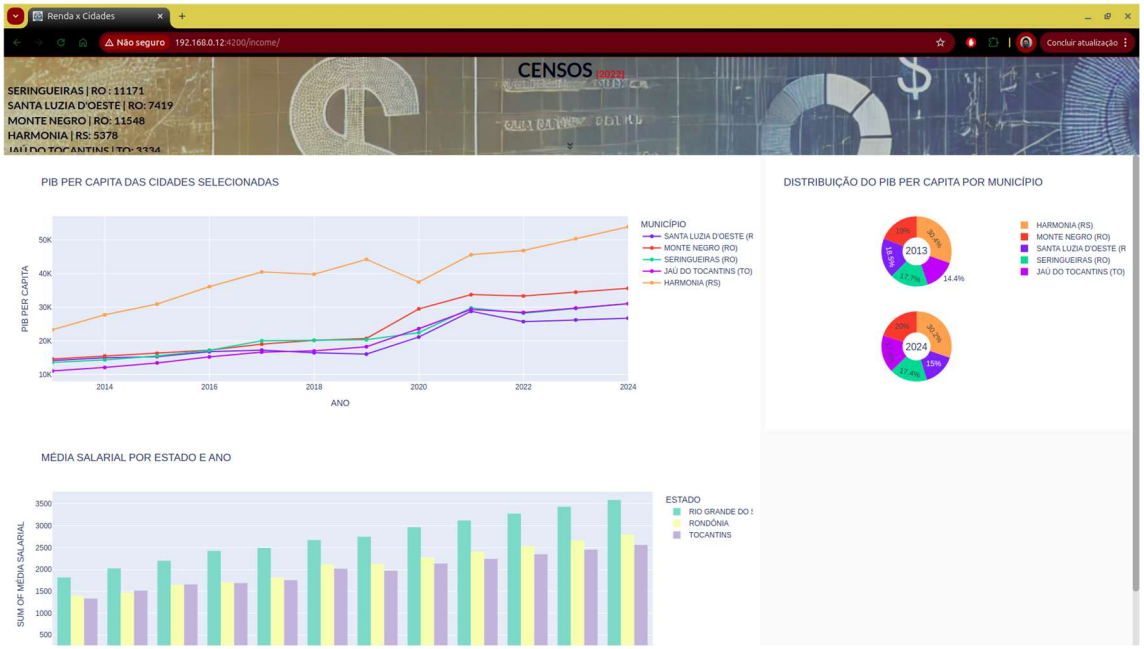
Na página de renda, o foco foi apresentar dados relacionados ao PIB per capita e à média salarial dos estados das cidades selecionadas. A **Figura 26** demonstra o uso de gráficos de linha, gráficos de pizza e gráficos de barras para exibir essas informações de maneira compreensível e comparativa.

O gráfico de linha apresenta a evolução do PIB per capita ao longo dos anos, permitindo identificar quais cidades tiveram maior crescimento econômico no período analisado. Já os gráficos de pizza oferecem uma visão clara da distribuição do PIB entre os municípios, destacando as proporções ao

longo de diferentes anos. Por fim, o gráfico de barras exibe a média salarial por estado e ano, permitindo comparações rápidas entre as localidades.

O layout dessa página foi pensado para facilitar a interpretação visual, com uma combinação de gráficos interativos e filtros que permitem ao usuário ajustar os dados de acordo com suas necessidades. A imagem de fundo foi cuidadosamente escolhida para reforçar o tema econômico, criando um ambiente visual coerente.

Figura 26 – Página de renda **Desktop**



Fonte: Próprio Autor

A **Figura 27** apresenta uma visão abrangente da evolução econômica das cidades de Manaus, Rio de Janeiro e São Paulo, com foco no PIB per capita e na média salarial por estado. A análise permite identificar tendências de crescimento e disparidades econômicas entre as localidades ao longo do período de 2013 a 2024.

O **gráfico de linha**, que detalha o PIB per capita das cidades, confirma São Paulo como líder econômico, registrando valores significativamente superiores aos de Rio de Janeiro e Manaus. Em 2024, o PIB per capita de São Paulo ultrapassa os **77 mil reais**, enquanto o Rio de Janeiro se aproxima de **63 mil reais** e Manaus permanece abaixo de **55 mil reais**, apesar de apresentar

crescimento linear ao longo do tempo. Isso reforça o papel de São Paulo como o maior polo econômico do Brasil, mas também destaca a desigualdade econômica entre as regiões.

