

**UNIVERSIDADE DE CAXIAS DO SUL
ÁREA DO CONHECIMENTO DE CIÊNCIAS EXATAS E
ENGENHARIAS**

LUÍS PEDRO TORESAN BET

**DESENVOLVIMENTO DE UMA PLATAFORMA MOBILE PARA O
PROJETO CIDADES DO CONHECIMENTO**

CAXIAS DO SUL

2026

LUÍS PEDRO TORESAN BET

**DESENVOLVIMENTO DE UMA PLATAFORMA MOBILE PARA O
PROJETO CIDADES DO CONHECIMENTO**

Trabalho de Conclusão de Curso
apresentado como requisito parcial
à obtenção do título de Bacharel em
Ciência de Computação na Área do
Conhecimento de Ciências Exatas e
Engenharias da Universidade de Caxias
do Sul.

CAXIAS DO SUL

2026

LUÍS PEDRO TORESAN BET

**DESENVOLVIMENTO DE UMA PLATAFORMA MOBILE PARA O
PROJETO CIDADES DO CONHECIMENTO**

Trabalho de Conclusão de Curso
apresentado como requisito parcial
à obtenção do título de Bacharel em
Ciência de Computação na Área do
Conhecimento de Ciências Exatas e
Engenharias da Universidade de Caxias
do Sul.

Aprovado(a) em 10/07/2026

BANCA EXAMINADORA

Daniel Luís Notari
Universidade de Caxias do Sul - UCS

Helena Graziottin Ribeiro
Universidade de Caxias do Sul - UCS

Leonardo Pellizzoni
Universidade de Caxias do Sul - UCS

*Este trabalho é dedicado a todos que, mesmo
nos momentos de dúvida, escolheram seguir em
frente e perseveraram.*

AGRADECIMENTOS

Quero aproveitar este momento para expressar minha imensa gratidão à minha família que sempre esteve ao meu lado sem hesitar, para me apoiar nessa fase tão significativa da minha vida. Mesmo nos momentos difíceis, o amor e o carinho de vocês me deram a força necessária para seguir em frente. À minha irmã, que mesmo indiretamente estava ali presente para ouvir quando era preciso. À minha mãe, por me dar todo suporte necessário e principalmente por acreditar em mim. Ao meu pai, por estar sempre de pé ao meu lado me dando forças. À minha avó que sempre foi um exemplo para mim, de superação e de humildade, pessoa que sempre esteve disposta a ajudar e desconstrair. E especialmente ao meu querido avô que sempre esteve próximo curioso e interessado em tudo que eu tinha para falar sobre a minha vida pessoal, profissional e acadêmica. Sei que isto nunca irá chegar até ele, mas onde quer que ele esteja agora, sei que todo esse caminho foi possível graças a ele.

Aos meus amigos e colegas, que me incentivaram nos momentos em que pensei em desistir, meu agradecimento vem do fundo do coração pela força, apoio e amizade. É complicado citar todos os nomes, pois cada um de vocês teve um papel fundamental na minha jornada. No entanto, quero fazer um agradecimento especial ao Alisson, ao Gustavo, ao Otho, ao Otávio e a Thaís, que estiveram comigo do começo ao fim da minha graduação, todas as indas e vindas, encontros e desencontros no campus me deram forças durante todo esse período. Saibam que, se não fosse por vocês, nada disso teria sido possível. E fora do ambiente acadêmico, sou imensamente grato ao Mateus e a Alice, que desde muito cedo sempre estiveram ao meu lado me apoiando e me incentivando a ser uma pessoa melhor, permitindo que eu voltasse a acreditar em mim mesmo quando tudo parecia quebrado. Agradeço também à minha namorada, Maria Carolina, por ter me incentivado e apoiado durante o desenvolvimento da aplicação.

Agradeço também ao Professor Daniel Luís Notari, que foi fundamental na orientação deste trabalho. Desde o início da minha trajetória acadêmica, ele me motivou a seguir em frente e sempre se dispôs a esclarecer qualquer dúvida que surgisse, me guiando com sabedoria e paciência. Tenho certeza de que, além de orientador e professor, construímos uma amizade valiosa.

*“Aqueles que passam por nós, não vão sós, não nos deixam sós. Deixam um pouco de si,
levam um pouco de nós.”*
Antoine de Saint-Exupéry

RESUMO

A plataforma Cidades do Conhecimento é uma iniciativa voltada à disponibilização e análise de indicadores urbanos relacionados ao contexto das cidades inteligentes. E este Trabalho de Conclusão de Curso teve como objetivo projetar e desenvolver uma versão mobile para a plataforma que hoje está disposta na *web*, com o propósito de ampliar o acesso aos dados disponibilizados pelo projeto. Para isso, foi realizado um estudo sobre desenvolvimento mobile, usabilidade, responsividade e visualização de dados, orientando a definição da navegação, do layout e das formas de apresentação das informações. Também foram selecionadas as tecnologias, frameworks e bibliotecas utilizadas na implementação da aplicação. Como resultado, foi desenvolvida uma aplicação mobile que disponibiliza os dados e funcionalidades da plataforma em dispositivos móveis, permitindo a consulta, análise e comparação de indicadores urbanos.

Palavras-chave: Cidades Inteligentes. Programação Mobile. Visualização de Dados. Indicadores Urbanos.

LISTA DE FIGURAS

Figura 1 – Exemplo de gráfico de linha. Fonte: Geração própria.	19
Figura 2 – Exemplo de gráfico de barra. Fonte: Geração própria.	20
Figura 3 – Exemplo de histograma. Fonte: Geração própria.	20
Figura 4 – Exemplo de gráfico de caixa (boxplot). Fonte: Geração própria.	20
Figura 5 – Exemplo de gráfico de dispersão. Fonte: Geração própria.	21
Figura 6 – Exemplo de gráfico de bolhas. Fonte: Geração própria.	21
Figura 7 – Exemplo de mapa de árvores (treemap).	21
Figura 8 – Exemplo de mapa de calor (heatmap). Fonte: Geração própria.	22
Figura 9 – Exemplo de Power BI demonstrando uma matriz de indicadores dos últimos cinco anos, retirado do projeto Cidades do Conhecimento.	22
Figura 10 – Página de ranking do Power BI demonstrando gráficos de diferentes indicadores do modelo <i>Smart City</i> , desenvolvido para o projeto Cidades do Conhecimento.	23
Figura 11 – Exemplo de Tableau.	23
Figura 12 – Exemplo de cada tipo de visualização e interação.	24
Figura 13 – Exemplo de arquitetura de duas camadas. Fonte: NYABUTO; MONY; MBUGUA (2024)	30
Figura 14 – Exemplo de arquitetura de três camadas. Fonte: NYABUTO; MONY; MBUGUA (2024)	30
Figura 15 – Exemplo do funcionamento de uma API. Fonte: MUHAMMAD; PAPUTUNGAN (2024)	31
Figura 16 – Página inicial primeira versão da plataforma web. Fonte: BATTISTELO (2018)	34
Figura 17 – Seleção de cidades da primeira versão da plataforma web. Fonte: BATTISTELO (2018)	35
Figura 18 – Sistema da primeira versão da plataforma web. Fonte: BATTISTELO (2018) .	36
Figura 19 – Página inicial da segunda versão da plataforma web. Fonte: MARIA (2024) .	37
Figura 20 – Seleção de cidades da segunda versão da plataforma web. Fonte: MARIA (2024)	38
Figura 21 – Gráfico comparativo da segunda versão da plataforma web. Fonte: MARIA (2024)	38
Figura 22 – Diagrama de casos de uso da aplicação. Fonte: Geração própria.	42
Figura 23 – Tela Inicial da aplicação. Fonte: Geração própria.	44
Figura 24 – Tela Inicial. Fonte: Geração própria.	45
Figura 25 – Tela de Modelos. Fonte: Geração própria.	45
Figura 26 – Tela de Dimensão. Fonte: Geração própria.	46
Figura 27 – Tela de Comparativo. Fonte: Geração própria.	47
Figura 28 – Tela de Resultado. Fonte: Geração própria.	48

Figura 29 – Arquitetura para o projeto. Fonte: Geração própria.	49
Figura 30 – Modelo de dados proposto para a plataforma. Fonte: MARIA (2024).	52
Figura 31 – Diagrama com a ideia geral da aplicação.	54
Figura 32 – Telas de abertura, página inicial, indicadores, dimensões e informações sobre o projeto. Fonte: Geração própria.	68
Figura 33 – Fluxo de comparação: seleção de cidades, gráfico radar, médias e detalhamento por dimensão. Fonte: Geração própria.	70
Figura 34 – Gráfico gerado pela aplicação web. Fonte: < https://cidadesinteligentes.agrocitylivinglab.com.br/comparar_cidades_resultado/ >.	71
Figura 35 – Gráfico gerado pela aplicação mobile. Fonte: Geração própria.	72
Figura 36 – Dados de telecomunicações na aplicação web. Fonte: < https://cidadesinteligentes.agrocitylivinglab.com.br/comparar_cidades_resultado/ >.	72
Figura 37 – Dados de telecomunicações na aplicação mobile. Fonte: Geração própria.	73
Figura 38 – Dados de economia na aplicação web. Fonte: < https://cidadesinteligentes.agrocitylivinglab.com.br/comparar_cidades_resultado/ >.	73
Figura 39 – Dados de economia na aplicação mobile. Fonte: Geração própria.	73
Figura 40 – Comparação da tela de dimensões entre as versões web e mobile. Fonte: Geração própria.	75
Figura 41 – Comparação da tela de comparativos entre as versões web e mobile. Fonte: Geração própria.	76
Figura 42 – Comparação da apresentação tabular dos indicadores entre as versões web e mobile.	77
Figura 43 – Tela de exibição das médias por dimensão da versão <i>mobile</i> . Fonte: Geração própria.	77
Figura 44 – Comparação da navegação pelas dimensões entre as versões web e mobile. Fonte: Geração própria.	78

LISTA DE TABELAS

Tabela 1 – Detalhamento e síntese comparativa das versões anteriores da plataforma Cidades do Conhecimento	40
---	----

LISTA DE ABREVIATURAS E SIGLAS

AHP	Processo Analítico Hierárquico
API	Application Programming Interface (Interface de Programação de Aplicações)
CC	Cidades do Conhecimento
CCS	Construção de Cidades Sustentáveis
IDH	Índice de Desenvolvimento Humano
ISO	International Organization for Standardization (Organização Internacional de Normalização)
IoT	Internet das Coisas
LL	Living Lab
NBC	Negócios Baseado em Conhecimento
ODM	Objetivos de Desenvolvimento do Milênio
ODITT	Observação, Desenvolvimento e a Inteligência Turística e Territorial
ODS	Objetivos de Desenvolvimento Sustentável
OGD	Dados Governamentais Abertos (Open Government Data)
PIB	Produto Interno Bruto
SC	Sistemas de Capitais
UCS	Universidade de Caxias do Sul
UI	User Interface (Interface de Usuário)
UX	User Experience (Experiência do Usuário)

SUMÁRIO

1	INTRODUÇÃO	13
1.1	PROBLEMAS DE PESQUISA	15
1.2	OBJETIVOS	16
1.3	ESTRUTURA DO TRABALHO	16
2	REFERENCIAL TEÓRICO	18
2.1	Cidades Inteligentes e o Papel dos Living Labs - Ferramentas para acesso e interpretação dos dados	18
2.2	Desenvolvimento mobile	24
2.2.1	Interfaces em mobile	25
2.2.2	Abordagens de desenvolvimento	27
2.2.3	Arquitetura de mobile	29
2.3	Trabalhos correlatos	32
2.3.1	Living Labs	32
2.3.2	Citing Living Lab e a plataforma desenvolvida para o Cidades do Conhecimento	33
3	PROPOSTA DE SOLUÇÃO	41
3.1	Requisitos funcionais	41
3.2	Arquitetura da aplicação	48
3.3	Modelagem dos dados	50
4	IMPLEMENTAÇÃO DA PROPOSTA DE SOLUÇÃO	53
4.1	Ferramentas e bibliotecas utilizadas	53
4.2	Estrutura da aplicação	54
4.3	Ajustes em comparação a aplicação web	56
4.4	Atualização de bibliotecas e carregamento da aplicação	57
4.5	Implementação da Aplicação Mobile	57
4.5.1	Inicialização e autenticação	58
4.5.2	Estrutura visual e navegação	59
4.5.3	Consulta de modelos, indicadores e dimensões	59
4.5.4	Seleção de cidades para comparação	60
4.5.5	Serviço de comparação e processamento dos dados	61
4.5.6	Apresentação do resultado comparativo	64
4.5.7	Funcionalidades implementadas	66
4.6	Casos de uso da aplicação	67

4.6.1	Comparativo dos dados com a aplicação web	70
5	CONSIDERAÇÕES FINAIS	74
5.1	Adequações da Interface da Versão Mobile em Relação à Versão Web	75
5.2	Sugestões de Trabalhos Futuros	78
	REFERÊNCIAS	79

1 INTRODUÇÃO

O conceito de cidade sempre foi algo moldado ao longo do tempo e, como destaca ULTRAMARI (2019), defini-lo é uma tarefa complexa, pois envolve aspectos geográficos, sociológicos e históricos. Também destaca que uma cidade é algo além de um simples aglomerado físico ou populacional, a cidade é uma construção ou parte de um fenômeno social, moldada por meio de relações humanas, transformações no território e disputas por espaço.