Os **gráficos de pizza**, que mostram a distribuição percentual do PIB per capita por município, ilustram a ampliação da participação de São Paulo, que sobe de **35,1%** em 2013 para **40%** em 2024. Enquanto isso, o Rio de Janeiro apresenta uma ligeira redução percentual, passando de **26,8%** para **27,7%**, e Manaus mantém uma participação modesta, com crescimento pouco expressivo.

No **gráfico de barras**, que apresenta a média salarial por estado, percebe-se uma recuperação significativa no Rio de Janeiro e em São Paulo após 2020. Em 2013, o Rio de Janeiro registrava uma média salarial de **1.764 reais**, São Paulo de **2.124 reais** e o Amazonas de **1.501 reais**. Em 2024, os valores subiram para **3.907 reais** no Rio de Janeiro e **4.065 reais** em São Paulo, indicando um crescimento expressivo. Por outro lado, o Amazonas, apesar de também apresentar crescimento, subiu para apenas **2.093 reais**, demonstrando um avanço consideravelmente menor.

Conclusões

1. **Divergência econômica crescente:** A análise da média salarial mostra que a diferença entre São Paulo e o Amazonas, assim como entre o Rio de Janeiro e o Amazonas, aumentou ao longo dos anos. Enquanto São Paulo e Rio de Janeiro tiveram crescimentos robustos, o Amazonas apresentou sinais de estagnação relativa, resultando em uma ampliação das desigualdades salariais.

2. **Boas recuperações em São Paulo e Rio de Janeiro:** Ambos os estados demonstraram resiliência econômica, com recuperação expressiva após 2020. Esses dados indicam políticas ou estruturas econômicas mais robustas em comparação ao Amazonas.

3. **Estagnação no Amazonas:** O crescimento lento do Amazonas, tanto em termos de PIB per capita quanto de média salarial, reforça a necessidade de políticas específicas para estimular o desenvolvimento

econômico da região, especialmente em comparação aos outros estados.

4. **Desigualdades regionais acentuadas:** Ao longo dos anos, as diferenças econômicas entre os estados analisados se tornaram mais evidentes, principalmente no que diz respeito aos salários. Em 2013, a diferença entre São Paulo e o Amazonas era de **623 reais**, enquanto em 2024 essa diferença aumentou para **1.972 reais**.

Esses resultados enfatizam a importância de estratégias voltadas para o desenvolvimento regional equilibrado, com atenção especial às áreas que apresentam crescimento mais lento, como o Amazonas. A análise também destaca o papel das economias mais fortes, como as de São Paulo e Rio de Janeiro, em impulsionar o crescimento econômico do país, mas reforça a necessidade de reduzir as disparidades entre as regiões.

Figura 27 – Renda nas cidades de Manaus, Rio de Janeiro e São Paulo e seus estados

Fonte: Próprio Autor

6.5.1.3 Página de Saúde

A página de saúde, ilustrada na **Figura 28**, apresenta informações sobre a quantidade de ocorrências relacionadas a causas de morte específicas, como infarto agudo do miocárdio, insuficiência cardíaca e doenças pulmonares. O objetivo principal desta página é fornecer ideias ou pensamentos sobre os principais desafios de saúde pública enfrentados pelas cidades analisadas.

Os gráficos apresentados incluem séries temporais para monitorar a evolução das ocorrências por ano e um gráfico de barras que destaca as cinco principais causas de morte em cada cidade. Essa abordagem facilita a identificação de padrões críticos, como o aumento ou redução de certas condições ao longo do tempo, permitindo uma análise mais direcionada e estratégica.

O design da página também foi planejado para ser visualmente atraente e funcional. A imagem de fundo remete ao tema da saúde, enquanto os elementos interativos permitem que o usuário explore os dados de forma detalhada, selecionando cidades ou causas específicas.

Figura 28 – Página de saúde **Desktop**

Fonte: Próprio Autor

A **Figura 29** apresenta um gráfico com a quantidade de ocorrências de suicídio por enforcamento/sufocamento em três cidades brasileiras: **Manaus (AM)**, **Rio de Janeiro (RJ)** e **São Paulo (SP)**, ao longo dos anos de 2014 a 2024. Os dados revelam diferenças marcantes entre as cidades, indicando padrões preocupantes em Manaus, especialmente quando comparados ao número de habitantes dessa cidade.

Em 2024, embora Manaus tenha uma população significativamente menor em relação ao Rio de Janeiro e São Paulo (como indicado pelo censo na parte superior da figura), a cidade apresenta a maior quantidade de ocorrências de suicídio por enforcamento, superando tanto São Paulo quanto o Rio de Janeiro. Esse dado é alarmante, pois demonstra que, proporcionalmente, o problema é mais grave em Manaus, possivelmente indicando uma necessidade urgente de políticas públicas voltadas para a saúde mental na região.

Tendências ao Longo do Tempo

São Paulo (SP): A cidade apresenta os números absolutos mais altos no início da análise (2014), com um pico próximo de **300 ocorrências** em 2015. Entretanto, nota-se uma redução significativa nos anos seguintes, chegando a valores abaixo de **100 ocorrências** em 2024.

Rio de Janeiro (RJ): O Rio de Janeiro demonstra um crescimento gradual nas ocorrências até 2020, com um pico logo após. A partir de então, há uma redução moderada, atingindo valores menores de ocorrências aos de Manaus e São Paulo em 2024.

Manaus (AM): Apesar de apresentar números absolutos mais baixos em comparação com as outras cidades no início da análise, Manaus mostra um aumento consistente nas ocorrências ao longo dos anos, ultrapassando o Rio de Janeiro e São Paulo em 2024. Essa tendência torna-se ainda mais preocupante considerando a menor população da cidade.

Conclusões

Gravidade em Manaus: O número de ocorrências de suicídio por enforcamento em Manaus em 2024, mesmo com uma população significativamente menor, é um indicativo de que a cidade enfrenta desafios específicos relacionados à saúde mental. Fatores como desigualdade social, acesso limitado a serviços de saúde mental e questões culturais podem estar contribuindo para esse quadro.

Redução em São Paulo e Rio de Janeiro: Ambas as cidades mostram um declínio nas ocorrências após 2020, possivelmente indicando a eficácia de políticas públicas ou intervenções relacionadas à saúde mental. Esse

SOLICITAÇÃO ELETRÔNICA nº 2942/2025 PROT/CMC (1649851) 23042.002662/2025-90 pg. 88

progresso é positivo, mas deve ser monitorado para garantir que os números continuem diminuindo.

Atenção às disparidades regionais: O caso de Manaus evidencia a importância de considerar as disparidades regionais ao planejar políticas públicas voltadas para a saúde mental. Investimentos em campanhas de conscientização, capacitação de profissionais de saúde e maior disponibilidade de serviços especializados são essenciais para lidar com essa questão.

Figura 29 – Suicídio por enforcamento nas cidades de Manaus, Rio de Janeiro e São Paulo

Fonte: Próprio Autor

7 CONCLUSÃO

O presente trabalho teve como objetivo analisar dados socioeconômicos das cidades brasileiras e desenvolver um sistema que auxilie gestores públicos e outros interessados a entender e melhorar a competitividade urbana. Para alcançar esse objetivo, foi criada uma plataforma robusta que integra tecnologias modernas de desenvolvimento de software, análise de dados e visualização interativa, permitindo o acesso a informações precisas e contextualizadas sobre saúde, educação, renda e outros indicadores.

Durante o desenvolvimento do sistema, foi possível mapear dados de fontes confiáveis, como IBGE, PNUD e DATASUS, tratando-os com ferramentas avançadas, como Pandas e Dask. Esse processo garantiu que os dados fossem limpos, normalizados e organizados em um formato adequado para análises complexas. A integração entre o backend, desenvolvido em **Flask** e otimizado com **Cython**, e o frontend, estruturado com tecnologias como **Tailwind CSS**, **Plotly** e **Semantic UI**, resultou em um sistema eficiente, escalável e intuitivo.

A análise de índices socioeconômicos se mostrou uma ferramenta essencial para compreender a competitividade entre as capitais brasileiras, permitindo identificar padrões e tendências que afetam o desenvolvimento urbano. No entanto, a simples observação dos dados não é suficiente para promover melhorias; é necessário que gestores públicos e planejadores urbanos utilizem essas informações para embasar políticas estratégicas, otimizando investimentos em educação, saúde e infraestrutura. O sistema desenvolvido neste trabalho contribui para essa análise, oferecendo uma plataforma interativa que pode auxiliar no monitoramento e na formulação de estratégias voltadas para o crescimento sustentável das cidades brasileiras.

Além disso, a utilização de designs gerados por inteligência artificial para inspiração na construção das interfaces demonstrou a capacidade de alinhar inovação e funcionalidade, contribuindo para uma experiência de usuário moderna e impactante. O sistema foi projetado para ser responsivo, permitindo o uso em diferentes dispositivos e ampliando seu alcance.