Ampliando a reflexão sobre o que define uma cidade, VASCONCELOS (2015) analisou o conceito ao longo de diferentes períodos históricos. No início, entre 1810 e 1869, considerava-se cidade qualquer local que concentrasse população, meios de produção e serviços capazes de atender às necessidades básicas da sociedade. Embora o tempo tenha passado, para VASCONCELOS (2015), o conceito contemporâneo ainda carrega muitos traços dessa definição do século XVIII. A principal mudança está em que uma cidade passou a ser muito mais do que um aglomerado de pessoas, a cidade atual é entendida como a junção de funções e atividades que sustentam o convívio mútuo e o funcionamento do espaço urbano.

Com o avanço da urbanização e a complexificação das dinâmicas urbanas, surge com força o conceito de cidades inteligentes. Segundo BRUZZEGUEZ; MORAES; GUEDES (2024), esse modelo foi impulsionado pelo desenvolvimento de tecnologias como Internet das Coisas (IoT), *big data*, inteligência artificial, sensores urbanos e sistemas automatizados de controle. Essas inovações buscam não apenas modernizar o espaço urbano, mas também aprimorar a qualidade de vida, otimizar os serviços públicos e tornar mais eficiente o uso dos recursos municipais. A pesquisa de ALVES; DIAS; SEIXAS (2019) destaca que a aplicação do conceito varia conforme o contexto ao comparar cidades brasileiras e portuguesas, os autores evidenciam que fatores culturais, socioeconômicos e necessidades locais influenciam diretamente sua implementação. Apesar das particularidades, desafios recorrentes, como a infraestrutura tecnológica deficiente, ainda limitam o avanço dessas iniciativas.

O desenvolvimento e o progresso das cidades inteligentes dependem, fundamentalmente, da extração e do acesso a dados públicos. Nesse contexto, é imprescindível que esses dados sejam disponibilizados de forma acessível e transparente, ou seja, que estes sejam dados abertos. Dado aberto é toda e qualquer informação, seja ela científica ou não, que pode ser facilmente acessada, reutilizada e compartilhada livremente por qualquer pessoa, para MONTEIRO; LUCAS (2019) o conceito de dado aberto está ligado ao movimento que defende a democratização do conhecimento, com o intuito de incentivar a transparência, a reprodutibilidade de pesquisas e o avanço mais ágil da ciência. Porém o estudo de MACEDO; LEMOS (2021) destaca que a extração de dados ainda é um processo de mão dupla, pois, enquanto esses dados são essenciais para promover a melhoria da eficiência dos serviços públicos e estimular a inovação, além de facilitar a tomada de decisões, ainda persiste uma resistência significativa por parte

dos órgãos públicos quanto à liberação dessas informações, principalmente devido a questões relacionadas à segurança e à privacidade. Embora o Brasil tenha feito avanços consideráveis na implementação de políticas de dados abertos, ainda persiste um longo caminho a ser percorrido.

Nesse contexto GAO; JANSSEN; ZHANG (2023) analisaram a trajetória das pesquisas sobre dados governamentais abertos (Open Government Data - OGP) e observaram uma melhora significativa no campo. Inicialmente, GAO; JANSSEN; ZHANG (2023) demonstram que os estudos concentravam-se na abertura e disponibilização dos dados públicos como instrumento de transparência com viés democrático. Com o amadurecimento da área de OGP, o foco passou a incluir a reutilização dos dados e a geração de valor público a partir deles, explorando como cidadãos, empresas e organizações utilizam essas informações para promover inovação, participação social e desenvolvimento econômico. Por fim, GAO; JANSSEN; ZHANG (2023) enfatizam que ainda há a necessidade de avançar um pouco mais para uma abordagem orientada a sustentabilidade e a *smartness* dos dados abertos, onde a sustentabilidade envolve garantir qualidade, atualização contínua, governança adequada e integração institucional, evitando que portais de dados se tornem obsoletos ou subutilizados. Já o conceito de *smartness* refere-se ao uso estratégico e inteligente dos dados, integrando-os a tecnologias digitais, análises avançadas e sistemas de apoio à decisão, especialmente no contexto de governo digital e cidades inteligentes.

Em sua pesquisa FACHINELLI *et al.* (2023) abordaram todo processo de coleta e de como usar os indicadores em prol da sociedade, para isso FACHINELLI *et al.* (2023) mostram como os indicadores da cidade foram obtidos de fontes oficiais e selecionadas especificamente para capturar aspectos fundamentais das realidades demográficas, socioeconômicas, ambientais e de infraestrutura das cidades do Corede Serra. Esses dados foram analisados em comparação com outros municípios, o que permitiu identificar desigualdades e semelhanças regionais, bem como acompanhar a evolução dos indicadores ao longo do tempo para captar tendências e mudanças. Assim, os indicadores contribuíram para um diagnóstico detalhado da região, que serve como base para o desenvolvimento e aprimoramento de políticas, planejamento estratégico, alocação de recursos e é sustentado por evidências. Além disso, o principal objetivo ao divulgar esses dados foi de aumentar a transparência enquanto se fomenta o engajamento da comunidade na participação social no desenvolvimento local sustentável, para que possam fortalecer o conhecimento sobre sua própria região.

Um exemplo de aplicação foi o projeto SmartAirQ, apresentado por KAGINALKAR *et al.* (2022), que propôs um *framework* de governança de *big data* voltado ao monitoramento e à gestão da qualidade do ar em cidades, integrando diversas fontes de dados, como sensores IoT, estações de monitoramento, dados meteorológicos e informações de tráfego, a fim de apoiar decisões públicas mais eficazes. KAGINALKAR *et al.* (2022) estruturaram o modelo considerando aspectos técnicos, organizacionais e regulatórios. Nos aspectos técnicos destacam-se a coleta, integração, armazenamento e análise de grandes volumes de dados; nos organizacionais os papéis institucionais e a coordenação entre atores; e nos regulatórios a qualidade, a segurança e o

compartilhamento de dados. O estudo evidencia que a gestão integrada e estratégica de dados é fundamental para transformar dados brutos em insights acionáveis, contribuindo para políticas urbanas mais sustentáveis e baseadas em evidências.

1.1 PROBLEMAS DE PESQUISA

O City Living Lab é um laboratório vivo que promove a integração entre empresas, governo, academia e cidadãos, com o objetivo de desenvolver soluções voltadas ao crescimento sustentável e à inovação tecnológica na Serra Gaúcha, contemplando diversas áreas de estudo.

O projeto Cidades do Conhecimento teve como base a coleta e o tratamento de dados provenientes de diversas fontes, o que representou um desafio recorrente. O trabalho de BREGALDA (2021) contribuiu com o desenvolvimento de um método de carga semi automática de dados, reduzindo a necessidade de inserção manual. De forma complementar, o estudo de SILVA (2022) implementou um processo de automatização da coleta e carga de dados, com a criação de um banco de dados responsável pelo armazenamento das informações relacionadas à execução do automatizador. Posteriormente, RODRIGUES (2023) reuniu as duas abordagens e propôs uma solução integrada. Como os trabalhos anteriores foram desenvolvidos com estruturas e dependências distintas, havia a necessidade de preparação de ambientes separados. Para resolver essa questão, RODRIGUES (2023) implementou um container Docker com o objetivo de integrar as ferramentas desenvolvidas por SILVA (2022) e BREGALDA (2021), permitindo a unificação dos dados utilizados pelo laboratório City Living Lab.

A visualização dos dados teve início em 2018, quando BATTISTELO (2018) desenvolveu uma plataforma *web* no contexto de seu Trabalho de Conclusão de Curso. A aplicação foi projetada para apresentar os indicadores contemplados pelo projeto, que incluem: Identidade, Inteligência, Relacional, Financeiro, Investimento, Humano Individual, Humano Coletivo, Instrumental-Material e Instrumental-Intangível. Antes da criação dessa plataforma, os dados eram disponibilizados apenas por meio de planilhas, o que acarretava problemas como redundância de informações e ausência de centralização, dificultando o acesso e a análise eficiente dos indicadores. No ano de 2024, MARIA (2024) identificou limitações na plataforma original, especialmente quanto à sua abrangência, que permitia comparações apenas entre cidades da Serra Gaúcha. Com isso, realizou a readequação da aplicação *web*, estendendo seu escopo para incluir municípios de todo o país e reformulando sua estrutura para possibilitar uma visualização mais funcional dos dados.

Apesar das evoluções promovidas em MARIA (2024), a plataforma permanece disponível exclusivamente na versão *web*¹, o que restringe seu acesso a computadores e a determinados navegadores. Essa limitação evidencia um problema de pesquisa relacionado à disponibilidade e ao alcance dos dados do projeto Cidades do Conhecimento, uma vez que o consumo de in-

¹ Disponível em: <<https://cidadesinteligentes.agrocitylivinglab.com.br>>

formações em dispositivos móveis é cada vez mais predominante. Assim, questiona-se como ampliar o acesso e as possibilidades de visualização desses dados em ambientes móveis, mantendo a organização, a clareza e a eficiência na apresentação dos indicadores.

Dessa forma, este trabalho busca enfrentar essa limitação por meio do desenvolvimento de uma versão móvel da plataforma, permitindo que os dados coletados possam ser consultados de forma prática e eficiente também em dispositivos móveis, promovendo maior flexibilidade e alcance no uso das informações.

1.2 OBJETIVOS

Este trabalho tem como objetivo o desenvolvimento de uma aplicação mobile para o projeto Cidades do Conhecimento, sendo os objetivos específicos os seguintes:

1. Realizar uma análise da plataforma atual, com foco na acessibilidade dos dados em dispositivos móveis, a fim de embasar a criação de uma solução adequada.
2. Escrever uma proposta de solução que descreva as funcionalidades, estrutura e interfaces da versão móvel da plataforma, considerando as necessidades dos usuários e as tecnologias disponíveis.
3. Desenvolver uma plataforma mobile para o projeto Cidades do Conhecimento, sendo uma adaptação da plataforma web hoje existente.
4. Criar painéis e dashboards para apresentação dos dados coletados, com foco na organização e funcionalidade das informações.
5. Desenvolver interfaces gráficas compatíveis para dispositivos móveis, com estrutura adaptada à navegação e interação nesses ambientes.

1.3 ESTRUTURA DO TRABALHO

O presente trabalho está organizado da seguinte forma: o capítulo um apresenta o contexto geral da pesquisa, delineando a problemática que motivou o estudo, bem como os objetivos propostos para sua realização. Este capítulo tem como propósito situar o leitor quanto à relevância do tema e às questões que nortearão a investigação. O capítulo dois reúne os principais conceitos, teorias e estudos que fundamentam a pesquisa. Este capítulo busca oferecer embasamento teórico para a compreensão do tema, por meio da análise crítica da literatura existente. O capítulo três apresenta os protótipos de tela da aplicação e também detalha como cada elemento contribui para a solução do problema central abordado neste trabalho, evidenciando de que forma a aplicação busca atender às necessidades identificadas durante a fase de levantamento

de requisitos. O capítulo quatro demonstra tecnologias e ferramentas utilizadas para o desenvolvimento da plataforma, está disposto nele informações sobre arquitetura do projeto, trechos do código e o fluxo da aplicação. Também está disposto um breve comparativo em relação a plataforma *web*. Por fim, o capítulo cinco apresenta as considerações finais e conclusão deste trabalho de conclusão de curso.

2 REFERENCIAL TEÓRICO

Este capítulo tem como objetivo apresentar os fundamentos teóricos que embasam o desenvolvimento da aplicação proposta, abordando os principais conceitos relacionados às cidades inteligentes, visualização de dados, desenvolvimento de interfaces gráficas e tecnologias para dispositivos móveis.

2.1 CIDADES INTELIGENTES E O PAPEL DOS LIVING LABS - FERRAMENTAS PARA ACESSO E INTERPRETAÇÃO DOS DADOS

Segundo COVAS; COVAS (2021), uma cidade inteligente deve ser embasada em princípios como o equilíbrio ecológico, a inclusão social e a democracia. Para que tais objetivos sejam alcançados, é necessário adotar novas formas de regulação e governança. Ainda de acordo com os autores, a *smartificação* do território deve ir além dos limites do centro urbano, abrangendo também as áreas suburbanas e rurais. Essa abordagem visa promover uma integração efetiva entre cidade e campo, contribuindo para um desenvolvimento territorial mais equilibrado. Além disso, COVAS; COVAS (2021) enfatizam a importância de prevenir a fratura digital, combater a iliteracia digital e fomentar a criação de ecossistemas digitais que estimulem a participação cidadã em prol do desenvolvimento territorial. COVAS; COVAS (2021) defende que uma cidade só pode ser considerada verdadeiramente inteligente e criativa quando consegue integrar, de maneira equilibrada e inclusiva, os elementos tecnológicos, culturais, ecológicos e de governança.

O estudo de STEEN; BUEREN (2017) apresentou a base conceitual e estrutural para a criação de Living Labs voltados à inovação sustentável nas cidades, propondo uma metodologia que abrange desde a identificação do problema até a disseminação das soluções desenvolvidas. A partir da análise de 90 projetos na região metropolitana de Amsterdã, foram identificadas nove características essenciais desses laboratórios, como sustentabilidade urbana, co-criação, participação ativa, diversidade de atores, contexto real de uso, clareza de objetivos e estrutura organizacional bem definida. Complementando essa abordagem, SOEIRO (2021) analisou os Living Labs no contexto do projeto europeu *ROCK*, destacando seu papel na regeneração urbana e no equilíbrio das estratégias de governança. O autor enfatiza a importância da cooperação entre agentes públicos e privados e da atuação governamental como fatores determinantes para o sucesso desses ambientes, que se consolidam como espaços estratégicos para a geração e análise de dados urbanos.

Esses dados, coletados e geridos dentro de Living Labs, são fundamentais para entender e tomar decisões mais assertivas sobre as cidades e sua gerência. A visualização desses dados, então, desempenha um papel essencial, não apenas na sua análise, mas também na comunicação eficaz das informações, facilitando a interpretação e a implementação de estratégias baseadas

em evidências. Em seu estudo, SILVA (2019) apresentou os princípios essenciais da ciência de dados, com ênfase na visualização de dados como uma área fundamental de estudo. SILVA (2019) explorou como dados de diferentes tipos têm sido representados por meio de imagens, infográficos e gráficos ilustrados, destacando a relação entre os elementos de sintaxe e semântica para facilitar a compreensão de informações complexas. Segundo SILVA (2019), a utilização de representações visuais funciona como uma espécie de ferramenta cognitiva importante para apoiar a análise de dados e informações. Assim SILVA (2019) defende que a visualização de algo vai além de um simples fator estético e passa a ser algo que se transforma em uma linguagem estruturada, sendo assim essencial para o pensamento analítico.

Pesquisas na área de visualização de dados têm enfatizado a relevância da adoção de boas práticas e do uso de ferramentas adequadas para o desenvolvimento de dashboards eficientes. BACH *et al.* (2023) identificaram, a partir da análise de 144 dashboards reais, 42 padrões de design organizados em aspectos de conteúdo (tipos de dados e informações apresentadas) e de composição (estrutura visual, organização, cores e interatividade), reforçando a necessidade de diretrizes sistematizadas para apoiar o processo de criação. Já BIKAKIS (2018) evidencia os desafios relacionados à visualização de grandes volumes de dados, defendendo soluções capazes de oferecer interação em tempo real, escalabilidade e personalização. Complementarmente, GAHAR; ARFAOUI; HIDRI (2024) ressaltam a importância de interfaces intuitivas aliadas a funcionalidades avançadas, buscando equilibrar acessibilidade para usuários iniciantes e recursos analíticos mais sofisticados para usuários experientes.

Dentre as diversas técnicas de visualização de dados, destacam-se algumas das mais utilizadas, conforme os exemplos ilustrados nas figuras a seguir:

- **Gráficos de linha:** ideais para mostrar a evolução de uma variável ao longo do tempo, facilitando a visualização de tendências e padrões temporais (Figura 1).

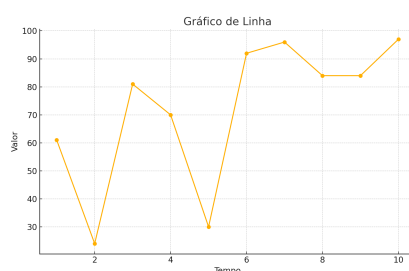


Figura 1 – Exemplo de gráfico de linha. Fonte: Geração própria.

- **Gráficos de barra:** utilizados para comparar quantidades entre diferentes categorias, onde o comprimento das barras representa a magnitude dos valores (Figura 2).

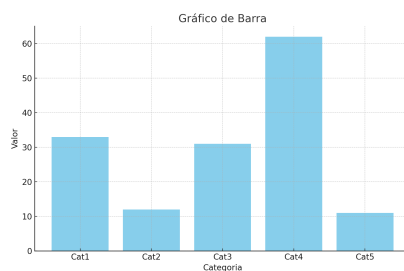


Figura 2 – Exemplo de gráfico de barra. Fonte: Geração própria.

• **Histogramas:** semelhantes aos gráficos de barra, porém focados em mostrar a distribuição de frequência de um conjunto de dados contínuos, divididos em intervalos (bins) (Figura 3).

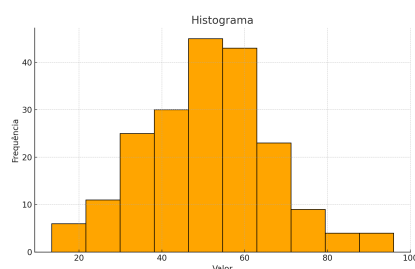


Figura 3 – Exemplo de histograma. Fonte: Geração própria.

• **Gráficos de caixa (boxplot):** representam a dispersão, mediana, quartis e possíveis outliers de um conjunto de dados, sendo muito úteis para análise estatística exploratória (Figura 4).

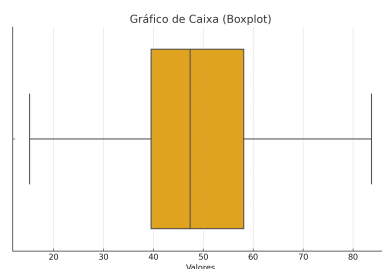


Figura 4 – Exemplo de gráfico de caixa (boxplot). Fonte: Geração própria.

• **Gráficos de dispersão (scatter plots):** exibem a relação entre duas variáveis numéricas, ajudando a identificar correlações ou padrões entre elas (Figura 5).

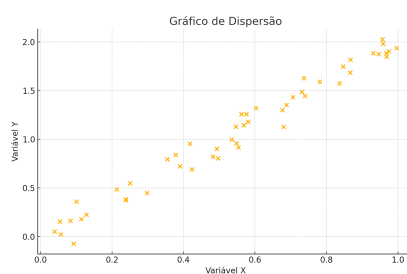


Figura 5 – Exemplo de gráfico de dispersão. Fonte: Geração própria.

• **Gráficos de bolhas:** uma variação do gráfico de dispersão que utiliza o tamanho das bolhas para representar uma terceira variável (Figura 6).

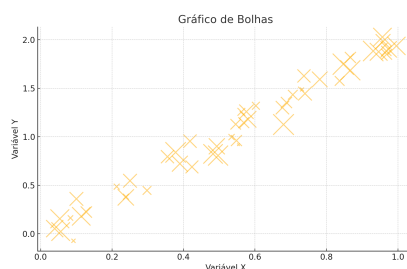


Figura 6 – Exemplo de gráfico de bolhas. Fonte: Geração própria.

• **Mapas de árvores (treemaps):** usados para representar hierarquias e proporções entre categorias, com retângulos aninhados em tamanhos proporcionais (Figura 7).

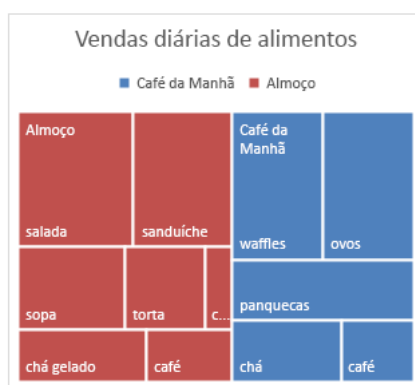


Figura 7 – Exemplo de mapa de árvores (treemap).

Fonte: Microsoft Support¹.

• **Mapas de calor (heatmaps):** representam dados em formato matricial com uso de cores para indicar intensidade ou frequência, facilitando a identificação de padrões visuais (Figura 8).

¹ <<https://support.microsoft.com/images/en-us/b3bfff05-380e-4dd9-b49b-7a2bfd128870>> (Acessado em 23/06/2025).

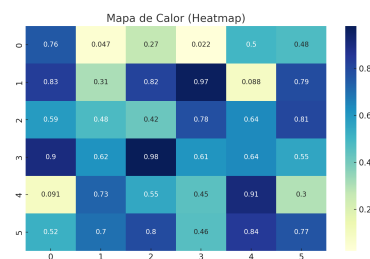


Figura 8 – Exemplo de mapa de calor (heatmap). Fonte: Geração própria.

No estudo de JACOB; WALIZER; KINZIE (2023) foi enfatizado a importância de selecionar ferramentas adequadas para a visualização de dados, destacando opções como gráficos de barras e linhas, diagramas de dispersão, mapas de calor e infográficos. Além disso, mencionaram ferramentas digitais avançadas, como Tableau conforme mostrado pela Figura 11 e também por meio do PowerBI como mostram os exemplos retirados do próprio projeto Cidades do Conhecimento na Figura 10 e 12, que possibilitam a criação de visualizações interativas e dinâmicas. Para JACOB; WALIZER; KINZIE (2023), a escolha da ferramenta deve sempre ser feita de forma a simplificar o entendimento, garantindo que o público não se sinta sobrecarregado ou incapaz de compreender a informação. O estudo de JACOB; WALIZER; KINZIE (2023) também abordou o uso do *data storytelling* para contextualizar as visualizações, apresentando números e dados não como informações isoladas, mas como parte de uma narrativa que explica o contexto, os desafios e os impactos dessas informações.

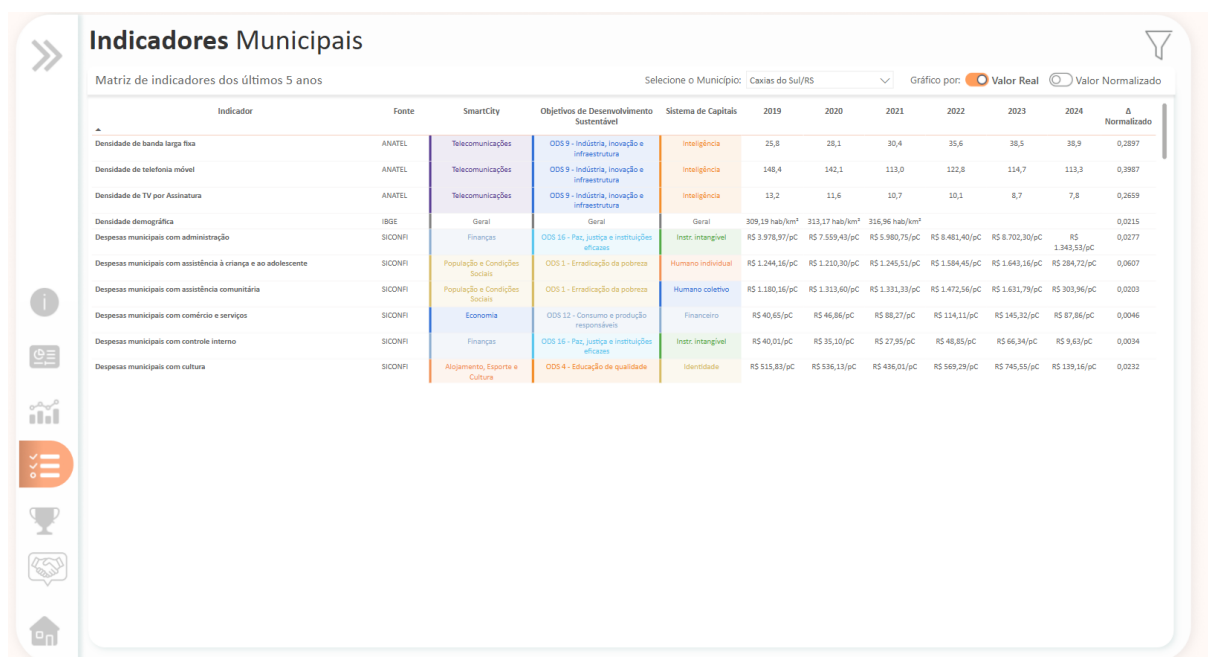


Figura 9 – Exemplo de Power BI demonstrando uma matriz de indicadores dos últimos cinco anos, retirado do projeto Cidades do Conhecimento.

Fonte: Citing Living Lab²



Figura 10 – Página de ranking do Power BI demonstrando gráficos de diferentes indicadores do modelo *Smart City*, desenvolvido para o projeto Cidades do Conhecimento.

Fonte: Citing Living Lab³

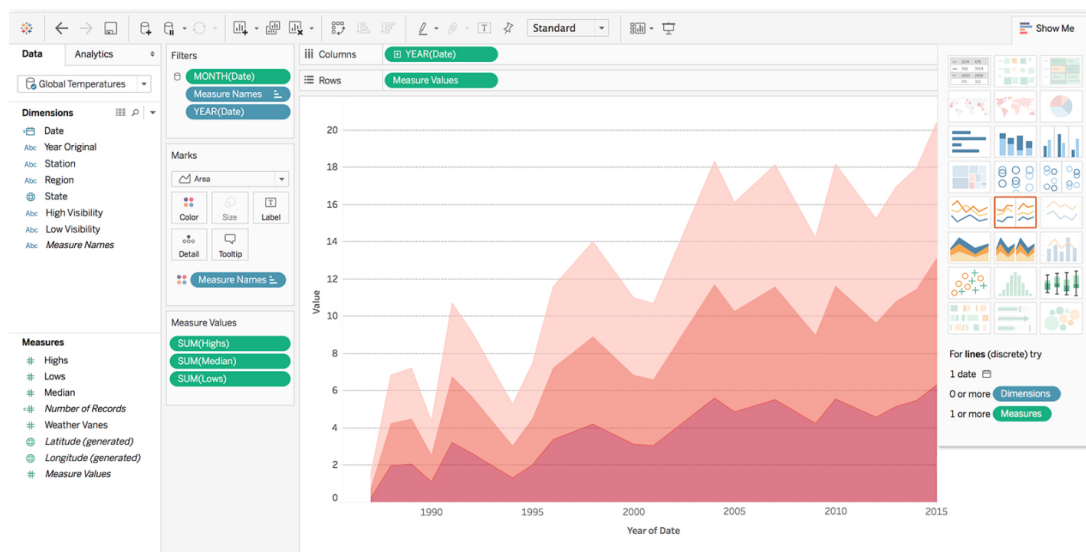


Figura 11 – Exemplo de Tableau.