SOLICITAÇÃO ELETRÔNICA nº 2942/2025 PROT/CMC (1649851) 23042.002662/2025-90 pg. 90

Os resultados obtidos confirmam que a análise e a visualização interativa de dados são ferramentas poderosas para identificar padrões e propor estratégias de melhoria. O sistema não apenas facilitou o entendimento dos dados socioeconômicos, mas também fornece ideias relevantes para a formulação de políticas públicas mais eficientes e inclusivas. A modularidade e escalabilidade do sistema permitem que ele seja expandido para incluir novas funcionalidades ou abarcar outros contextos, como regiões específicas ou outros países.

Como limitações, destaca-se a dependência de atualizações periódicas nas bases de dados utilizadas, além de eventuais dificuldades na obtenção de dados mais granulares. No entanto, essas limitações não comprometem a relevância do sistema, mas reforçam a importância de manter um ciclo contínuo de atualização e melhoria.

O desenvolvimento deste trabalho também abriu possibilidades para futuros estudos e aprimoramentos. Entre eles, a implementação de algoritmos de aprendizado de máquina mais avançados para prever tendências nos indicadores e a inclusão de ferramentas de simulação para cenários urbanos. Outra linha de evolução é o desenvolvimento de **APIs** públicas que permitam a integração com outros sistemas e ampliem a utilidade do sistema para diferentes áreas.

Conclui-se que o sistema desenvolvido alcançou os objetivos propostos, sendo uma ferramenta eficaz para análise e visualização de dados socioeconômicos. Mais do que apresentar informações, o sistema contribui para a tomada de decisões estratégicas, promovendo uma visão integrada das cidades brasileiras e auxiliando no planejamento urbano voltado para o desenvolvimento sustentável e inclusivo.

SOLICITAÇÃO ELETRÔNICA nº 2942/2025 PROT/CMC (1649851) 23042.002662/2025-90 pg. 91

8 CRONOGRAMA DE EXECUÇÃO

Para execução do presente software especificado foi definido o seguinte cronograma:

Quadro 1 – Cronograma das atividades para desenvolvimento

ATIVIDADES	CRONOGRAMA	
	INÍCIO	TÉRMINO
Levantamento de Requisitos	05/09/2024	05/10/2024
Extração, Tratamento e Carregamento (ETL)	05/10/2024	19/10/2024
Desenvolvimento do Backend	19/10/2024	01/12/2024
Desenvolvimento do Frontend	01/12/2024	31/12/2024
Desenvolvimento dos Painéis Interativos	01/01/2025	11/01/2025
Testes e Validação	11/01/2025	20/01/2025
Implantação e Lançamento	21/01/2025	25/01/2025

REFERÊNCIAS

Al-Masri, A., & Saad, M. (2015). Smart city ranking: An efficient tool for decision making. *Smart and Sustainable Built Environment*, 4(3), 195-210.

Angelidou, M. (2017). Smart city policies: A spatial approach. *Cities*, 69, 94-101.
Anthopoulos, L. G. (2017). Understanding smart cities: A framework for classifying projects in Europe's smart urban future. *Journal of Urban Technology*, 24(1), 3-27.

Batty, M., Axhausen, K. W., Giannotti, F., Pozdnoukhov, A., Bazzani, A., Wachowicz, M., ... & Portugali, Y. (2012). Smart cities of the future. *The European Physical Journal Special Topics*, 214(1), 481-518.

BEHZAD, Ehsan et al. *Cython: C-extensions for Python*. 2023. Disponível em: https://cython.org/. Acesso em: 24 jan. 2025.

Caragliu, A., Del Bo, C., & Nijkamp, P. (2011). Smart cities in Europe. *Journal of Urban Technology*, 18(2), 65-82.

Carvalho, M. S. (2017). *Metodologia da Pesquisa: o que é, como se faz.* São Paulo: Papirus Editora.

DASK DEVELOPMENT TEAM. *Dask: Parallel computing with task scheduling*. 2016. Disponível em: https://dask.org/. Acesso em: 24 jan. 2025.

Draper, N. R., & Smith, H. (1998). *Applied Regression Analysis* (3rd ed.). Wiley-Interscience.

FLANAGAN, David. *JavaScript: The Definitive Guide*. 7. ed. Sebastopol: O'Reilly Media, 2020. Disponível em: https://developer.mozilla.org/pt-BR/docs/Web/JavaScript. Acesso em: 24 jan. 2025.