Fonte: Tableau⁴

O livro *Mobile Data Visualization*, editado por LEE; HEER; AGRAWALA (2021), reúne contribuições de diversos pesquisadores com foco nos desafios e nas oportunidades da visualização de dados em dispositivos móveis. Ao longo da obra, discute-se como as limitações

² <<https://www.citylivinglab.com/>> (Acessado em 25/06/2025).

³ <<https://www.citylivinglab.com/>> (Acessado em 25/06/2025).

⁴ <<https://www.tableau.com/pt-br/blog/hello-world-its-tableau-10-55481>> (Acessado em 25/06/2025).

inerentes a *smartphones* e *tablets*, como telas reduzidas, interação por toque e a própria mobilidade do usuário, demandam abordagens específicas de *design* e interação. Nesse contexto, LEE; HEER; AGRAWALA (2021) também apresenta diferentes níveis de interação em visualizações móveis, que variam desde a interação passiva (*passive interaction*), em que o usuário apenas observa a informação apresentada, até formas mais avançadas, como a interação simples na qual é possível realizar ajustes básicos como filtros ou ordenações; a especificação e manipulação da visualização que permite alterar parâmetros e explorar dados de maneira mais ativa; e a interação livre caracterizada por um alto grau de controle, personalização e manipulação dinâmica dos dados.



Figura 12 – Exemplo de cada tipo de visualização e interação.

Fonte: Retirado do artigo LEE; HEER; AGRAWALA (2021)

Além disso, LEE; HEER; AGRAWALA (2021) cita diversas estratégias voltadas ao contexto de visualização de grandes quantias de dados, entre elas a simplificação visual, técnicas de navegação, zoom ou ampliação, uso de múltiplas visualizações coordenadas, adaptação responsiva e exploração baseada em gestos. Por fim, destaca-se que visualizações para dispositivos móveis não devem ser meras versões reduzidas de interfaces *desktop*, mas sim soluções concebidas especificamente para o contexto móvel, considerando cenários reais de uso e interação.

Para complementar, no estudo de SVALINA *et al.* (2023) foi avaliada a adequação de visualizações interativas de dados para uso em dispositivos móveis. A partir de avaliações empíricas, os autores identificaram problemas relacionados à sobrecarga visual, dificuldade de seleção precisa e perda de legibilidade em layouts densos. Como solução, o estudo aponta que versões simplificadas das visualizações, com redução do número de categorias exibidas simultaneamente, maior espaçamento entre elementos e uso de interações como zoom e destaque sob demanda, demonstraram melhor desempenho e usabilidade. Os achados reforçam que adaptações estruturais no *design*, e não apenas a redução de escala, são uma pré-condição para tornar as visualizações de dados efetivas em *smartphones*.

2.2 DESENVOLVIMENTO MOBILE

O desenvolvimento mobile abrange o conjunto de processos, técnicas e ferramentas empregados na criação de aplicações para dispositivos móveis. O crescimento da utilização desses

dispositivos impulsionou a adoção de novas tecnologias e práticas de desenvolvimento, permitindo a construção de aplicações cada vez mais completas e adaptáveis às diferentes necessidades dos usuários. Dessa forma, o desenvolvimento de aplicações móveis contempla aspectos que vão desde a definição da arquitetura e das tecnologias utilizadas até questões relacionadas ao desempenho, à segurança e à usabilidade.

2.2.1 Interfaces em mobile

A Interface do Usuário (UI) refere-se ao design visual e interativo de um produto, sendo o ponto de contato entre o usuário e o sistema. De acordo com o estudo de HAMIDLI (2023), a UI abrange os princípios de criação de elementos gráficos, como botões, menus, ícones, tipografias e layouts. Sua principal ênfase está na estética e na funcionalidade do design, assegurando que os elementos na tela sejam não apenas visualmente agradáveis, mas também coerentes e acessíveis. Além disso, HAMIDLI (2023) destaca que a UI deve ser responsiva, ou seja, deve se adaptar de forma fluida e dinâmica a diferentes dispositivos e tamanhos de tela.

Em sua pesquisa SOLER; FARINA; FLORIAN (2021) trouxeram para discussão a importância da acessibilidade no design das interfaces digitais. Os autores defendem que, além do cumprimento das normas e diretrizes, a interface deve ser intuitiva — fácil de usar e de compreender, independentemente da experiência do usuário ou do seu conhecimento acerca da tecnologia. De acordo com SOLER; FARINA; FLORIAN (2021), a acessibilidade intuitiva não privilegia apenas aquelas pessoas que têm alguma limitação, mas todos os usuários que estão em relação com o sistema. A partir das análises e dos estudos de caso, SOLER; FARINA; FLORIAN (2021) evidenciaram que a navegação poderia tornar-se mais fluída por meio de elementos como o uso consistente de ícones, a utilização de uma linguagem clara e objetiva e também um feedback imediato do sistema após alguma ação do usuário.

No estudo apresentado por LIU (2025), foi desenvolvido um modelo para avaliação de interfaces que integra elementos do design visual com a abordagem analítica do Processo Analítico Hierárquico (AHP). A pesquisa identificou os principais fatores essenciais para a construção de interfaces eficazes, incluindo cor, layout, tipografia, legibilidade e interatividade. A partir do AHP, foi criada uma hierarquia de critérios que permitiu aos especialistas atribuir pesos específicos a cada um desses fatores. Os resultados revelaram que os fatores mais impactantes na eficácia da interface são o layout e a interatividade, seguidos pela escolha das cores e pela legibilidade do texto.

WOOD (2014) em seus estudos abrangeu os princípios fundamentais do design de interfaces interativas. Abordou como os elementos visuais, como tipografia, cor, layout, formas e imagens, contribuem diretamente para a usabilidade e a eficácia de uma interface/sistema digital. Para WOOD (2014) o design da interface não é apenas uma questão estética, mas sim uma parte funcional voltada a mediação entre o usuário e a informação/sistema, ele explora o papel do designer como um comunicador visual, responsável por tornar a experiência do usuário

mais clara, intuitiva e agradável. Já JUNG (2017) em sua pesquisa mostrou como a qualidade do design da interface do usuário influenciava na intenção do usuário em realizar compras, JUNG (2017) destacou que a experiência visual é responsável por atrair e reter consumidores/usuários em ambientes digitais, em sua análise também abordou elementos como estética, funcionalidade e usabilidade, os ligando diretamente que a satisfação do usuário ao se comunicar com o sistema. Assim, o design da interface, que integra aspectos visuais e funcionais, não apenas facilita a interação do usuário com o sistema, mas também influencia diretamente sua experiência e satisfação, tornando o sistema mais intuitivo e agradável.

Essa interação, que envolve elementos como estética, funcionalidade e usabilidade, é ainda mais crucial quando se trata de dispositivos móveis. Nesses dispositivos, a experiência do usuário (UX) se torna um fator determinante, já que o usuário busca uma interação simples, eficiente e prazerosa. A (UX) diz respeito ao sentimento geral e à satisfação do usuário ao interagir com um produto. Segundo HAMIDLI (2023), UX é um conjunto de fatores que impactam a percepção do usuário em relação à facilidade, prazer e eficácia na utilização de um sistema. A UX abrange não apenas aspectos visuais, mas também elementos como usabilidade, acessibilidade e o tempo que o usuário dedica à interação com o sistema. Para HAMIDLI (2023), a pesquisa com usuários, a realização de testes de funcionalidades e a análise do comportamento são fundamentais para a criação de experiências agradáveis

LOPES (2014) em seu trabalho definiu o conceito de design responsivo, o qual, em resumo, se propõe a adequar o layout e o conteúdo de uma aplicação ou até uma página web às características do dispositivo do usuário, como o tamanho da tela, resolução e orientação. LOPES (2014) investigou várias ferramentas para a implementação do design responsivo, entre elas layouts fluídos, media queries e unidades de medida relativas. No contexto de aplicações móveis, LOPES (2014) abordou práticas como o uso de gestos de navegação, formulários otimizados para interfaces touch e a implementação de estratégias de carregamento condicional para melhorar o desempenho e a experiência do usuário. Outro ponto importante discutido foi a estratégia mobile-first, que defende que o desenvolvimento de sites e aplicativos comece pela própria versão móvel, sendo depois os ajustes para telas maiores.

MARTÍNEZ; CEBALLOS (2013) trouxe em seu trabalho a comparação entre design responsivo, mencionado anteriormente, e design adaptativo. A diferença do adaptativo para o responsivo é que o responsivo trabalha com um layout que vai se adaptando automaticamente de acordo com o dispositivo utilizado pelo usuário. Já no adaptativo é elaborado diversas versões do mesmo layout para diferentes dispositivos, ou seja, em vez de criar um único layout que se adapta, o aplicativo irá carregar um layout já pré-definido e otimizado para a resolução do dispositivo do usuário. O comparativo que MARTÍNEZ; CEBALLOS (2013) trazem é em questão de custo, onde o adaptativo por ter diversas versões de layout acaba sendo custoso em casos de manutenções ou alterações futuras, mas é mais otimizado em questão de performance pois não obriga o aplicativo a se ajustar em tempo real por já carregar o layout preciso para o dispositivo

em que o usuário está navegando.

Em sua pesquisa, WEICHBROTH (2021) analisou os principais problemas de usabilidade encontrados em dispositivos móveis, conforme apontado pelos profissionais da área, constituídos por designers, desenvolvedores, testers e gerentes de produto. Os problemas mais citados foram limitações das telas pequenas, interfaces pouco intuitivas, respostas inconsistentes aos gestos, baixo desempenho e falta de padronização entre os sistemas operacionais, como iOS e Android. No entanto, os profissionais entrevistados no estudo realizado por WEICHBROTH (2021) indicaram a necessidade de testes contínuos e iterativos para uma experiência satisfatória do usuário, também enfatizam que a usabilidade é um fator essencial para o sucesso de uma aplicação móvel e que a colaboração entre pesquisadores, desenvolvedores e pessoas do mercado é essencial para desenvolver soluções mais eficientes que realmente atendam as necessidades dos usuários.

O estudo de SILVA; JUNIOR; FREIRE (2022) apresenta uma investigação sobre padrões de design de interação utilizados em interfaces móveis. O objetivo central foi de compreender como tais padrões têm sido desenvolvidos, documentados e implementados na prática no intuito de ajudar os designers e desenvolvedores a criar experiências de usuário mais eficientes e consistentes. O design da interação diz respeito ao conjunto de práticas e soluções sobre como os usuários irão interagir com aplicativos móveis, abrangendo gestos, toques, transições, navegações, feedback visual e controle de ações. Os padrões de design de interação representam soluções para problemas recorrentes nesse tipo de interface. O estudo de SILVA; JUNIOR; FREIRE (2022) reuniu 336 padrões documentados em 23 fontes diferentes, onde 261 continham propostas únicas, agrupadas em 18 categorias, como navegação, entrada de dados, feedback, autenticação, layout, ações do usuário e até padrões negativos (como dark patterns e antipadrões), mostrando a diversidade de soluções para diferentes aspectos da interação em dispositivos móveis. Esses padrões de design de interação, abrangendo desde gestos e toques até transições e feedback visual, são essenciais para criar experiências de usuário mais consistentes e eficientes em dispositivos móveis.

2.2.2 Abordagens de desenvolvimento

Para dar início ao desenvolvimento de aplicações móveis de maneira estruturada os desenvolvedores precisam escolher as tecnologias apropriadas para o desenvolvimento do aplicativo. As abordagens de desenvolvimento, como nativas ou multiplataforma, têm um impacto direto na criação de interfaces móveis, pois cada uma oferece diferentes vantagens e desafios em termos de desempenho, usabilidade e manutenção.

Em seu estudo BUŞE (2023) analisou as principais diferenças entre o desenvolvimento de aplicações móveis nativas e multiplataforma, discutindo suas vantagens, limitações e contextos de uso. O autor destaca que aplicações nativas, desenvolvidas especificamente para cada sistema operacional, tendem a apresentar melhor desempenho, maior acesso aos recursos do dis-

positivo e uma experiência de usuário mais otimizada, embora demandem mais tempo, custos maiores e equipes especializadas para cada plataforma. Em contrapartida, o desenvolvimento multiplataforma permite a criação de aplicações para diferentes sistemas a partir de uma única base de código, reduzindo custos e tempo de desenvolvimento, além de facilitar a manutenção. No entanto, essa abordagem pode apresentar limitações de desempenho e menor acesso a funcionalidades específicas do hardware. Dessa forma, BUŞE (2023) conclui que a escolha entre desenvolvimento nativo ou multiplataforma depende de fatores como os objetivos do projeto, orçamento disponível, especialização da equipe, prazos de desenvolvimento e também o nível de desempenho requerido pela aplicação. Complementando, em seu estudo NUNES; QUEIROZ (2021) apresentaram uma análise crítica sobre as diversas estratégias de desenvolvimento de aplicativos móveis. Dentro do desenvolvimento mobile, as abordagens nativas se referem ao uso de linguagens e ferramentas específicas para cada sistema operacional. Por exemplo, no caso do Android, utiliza-se:

- **Java:** Linguagem de programação orientada a objetos amplamente usada para o desenvolvimento Android, conhecida por sua estabilidade e grande comunidade de suporte.
- **Kotlin:** Linguagem moderna, também orientada a objetos, desenvolvida pela JetBrains. Oferece maior concisão e segurança em relação ao Java, sendo atualmente a linguagem recomendada pelo Google para Android.

Para o sistema iOS, as principais linguagens nativas são:

- **Swift:** Linguagem desenvolvida pela Apple, moderna, segura e eficiente. É a principal escolha para o desenvolvimento de aplicativos iOS atualmente.
- **Objective-C:** Linguagem mais antiga, baseada em C, anteriormente usada como padrão para aplicações iOS e ainda mantida em muitos projetos legados.

Nesse contexto, enquanto o desenvolvimento nativo utiliza linguagens e ferramentas específicas de cada sistema operacional, o desenvolvimento multiplataforma baseia-se em frameworks que permitem a criação de aplicações capazes de operar em diferentes plataformas a partir de uma única base de código. Entre os principais frameworks multiplataforma estão:

- **React Native:** Framework criado pelo Facebook que permite o desenvolvimento de aplicativos usando JavaScript e React. Ele traduz componentes da interface para elementos nativos, garantindo boa performance e reutilização de código.
- **Flutter:** Framework desenvolvido pelo Google que utiliza a linguagem Dart. Ele permite a criação de interfaces ricas e altamente personalizáveis, com alto desempenho e um motor gráfico próprio.

- **NativeScript:** Framework que permite o desenvolvimento de aplicativos com JavaScript, TypeScript ou Angular, oferecendo acesso direto às APIs nativas do sistema operacional.

SYSTÄ; SAND (2023) realizaram uma série de experimentos de diversas aplicações móveis desenvolvidas de ambas formas, tanto de forma nativa quanto utilizando frameworks multiplataforma. No estudo foram analisados diferentes aspectos, como desempenho, consumo de recursos, tempo gasto no desenvolvimento e a facilidade de manutenção, buscando identificar as diferenças práticas entre essas abordagens. Os resultados mostraram que aplicações nativas geralmente apresentam melhor desempenho e integração mais eficiente com os recursos do dispositivo, enquanto soluções multiplataforma oferecem vantagens em termos de reutilização de código e redução do esforço de desenvolvimento para múltiplos sistemas operacionais. O estudo de SYSTÄ; SAND (2023) concluiu que, embora frameworks multiplataforma sejam adequados e abrangem a maior quantia de cenários, aplicações nativas ainda tendem a ser preferidas quando o desempenho máximo e a integração profunda com o hardware são requisitos prioritários.

Complementando essa discussão, NUNES; QUEIROZ (2021) conduziram um estudo baseado no desenvolvimento de um aplicativo utilizando uma abordagem multiplataforma, com o objetivo de analisar suas vantagens e limitações na prática. Foram avaliados critérios como desempenho, reutilização de código, manutenção, acesso a funcionalidades específicas do dispositivo e experiência do usuário final. Os resultados demonstraram que, apesar de existirem algumas limitações — especialmente relacionadas ao desempenho e à integração com determinados recursos do dispositivo —, as abordagens multiplataforma proporcionam ganhos significativos em produtividade, redução de custos e agilidade no processo de desenvolvimento. Ainda assim, os autores destacam que a escolha da abordagem mais adequada depende das características e dos requisitos específicos de cada aplicação.

2.2.3 Arquitetura de mobile

A arquitetura de desenvolvimento mobile pode ser compreendida como a forma de organização dos componentes de uma aplicação para dispositivos móveis, definindo como elementos como a interface do usuário, a lógica de negócio e o acesso a dados são estruturados e se comunicam entre si. Essa organização orienta o desenvolvimento da aplicação, permitindo separar responsabilidades entre diferentes camadas ou módulos, o que contribui para a manutenção, escalabilidade e melhor gerenciamento do código.

Em seu artigo, NYABUTO; MONY; MBUGUA (2024) apresentaram uma revisão arquitetural do modelo cliente-servidor, destacando sua importância no desenvolvimento de sistemas distribuídos. NYABUTO; MONY; MBUGUA (2024) descrevem como essa arquitetura organiza aplicações em duas ou mais camadas, nas quais os clientes são responsáveis por solicitar serviços e os servidores por processar requisições, gerenciar recursos e fornecer respostas. O estudo discute diferentes variações do modelo, como arquiteturas de duas camadas (Figura 13) e três

camadas (Figura 14), ressaltando aspectos como escalabilidade, manutenção, segurança e desempenho.

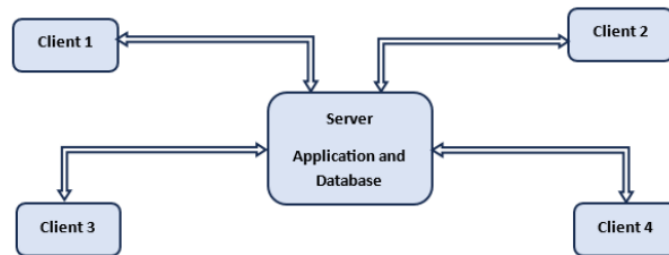


Figura 13 – Exemplo de arquitetura de duas camadas. Fonte: NYABUTO; MONY; MBUGUA (2024)



Figura 14 – Exemplo de arquitetura de três camadas. Fonte: NYABUTO; MONY; MBUGUA (2024)

Além disso, o trabalho de NYABUTO; MONY; MBUGUA (2024) aponta vantagens do modelo, como centralização do processamento e facilidade de gerenciamento de dados, ao mesmo tempo em que reconhece desafios relacionados à dependência do servidor e possíveis gargalos de desempenho. A análise reforça que, apesar do surgimento de novas arquiteturas, o modelo cliente-servidor continua sendo um dos fundamentos mais utilizados na construção de aplicações.

Uma API (*Application Programming Interface*) pode ser compreendida como um conjunto de regras e interfaces que permite a comunicação entre diferentes sistemas ou aplicações, possibilitando que funcionalidades e dados de um sistema sejam acessados de forma padronizada por outros. Nesse contexto, MUHAMMAD; PAPUTUNGAN (2024) descrevem o desenvolvimento de APIs baseadas em arquitetura REST, nas quais o lado do servidor, o *backend*, disponibiliza recursos e operações por meio de *endpoints* acessíveis por requisições HTTP, como demonstra a figura 15. Nesse modelo, aplicações clientes podem solicitar, enviar ou modificar dados através da API, que atua como intermediária entre o cliente e o servidor, processando as requisições e retornando as respostas apropriadas.

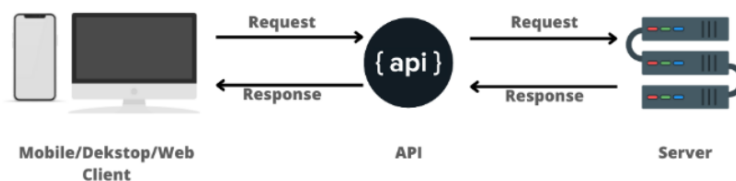


Figura 15 – Exemplo do funcionamento de uma API. Fonte: MUHAMMAD; PAPUTUNGAN (2024)

Dessa forma, as APIs permitem a integração entre diferentes aplicações, como sistemas web e móveis, garantindo uma comunicação estruturada e organizada entre os componentes do sistema. Complementando, (DELJOUYI; RAMSIN, 2022) apresentaram uma metodologia orientada a modelos para o desenvolvimento de aplicações, destacando os princípios do estilo arquitetural REST (*Representational State Transfer*). Segundo (DELJOUYI; RAMSIN, 2022), REST define uma forma padronizada de estruturar a comunicação entre aplicações clientes e servidores, na qual os dados e funcionalidades de um sistema são organizados como recursos, identificados por endereços específicos. Esses recursos podem ser acessados e manipulados por meio de requisições do protocolo HTTP direcionadas a esses endereços. Dessa forma, REST estabelece um modelo de comunicação simples, organizado e escalável entre diferentes sistemas, sendo amplamente utilizado no desenvolvimento de APIs e serviços web modernos.

Nesse contexto, em seu estudo ZIMMERMANN *et al.* (2022) apresentaram uma abordagem estruturada para o design de APIs que facilitam a integração entre sistemas de forma desacoplada e sustentável, os autores partiram da premissa que uma API bem projetada é essencial para interoperabilidade de sistemas e aplicações modernas. ZIMMERMANN *et al.* (2022) propuseram 44 padrões de design que visam orientar decisões desde os estágios iniciais do projeto até a evolução e documentação das APIs. Os padrões foram agrupados para contemplar tópicos como *endpoints*, operações, estruturas de mensagens, controle de versão e boas práticas de documentação. Os padrões foram desenvolvidos no contexto de práticas e pesquisas aplicadas em projetos reais, ZIMMERMANN *et al.* (2022) descrevem como aplicar os padrões com o intuito de responder a decisões comuns de design tais quais o estilo de interação (REST, RPC, eventos), modelagem de endpoints, estruturação de mensagens de entrada e saída, tratamento de erro, versionamento e garantias de que a API continue compreensível e sustentável no contexto de sua evolução. Cada padrão é completado por forças (contexto e motivações), problemas comuns, soluções sugeridas, implicações da adoção e exemplos reais de adoção, deixando a equipe de desenvolvimento apta a analisar as opções e a adaptar as soluções ao seu ambiente. Esses padrões de design de APIs são fundamentais para garantir a interoperabilidade de sistemas, permitindo que diferentes componentes de software interajam de forma eficiente e sustentável, mesmo à medida que as aplicações evoluem.

O estudo de GUO *et al.* (2022) investigou como aplicações clientes interagem com APIs REST para acessar serviços e dados disponibilizados por sistemas distribuídos. GUO *et al.* (2022)

destacaram que aplicações modernas, como aplicações web e móveis, frequentemente dependem de múltiplas chamadas a APIs para executar suas funcionalidades. Nesse contexto, o lado cliente, o *frontend*, atua como consumidor dessas APIs, enviando requisições a diferentes *end-points* para realizar operações como buscar informações, enviar dados ou atualizar registros no sistema. Por exemplo, uma aplicação pode solicitar dados de usuários, registrar novas informações ou atualizar conteúdos por meio dessas requisições, utilizando as respostas retornadas pela API para atualizar a interface e apresentar os resultados ao usuário. Dessa forma, as APIs permitem que aplicações clientes acessem e utilizem funcionalidades de sistemas remotos de maneira estruturada e padronizada.

2.3 TRABALHOS CORRELATOS

Foi realizado um estudo dos trabalhos relacionados, no qual se identificou a importância da interseção entre o desenvolvimento de soluções mobile e a necessidade de métodos estruturados para o levantamento e a análise de dados em Living Labs (LLs). Esse estudo permitiu compreender como as plataformas digitais atuam como instrumentos de monitoramento e suporte à tomada de decisão, convertendo as interações de cocriação em dados mensuráveis essenciais para o planejamento estratégico e a inovação tecnológica.

2.3.1 Living Labs

Nesse sentido, TERCANLI; JONGBLOED (2022) realizaram uma ampla revisão sobre os Living Labs (LLs) em universidades, demonstrando como esses ecossistemas integram ensino, pesquisa e extensão ao unir estudantes, cientistas e parceiros externos na resolução de problemas reais. Embora tenham surgido de maneira informal, os autores defendem que os LLs podem atuar como ferramentas estratégicas de mudança institucional, apesar de enfrentarem desafios críticos como a indefinição de modelos de governança, a dificuldade de integração à estrutura acadêmica tradicional e a necessidade de capacitação transdisciplinar. Expandindo essa visão para o contexto da sustentabilidade, GALWAY *et al.* (2022) reforçam o papel dos LLs como centros de co-criação multisetoriais, mas observam que temas como justiça social e o impacto do aprendizado dos usuários ainda são mais retóricos do que práticos, sugerindo uma expansão dessas iniciativas para além do ambiente urbano. Para viabilizar essas estruturas, WANG *et al.* (2022) identificam desafios operacionais na coordenação de diversos *stakeholders* e propõem um planejamento estratégico flexível, pautado no uso de tecnologias colaborativas, na transparência e na participação ativa da comunidade para garantir que as soluções desenvolvidas gerem impacto social e tecnológico real.