Giffinger, R., Fertner, C., Kramar, H., Kalasek, R., Pichler-Milanović, N., & Meijers, E. (2007). Smart cities: Ranking of European medium-sized cities. *Centre of Regional Science (SRF)*.
Hollands, R. G. (2008). Will the real smart city please stand up? *City*, 12(3), 303-320.

gov.br. *Sistema de Informação sobre Mortalidade – SIM*. 2022. Disponível em: https://dados.gov.br/dados/conjuntos-dados/sim-1979-2019. Acesso em: 24 jan. 2025.

GRINBERG, Miguel. *Flask Web Development: Developing Web Applications with Python*. 2. ed. Sebastopol: O'Reilly Media, 2018. Disponível em: https://flask.palletsprojects.com/. Acesso em: 24 jan. 2025.

GuiaDev. *Camadas*. 2021.Disponível em: https://guia.dev/pt/pillars/software-architecture/layers.html. Acesso em: 24 jan. 2025.

Hair, J. F., Black, W. C., Babin, B. J., & Anderson, R. E. (2019). *Multivariate data analysis.* Cengage Learning.
HARRIS, Charles R. et al. *Array programming with NumPy*. Nature, v. 585, n. 7825, p. 357-362, 2020. Disponível em: https://numpy.org/. Acesso em: 24 jan. 2025.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning: data mining, inference, and prediction.* Springer Science & Business Media.

Hecht-Nielsen, R. (1992). *Theory of the backpropagation neural network*. Neural Networks, 6(1), 25-29.

IBGE. *PNAD Contínua - Pesquisa Nacional por Amostra de Domicílios Contínua*. 2012. Disponível em: https://www.ibge.gov.br/estatisticas/sociais/trabalho/17270-pnad-continua.html?=&t=downloads. Acesso em: 24 jan. 2025.

IBGE. *Censo Demográfico 2022 - Indígenas: Principais características das pessoas e dos domicílios, por situação urbana ou rural do domicílio - Resultados do Universo*. 2022. Disponível em: https://sidra.ibge.gov.br/pesquisa/censo-demografico/demografico-2022/universo-indigenas-caracteristicas-pessoas-e-domicilios-situacao-urbana-ou-rural. Acesso em: 24 jan. 2025.

INEP. *Resultados*. 2020. Disponível em: https://www.gov.br/inep/pt-br/areas-de-atuacao/pesquisas-estatisticas-e-indicadores/censo-escolar/resultados. Acesso em: 24 jan. 2025.

KLINGE, Thomas et al. *Project Jupyter: Open tools for interactive computing*. 2015. Disponível em: https://jupyter.org/. Acesso em: 24 jan. 2025.

Komninos, N. (2008). Intelligent cities: towards interactive and global innovation environments. *International Journal of Innovation and Regional Development*, 1(4), 337-355.

PEDREGOSA, Fabian et al. *Scikit-learn: Machine Learning in Python*. Journal of Machine Learning Research, v. 12, p. 2825-2830, 2011. Disponível em: https://scikit-learn.org/. Acesso em: 24 jan. 2025.

PLOTLY TECHNOLOGIES INC. *Plotly: The Frontend for ML and Data Science Models*. 2015. Disponível em: https://plotly.com/. Acesso em: 24 jan. 2025.

Quinlan, J. R. (1986). *Induction of decision trees. Machine Learning*, 1(1), 81-106.

Semantic UI. *Semantic User Interface Framework*. Disponível em: https://semantic-ui.com/. Acesso em: 24 jan. 2025.

Sommerville, Ian. *Engenharia de Software, 9ª Edição. Pearson Education*, 2011. Seção 2.1.1 O Modelo em Cascata

TAILWIND LABS. *Tailwind CSS Documentation*. 2023. Disponível em: https://tailwindcss.com/. Acesso em: 24 jan. 2025.

THE PANDAS DEVELOPMENT TEAM. *pandas: Python Data Analysis Library*. 2012. Disponível em: https://pandas.pydata.org/. Acesso em: 24 jan. 2025.

TORVALDS, Linus; DIAMOND, David. *Just for fun: The story of an accidental revolutionary*. New York: HarperCollins, 2001. Disponível em: https://www.kernel.org/. Acesso em: 24 jan. 2025.

VAN ROSSUM, Guido; DRAKE, Fred L. *Python 3 Reference Manual*. Amsterdam: Centrum voor Wiskunde en Informatica, 2001. Disponível em: https://www.python.org/doc/. Acesso em: 24 jan. 2025.

Xu, R., & Wunsch, D. (2005). *Survey of clustering algorithms*. IEEE Transactions on Neural Networks, 16(3), 645-678.