O artigo de MOUNIR (2023) apresenta uma revisão abrangente dos principais desafios enfrentados no desenvolvimento de aplicações móveis e das soluções propostas. Segundo MOUNIR (2023), a enorme variedade de sistemas e aparelhos, a decisão entre desenvolver de forma

nativa ou optar por uma solução híbrida, a natureza sempre mutável dos cenários móveis e a incorporação de práticas ágeis juntam-se para tornar o trabalho bem mais complicado. Para lidar com isso, MOUNIR (2023) menciona ferramentas como frameworks híbridos, técnicas que permitem que os aplicativos sejam sensíveis ao contexto do usuário, a automação nos testes para acomodar essa diversidade de hardwares e a prática de integração contínua no âmbito das metodologias ágeis. MOUNIR (2023) defende que, para ter sucesso, um aplicativo precisa responder rapidamente ao ambiente de quem o usa, equilibrando bom desempenho e uma experiência uniforme, e conclui que a engenharia móvel deve avançar sem parar, acompanhando a velocidade das inovações tecnológicas e as expectativas cada vez maiores dos consumidores. Dando continuidade ao tema, o estudo de PATIDAR; SUMAN (2022) destaca o papel central das metodologias ágeis, apresentando um panorama claro das técnicas mais usadas na indústria de software. Segundo PATIDAR; SUMAN (2022), apesar de ainda haver espaço para abordagens clássicas, frameworks ágeis como Scrum, XP e DSDM conquistam preferência por conseguirem absorver mudanças constantes, colocar a experiência do usuário em primeiro lugar e entregar valor mais rapidamente. Além disso, PATIDAR; SUMAN (2022) traz evidências de uma pesquisa com profissionais da área que confirmam que a flexibilidade e a capacidade de adaptação dos métodos ágeis são fatores-chave para o sucesso de aplicativos móveis em um mercado cada vez mais rápido e exigente.

O artigo de HABIBIPOUR; STÅHLBRÖST (2022) investiga como a metodologia de Living Labs pode transformar o desenvolvimento de sistemas e aplicações ao constituir um ambiente colaborativo composto por usuários reais com diversos *stakeholders* que desempenham um papel ativo do início ao fim do processo de desenvolvimento, permitindo a cocriação e testes em ambientes reais de uso. HABIBIPOUR; STÅHLBRÖST (2022) discutem as limitações dos processos usuais, que frequentemente não refletem as verdadeiras necessidades dos usuários e introduzem um modelo conceitual que aproxima os princípios dos LLs — experimentação aberta, inovação centrada no usuário e adaptação dinâmica — ao ciclo do desenvolvimento, acentuando a importância da interação contínua e do aprendizado interpessoal para alcançar sistemas mais flexíveis e pertinentes. A pesquisa de HABIBIPOUR; STÅHLBRÖST (2022) indica que essa perspectiva não transforma apenas a inovação tecnológica, mas também promove um aumento do engajamento e alinhamento com os objetivos dos usuários e organizações, apresentando a Living Lab como uma estratégia para desenvolver sistemas mais responsivos e sustentáveis entre ambientes complexos e dinâmicos.

2.3.2 Citing Living Lab e a plataforma desenvolvida para o Cidades do Conhecimento

A primeira versão da plataforma web do projeto foi feita para atender à necessidade de automatizar a coleta, organização e visualização dos indicadores relacionados às Cidades do Conhecimento, conceito baseado no Sistema de Capitais Genérico. Antes disso, o processo era

manual, realizado em planilhas Excel, o que gerava erros, redundância e dificuldade na geração de relatórios e gráficos. A plataforma permitiu que usuários acessassem e comparassem dados de diferentes cidades e anos, categorizando-os em capitais como Identidade, Inteligência, Relacional, Financeiro, Humano Individual, Humano Coletivo, Instrumental Material e Instrumental Intangível. Por meio dela, era possível gerar gráficos comparativos, consultar indicadores detalhados, visualizar tabelas com dados normalizados e avaliar o desenvolvimento das cidades a partir de critérios além dos tradicionais, como PIB (Produto Interno Bruto) e IDH (Índice de Desenvolvimento Humano). A Figura 16 ilustra a página inicial da primeira versão da plataforma desenvolvida por BATTISTELO (2018).

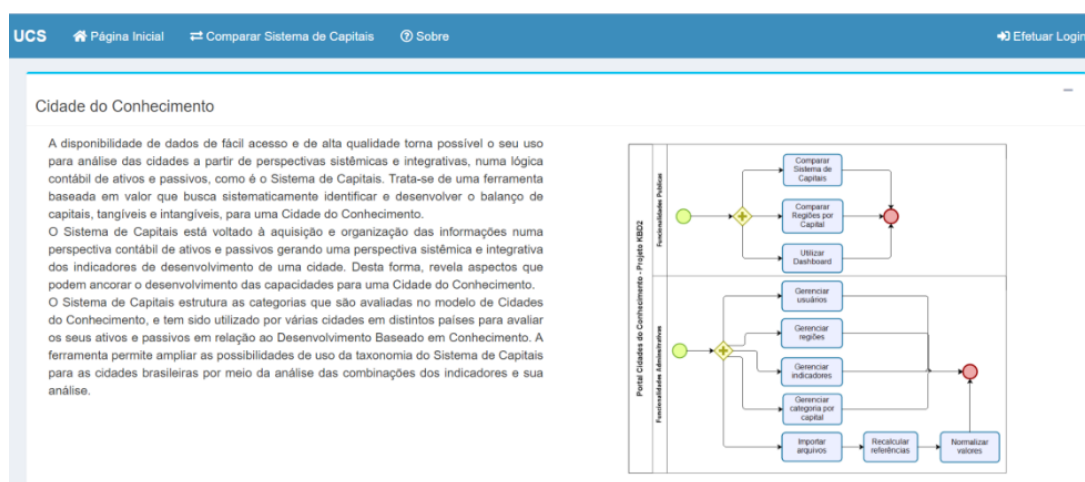


Figura 16 – Página inicial primeira versão da plataforma web. Fonte: BATTISTELO (2018)

Para viabilizar essas funcionalidades, a aplicação foi desenvolvida utilizando uma série de ferramentas que, em conjunto, garantem robustez, organização e usabilidade à plataforma. A seguir, estão as principais tecnologias utilizadas:

- **PHP**: Linguagem de programação para desenvolvimento web no lado servidor, amplamente utilizada para criação de aplicações dinâmicas.
- **CodeIgniter**: Framework PHP que adota o padrão arquitetural MVC (Model-View-Controller), permitindo uma separação clara entre regras de negócio, visualização e controle.
- **MySQL**: Sistema de gerenciamento de banco de dados relacional, responsável pelo armazenamento dos dados da aplicação.
- **Bootstrap**: Biblioteca front-end que oferece componentes prontos e responsivos, facilitando a construção de interfaces adaptáveis a diferentes tamanhos de tela.
- **AdminLTE**: Template de painel administrativo baseado em Bootstrap, que oferece uma série de elementos visuais e layouts prontos para criação de dashboards e sistemas de administração.

- **Chart.js**: Biblioteca JavaScript utilizada para geração de gráficos dinâmicos e interativos, facilitando a visualização dos dados na interface.
- **PHPExcel**: Biblioteca PHP que permite a leitura, escrita e manipulação de arquivos Excel, utilizada na plataforma para importação de dados a partir de planilhas.

A plataforma oferece funcionalidades tanto para usuários anônimos — que podem consultar e gerar comparativos — quanto para administradores, que podem cadastrar indicadores, importar dados e recalcular referenciais. Como resultado, o sistema resolveu os principais problemas do processo anterior, proporcionando dados mais acessíveis, centralizados e visualmente interpretáveis. A Figura 17 mostra a tela de seleção de cidades e a Figura 18 apresenta parte da interface interna do sistema utilizado na época.

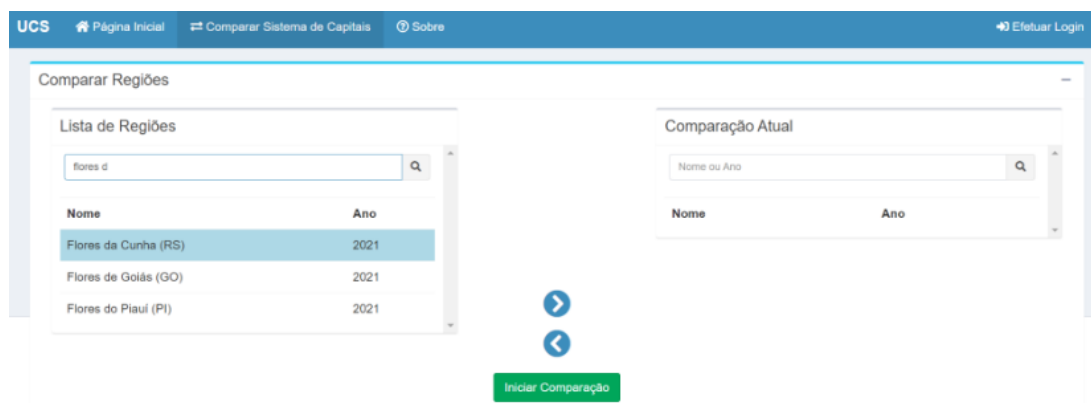


Figura 17 – Seleção de cidades da primeira versão da plataforma web. Fonte: BATTISTELO (2018)

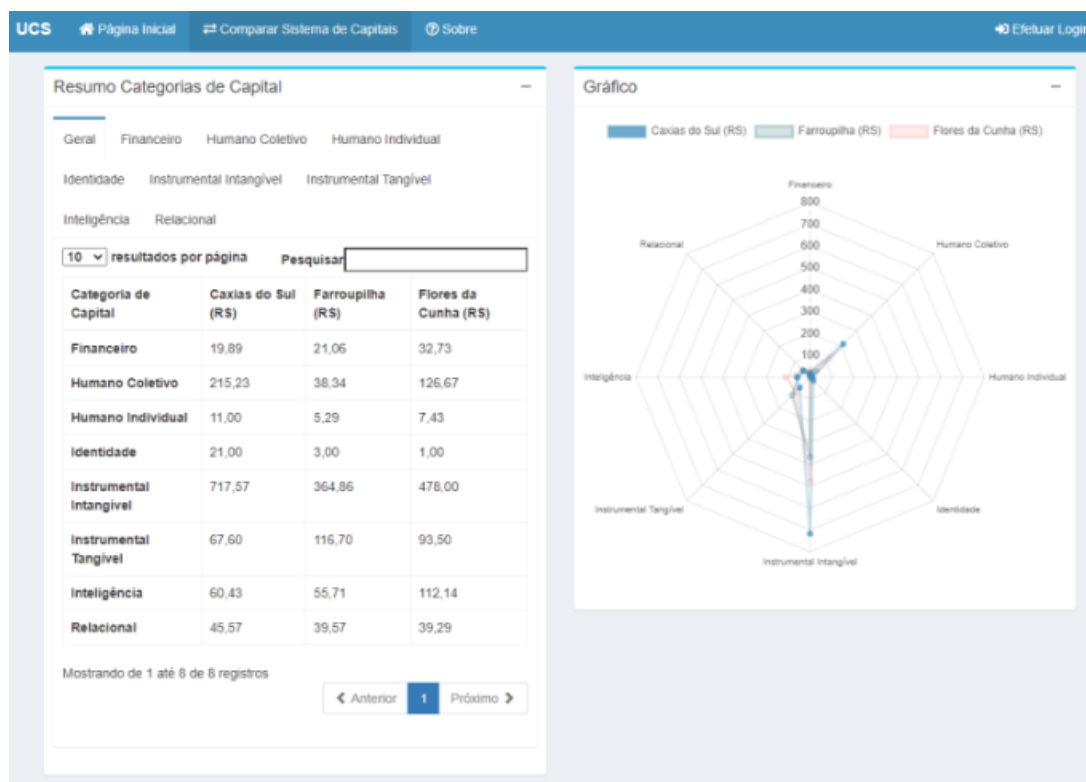


Figura 18 – Sistema da primeira versão da plataforma web. Fonte: BATTISTELO (2018)

Diante do avanço contínuo do projeto e da crescente importância da análise de dados no apoio à formulação de políticas públicas, o trabalho de MARIA (2024) representa uma continuidade e aprimoramento da plataforma originalmente proposta por BATTISTELO (2018). Desenvolvida inicialmente para a análise de indicadores e conceitos aplicados a um conjunto restrito de municípios da Serra Gaúcha, a plataforma passou por um processo de reestruturação nos níveis estrutural, conceitual e tecnológico. Essa reformulação teve como objetivo ampliar seu alcance, permitindo que usuários de diferentes regiões do Brasil pudessem acessar, consultar e comparar indicadores de desenvolvimento municipal. Além disso, buscou consolidar uma visão mais integrada e abrangente, em sintonia com as diretrizes de desenvolvimento sustentável e com os princípios das cidades inteligentes (Smart Cities). A Figura 19 apresenta a nova página inicial proposta por MARIA (2024), já com abordagem expandida.

Figura 23 – Página inicial



Figura 19 – Página inicial da segunda versão da plataforma web. Fonte: MARIA (2024)

Para além da sua maior abrangência, a nova proposta dotou a plataforma de uma abordagem mais abrangente e alinhada com as atuais necessidades de sustentabilidade, inovação e transformação digital das cidades. Para tal, para além do modelo baseado no Capitals Framework – que já permitia uma análise em profundidade dos ativos tangíveis e intangíveis dos territórios –, são acrescentados dois novos modelos de análise – os ODS e os indicadores de Smart Cities. Essa complementação amplia as oportunidades de análise e oferece uma visão mais completa dos problemas urbanos, possibilitando uma compreensão mais aprofundada das dinâmicas sociais, econômicas, ambientais e tecnológicas dos territórios. A Figura 20 mostra a tela de seleção de cidades, enquanto a Figura 21 exhibe um gráfico comparativo gerado pela plataforma.



Figura 20 – Seleção de cidades da segunda versão da plataforma web. Fonte: MARIA (2024)

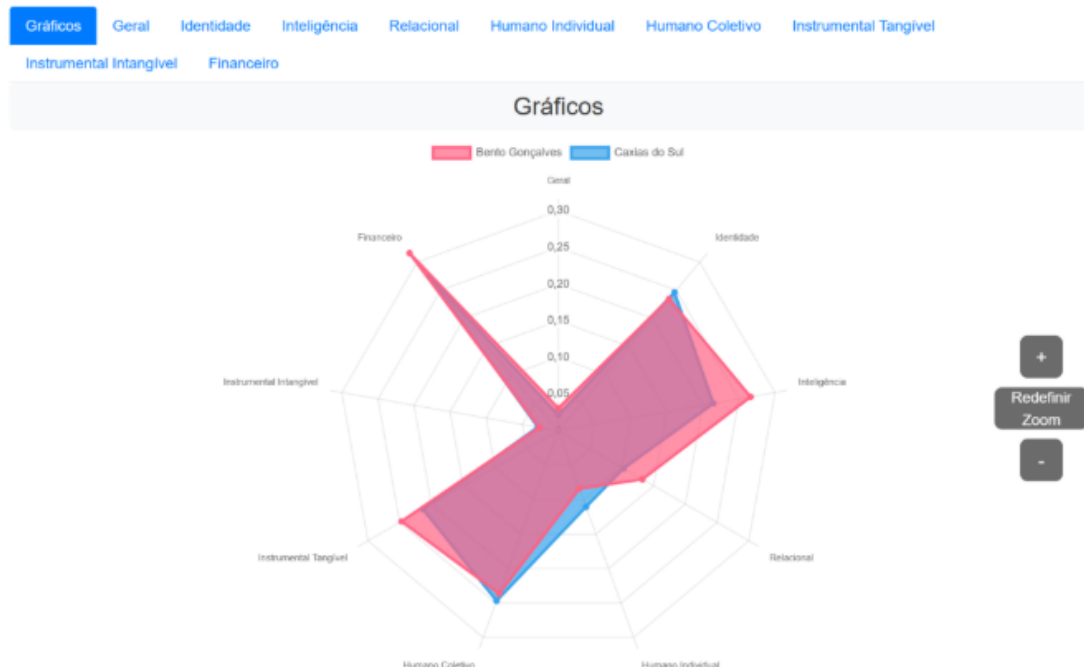


Figura 21 – Gráfico comparativo da segunda versão da plataforma web. Fonte: MARIA (2024)

Para atender a essas demandas, a plataforma foi completamente reestruturada com o uso de tecnologias mais modernas e eficientes, garantindo maior desempenho, escalabilidade e uma interface aprimorada para os usuários. A seguir, estão as principais tecnologias e ferramentas utilizadas nesse processo:

- **Django**: Framework de desenvolvimento web em Python que segue o padrão arquitetural MTV (Model-Template-View), oferecendo alto nível de segurança, escalabilidade e desenvolvimento ágil.
- **PostgreSQL**: Sistema de gerenciamento de banco de dados relacional, robusto e open source, conhecido pela sua confiabilidade, performance e suporte a consultas complexas.
- **Matplotlib**: Biblioteca em Python para geração de gráficos estáticos, animados e interativos, muito utilizada para visualização e análise de dados científicos.
- **Chart.js**: Biblioteca JavaScript que permite a criação de gráficos dinâmicos, interativos e responsivos diretamente na interface web, facilitando a interpretação visual dos dados.

A nova versão da plataforma oferece uma interface mais intuitiva, responsiva e dinâmica, proporcionando aos usuários uma experiência aprimorada. Além disso, disponibiliza funcionalidades que permitem a seleção de diferentes modelos de análise, cidades e períodos, possibilitando uma visualização mais abrangente, interativa e precisa dos dados urbanos por meio de dashboards e gráficos otimizados para apoiar a gestão pública e a tomada de decisões orientada por dados.

A Tabela 1 apresenta uma síntese dos principais pontos positivos e limitações das versões web da plataforma. Essa análise serviu como referência para o desenvolvimento da versão mobile proposta neste trabalho, orientando a manutenção de funcionalidades relevantes e a implementação de melhorias adequadas ao contexto de dispositivos móveis.

Tabela 1 – Detalhamento e síntese comparativa das versões anteriores da plataforma Cidades do Conhecimento

Aspecto	Versão 1 – BATTISTELO (2018)	Versão 2 – MARIA (2024)
Tecnologias Utilizadas	Baseada em <i>stack</i> LAMP clássica (PHP e MySQL). O uso do CodeIgniter focou em produtividade, enquanto o Chart.js gerava visualizações no lado do cliente.	Migração para ecossistema Python com Django, oferecendo maior segurança e suporte a bibliotecas de ciência de dados (Matplotlib) e banco PostgreSQL para consultas complexas.
Abrangência Geográfica	Foco experimental e regionalizado, atendendo especificamente às demandas e indicadores dos municípios da Serra Gaúcha.	Expansão da arquitetura de dados para suportar múltiplos estados e regiões, exigindo maior generalização dos parâmetros de entrada.
Modelos de Análise	Aplicação estrita do <i>Capitals Framework</i> , monitorando o desenvolvimento local através de 8 dimensões de capital (humano, social, etc.).	Abordagem multidimensional integrando o <i>Capitals Framework</i> às métricas globais da ONU (ODS) e indicadores específicos de <i>Smart Cities</i> .
Visualização de Dados	Exibição de informações em tabelas normalizadas e gráficos de barras/linhas simples, com foco em funcionalidade administrativa.	Implementação de <i>dashboards</i> interativos e intuitivos, priorizando a experiência do usuário (UX) e a facilidade de interpretação para não especialistas.
Público-alvo e Acesso	Sistema fechado para administradores e consulta anônima limitada, funcionando majoritariamente como um repositório de dados.	Foco em transparência e governança aberta, permitindo interações avançadas e consultas personalizadas para diferentes perfis de usuários.
Pontos Fortes	Pioneirismo na automatização de indicadores locais e estruturação da primeira base de dados organizada do projeto.	Alta escalabilidade de software, interface moderna com foco em usabilidade e alinhamento com diretrizes internacionais de sustentabilidade.
Limitações Identificadas	Arquitetura monolítica com tecnologias que dificultam a manutenção moderna e escopo limitado para expansão nacional.	Necessidade contínua de alimentação manual de dados e dependência de APIs externas para manter os indicadores atualizados em tempo real.

3 PROPOSTA DE SOLUÇÃO

Este trabalho propõe o desenvolvimento de uma plataforma *mobile* para o projeto Cidades do Conhecimento, com o objetivo de ampliar o acesso aos dados do projeto por meio de dispositivos móveis. A proposta visa oferecer aos usuários uma solução prática para consultar informações e indicadores, possibilitando o acompanhamento dos dados em qualquer lugar e a qualquer momento. Dessa forma, a aplicação *mobile* surge como uma evolução da proposta original desenvolvida por BATTISTELO (2018), posteriormente aprimorada por MARIA (2024), respondendo às demandas atuais por mobilidade e modernização no acesso às informações.

A versão *mobile* mantém os princípios e modelos conceituais adotados na plataforma web, que foram estabelecidos inicialmente por BATTISTELO (2018) e posteriormente expandidos por MARIA (2024), incorporando os três principais eixos metodológicos: o Sistema de Capitais, os ODS e o modelo de Cidades Inteligentes. No entanto, trata-se de uma solução projetada de forma independente, construída especificamente para atender às características e limitações do ambiente *mobile*, tanto no que diz respeito à arquitetura da aplicação quanto à modelagem dos dados e à experiência do usuário.

Dessa forma, a proposta da plataforma *mobile* representa uma evolução natural do projeto Cidades do Conhecimento, expandindo suas funcionalidades para além do ambiente web e proporcionando uma solução específica para o contexto *mobile*. A aplicação preserva os princípios conceituais que fundamentam o projeto, ao mesmo tempo em que oferece uma experiência de uso adequada às necessidades dos usuários atuais. Com isso, busca-se fortalecer ainda mais a capacidade da plataforma em apoiar análises, estudos e tomadas de decisão baseadas em dados, contribuindo para uma gestão urbana mais eficiente, sustentável e orientada por informações estruturadas e acessíveis.

3.1 REQUISITOS FUNCIONAIS

A definição dos requisitos funcionais é essencial para guiar o desenvolvimento da aplicação *mobile* proposta. Estes requisitos descrevem as funcionalidades esperadas do sistema, levando em conta a experiência do usuário, os objetivos do projeto Cidades do Conhecimento e os princípios de interoperabilidade e acessibilidade, a Figura 22 apresenta o diagrama de casos de uso da aplicação, destacando as principais funcionalidades acessíveis ao usuário final e também as funcionalidades que serão de obrigação do sistema.

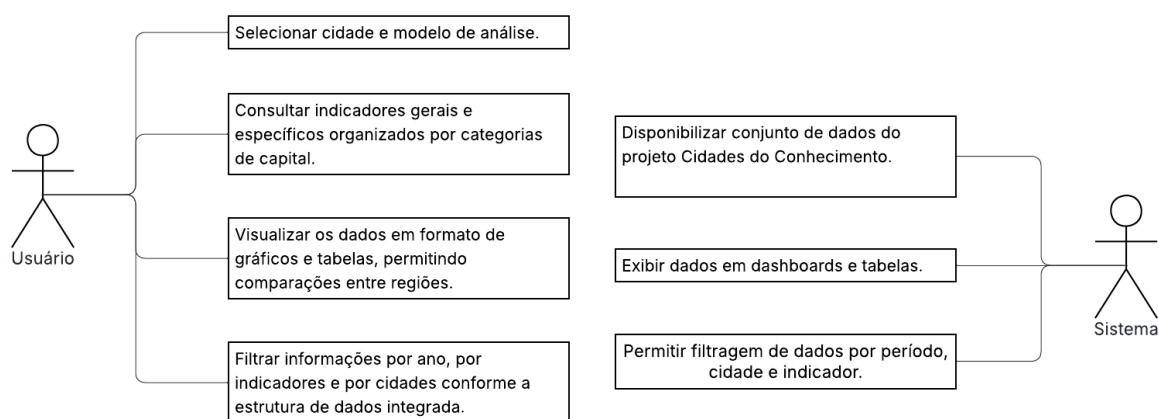


Figura 22 – Diagrama de casos de uso da aplicação. Fonte: Geração própria.

As funcionalidades da plataforma foram definidas para viabilizar um conjunto de ações essenciais voltadas à pesquisa, organização e visualização dos indicadores municipais. Essas funcionalidades foram identificadas com base nas demandas do projeto Cidades do Conhecimento e refletem os requisitos necessários para garantir uma apresentação estruturada e coerente dos dados. Dentre as funções atribuídas aos usuários podemos observar as seguintes:

- **Seleção de cidade e modelo de análise:** A plataforma irá permitir a definição de uma ou mais cidades, além da escolha entre diferentes modelos de análise. Os modelos disponibilizados organizam os indicadores conforme abordagens temáticas predefinidas, possibilitando diferentes formas de leitura dos dados.
- **Consulta a indicadores organizados por categorias de capital:** A plataforma oferecerá indicadores organizados em categorias chamadas de capitais, como capital humano, social, natural, produtivo, institucional e cultural. Esses grupos ajudam a separar os dados por temas relacionados ao desenvolvimento das cidades. Estão disponíveis tanto indicadores gerais, que servem para qualquer cidade, quanto indicadores mais específicos, usados em situações ou contextos particulares.
- **Visualização dos dados em gráficos e tabelas:** Os dados serão apresentados por meio de gráficos e tabelas interativos, com foco na clareza visual e na comparação entre regiões. As visualizações seguem padrões que facilitam a interpretação de resultados e a identificação de padrões ou contrastes entre os indicadores.
- **Filtragem por ano, indicador e cidade:** O sistema oferece mecanismos de filtragem que permitem restringir os dados visualizados a determinados anos, indicadores específicos e cidades selecionadas. Esses filtros atuam sobre a estrutura de dados integrada da aplicação, garantindo flexibilidade na consulta e análise dos resultados.

Além das funcionalidades atribuídas diretamente aos usuários, a plataforma também executará funções internas essenciais para o correto funcionamento do sistema. Essas funcionalidades são executadas automaticamente pelo sistema e asseguram a integração com a base de dados, o tratamento das informações e sua apresentação de forma organizada e consistente. Dentre as funcionalidades atribuídas ao sistema podemos observar as seguintes:

- **Fazer o link da aplicação com a base de dados existente do projeto Cidades do Conhecimento:** Essa funcionalidade garantirá que a aplicação (lado do cliente) esteja conectada à base de dados já construída pelo projeto (lado servidor), permitindo acesso direto e atualizado aos indicadores urbanos.
- **Retornar dados de forma gráfica, por meio de dashboards e/ou tabelas:** o sistema será responsável por processar as informações e exibi-las de maneira visual, facilitando a interpretação dos dados por meio de gráficos, painéis e tabelas dinâmicas.
- **Apresentar opções de filtragem por período, cidade e indicador conforme o modelo selecionado:** conforme o modelo de análise escolhido, o sistema disponibilizará filtros que permitem ajustar os dados exibidos de acordo com o ano, as cidades envolvidas e os indicadores desejados.

O fluxo de navegação da aplicação mobile foi estruturado de forma sequencial e orientada à tarefa, visando guiar o usuário desde o acesso inicial até as diferentes formas de visualização e análise de indicadores urbanos. Cada etapa conduz logicamente à próxima, permitindo tanto uma experiência direta quanto ramificações analíticas conforme as decisões do usuário. Para melhor visualização, podemos observar o processo abaixo:

A **Tela Inicial** apresenta o nome da plataforma e um *loading* que leva o usuário à próxima etapa do processo. Esta tela funciona como a porta de entrada para a experiência de navegação, como mostra na Figura 23.

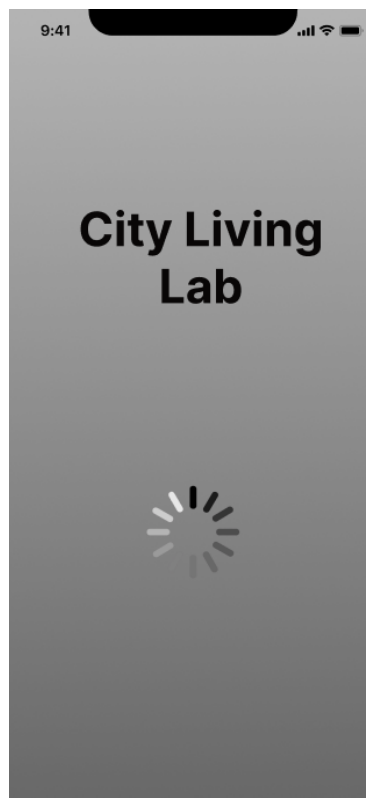


Figura 23 – Tela Inicial da aplicação. Fonte: Geração própria.

A partir da Tela Inicial (Figura 23), o usuário é direcionado à **Tela Inicial**. Nesta o usuário escolhe uma entre três opções, sendo a primeira uma tela que leva para a visualização dos indicadores, uma para o compartilhamento entre cidades e por último um link que leva para um informativo sobre o projeto, como mostra na Figura 24.

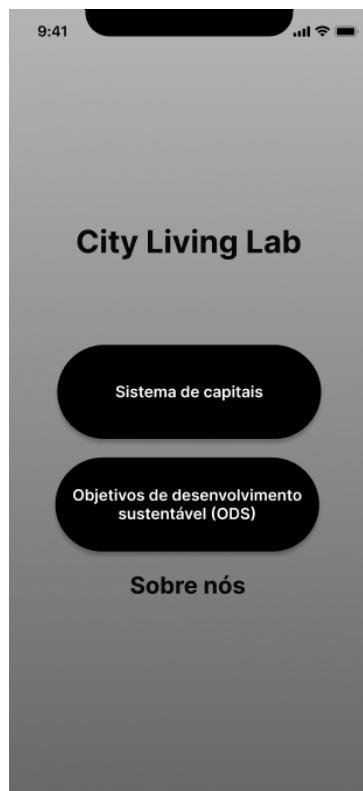


Figura 24 – Tela Inicial. Fonte: Geração própria.

Na primeira opção, o usuário é direcionado à **Tela de Modelos**, onde seleciona um dos três modelos disponíveis, como mostra na Figura 25.



Figura 25 – Tela de Modelos. Fonte: Geração própria.

Após selecionar um modelo, o usuário é direcionado à **Tela de Dimensão**, onde visualiza as informações das dimensões com descritivos, como mostra na Figura 26.

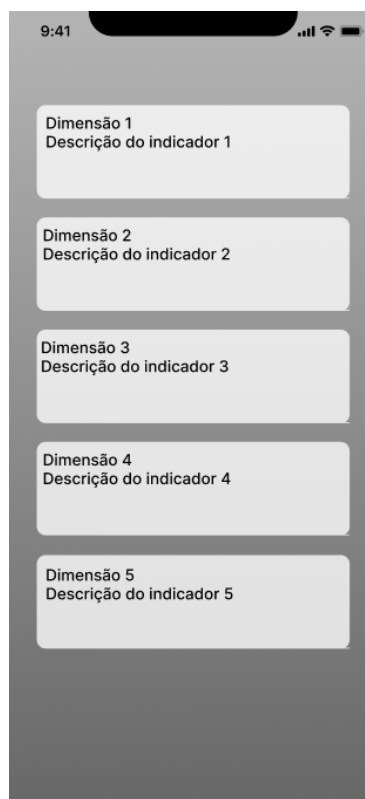


Figura 26 – Tela de Dimensão. Fonte: Geração própria.

Na segunda opção, o usuário é direcionado à **Tela de Comparativo**, onde é possível selecionar modelo, ano e as cidades para fazer o comparativo entre cada cidade de acordo com as dimensões presentes no modelo, como mostra na Figura 27.

9:41

Modelo:
Cidades Inteligentes

Ano:
2026

Selecionar cidade:
Caxias do Sul

Cidades selecionadas:

Bento Gonçalves	-
Farroupilha	-

Fazer comparativo

Figura 27 – Tela de Comparativo. Fonte: Geração própria.

Após a seleção de modelo, ano e cidades, o botão de fazer comparativo direciona o usuário à **Tela de Resultado**. Onde é exibido um painel com as informações da tela anterior, um gráfico de radar e também uma tabela/*cards* com informativos de cada dimensão de cada cidade selecionada, como mostra na Figura 28

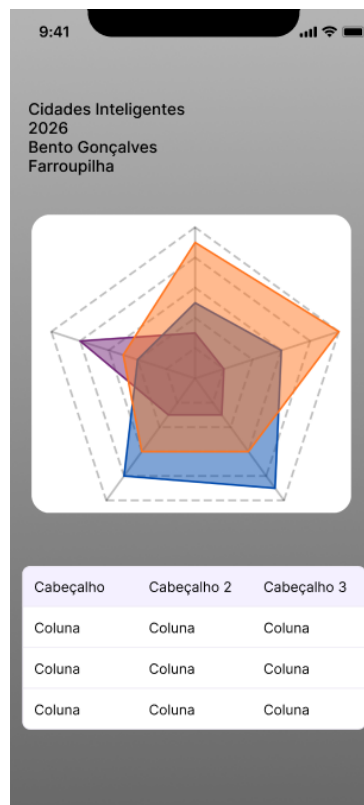


Figura 28 – Tela de Resultado. Fonte: Geração própria.

Esse fluxo foi planejado para proporcionar uma experiência de navegação fluida e orientada por contexto, permitindo que o usuário avance de maneira clara e lógica entre as etapas e escolha entre diferentes formas de análise conforme sua necessidade.

3.2 ARQUITETURA DA APLICAÇÃO

Para este projeto a arquitetura será composta por três camadas principais: interface de usuário, lógica da aplicação e camada de dados. A solução será desenvolvida com o framework Flutter, permitindo a construção de um único código-base para as plataformas Android e iOS.

A lógica da aplicação será responsável pela navegação, aplicação de filtros e manipulação dos dados. A camada de dados será composta por uma base previamente construída em PostgreSQL e as atualizações dela dependerão de atualizações e/ou disposições de novos dados inseridos nela. A arquitetura adotada segue o modelo cliente-servidor, no qual o dispositivo móvel assume o papel de cliente e o servidor de dados estará disposto via API externa.

A aplicação será estruturada em duas camadas principais de responsabilidade técnica: o *frontend* e o *backend*, ambos implementados no ambiente Flutter. O frontend corresponde à interface gráfica apresentada ao usuário, desenvolvida com os componentes visuais do Flutter, utilizando widgets customizados para garantir responsividade, navegação fluida e adaptação a diferentes tamanhos de tela. Essa camada também inclui os mecanismos de interação com filtros, seleção de modelos, visualização de dashboards e controle de navegação entre as telas.

O *backend*, por sua vez, será responsável por tratar as requisições internas geradas pelo *frontend*, acessando e manipulando os dados armazenados no banco PostgreSQL. Essa camada consumirá os *endpoints* da API. O *backend* também cuidará da lógica de filtragem, carregamento de dados normalizados e geração de dados para visualização, separando a lógica de negócio da apresentação. A comunicação entre essas camadas será mediada por controladores e serviços internos, respeitando os princípios de modularidade e desacoplamento definidos na arquitetura REST.

Será utilizado um gerenciador de estado próprio do *framework* Flutter. Também serão aplicadas práticas de adaptação para diferentes resoluções de tela e otimização do build, com uso de assets comprimidos, carregamento sob demanda e outras medidas que favoreçam o desempenho geral da aplicação. Podemos visualizar a lógica adotada para a arquitetura na Figura 29.

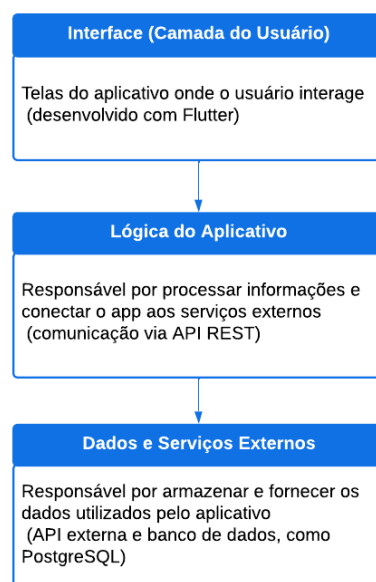


Figura 29 – Arquitetura para o projeto. Fonte: Geração própria.

A seguir, são descritos os principais elementos da arquitetura da solução proposta:

- **Desenvolvimento multiplataforma com Flutter:** a aplicação será implementada utilizando o framework Flutter, que possibilita o desenvolvimento de aplicativos nativos para Android e iOS a partir de uma única base de código.
- **Arquitetura cliente-servidor:** a plataforma móvel seguirá o modelo cliente-servidor, com consulta em um banco de dados disposto via API externa.
- **Comunicação via REST:** a interação entre módulos da aplicação será organizada por meio de endpoints baseados no padrão REST, garantindo modularidade, desacoplamento e aderência a boas práticas de projeto.

- **Integração de modelos conceituais:** a arquitetura incorpora, como base estrutural, os modelos do Sistema de Capitais, dos Objetivos de Desenvolvimento Sustentável (ODS) e do conceito de Cidades Inteligentes.
- **Interface adaptada ao ambiente mobile:** o design e a usabilidade da aplicação foram concebidos com foco específico em dispositivos móveis.
- **Dashboards interativos:** a aplicação contará com gráficos e painéis dinâmicos, com filtros por cidade, período e categoria temática.
- **Atualização da base de dados:** os dados consumidos pela aplicação dependem de uma base externa atualizada de forma independente pela infraestrutura do sistema, garantindo a integridade e a disponibilidade das informações mais recentes via API.

Essa arquitetura visa garantir a escalabilidade, robustez e aderência aos princípios já adotados na plataforma, ao mesmo tempo em que proporciona flexibilidade para futuras evoluções e integração com novos contextos analíticos.

3.3 MODELAGEM DOS DADOS

A estruturação dos dados é parte essencial para o funcionamento da plataforma mobile, pois garante a organização, consistência e possibilidade de expansão da base de indicadores urbanos. Esta modelagem tem origem no projeto descrito por BREGALDA (2021), que propôs uma estrutura inicial voltada à automação da carga de dados. Posteriormente, essa base foi significativamente adaptada e aprimorada por MARIA (2024), resultando no modelo atualmente utilizado e que servirá como referência para a implementação mobile.

O modelo de dados desenvolvido por MARIA (2024) foi projetado para suportar diferentes modelos conceituais de análise (Sistema de Capitais, ODS e Cidades Inteligentes), bem como permitir a consulta comparativa entre cidades. A arquitetura é baseada em um banco de dados relacional PostgreSQL, com tabelas organizadas por temas e funcionalidades que garantem a integridade e a escalabilidade das informações.

O núcleo da estrutura de dados é composto pela tabela valor, que armazena os dados brutos e normalizados dos indicadores por município, ano e modelo de referência. Essa tabela relaciona os valores com as demais entidades do sistema, como os municípios cadastrados na tabela região, os indicadores presentes na tabela indicador e os respectivos modelos conceituais representados na tabela modelo. A organização dos dados nesse formato permite a comparação entre cidades e o agrupamento temático com base nos modelos adotados (como o Sistema de Capitais, ODS e Cidades Inteligentes). A estrutura é complementada pelas tabelas dimensão e indicador_dimensao, que estabelecem a vinculação entre os indicadores e os eixos analíticos de cada modelo.

Contudo, as tabelas definidas por MARIA (2024) incluem:

- **modelo:** armazena os diferentes modelos de análise urbana disponíveis na plataforma, como Sistema de Capitais, Objetivos de Desenvolvimento Sustentável (ODS) e Cidades Inteligentes. Cada modelo possui um nome, sigla e identificador único.
- **dimensao:** representa as dimensões que compõem cada modelo (por exemplo, Educação, Mobilidade, Governança). Está relacionada a um modelo e contém nome e descrição.
- **indicador:** contém os indicadores utilizados na análise dos dados urbanos. Cada registro representa um indicador específico com seu nome e identificador.
- **indicador_dimensao:** tabela intermediária que estabelece a relação entre indicadores e dimensões, permitindo que um indicador pertença a uma ou mais dimensões e vice-versa.
- **indicador_referencial:** armazena os valores médios de referência dos indicadores por dimensão, usados para normalização e comparação entre cidades.
- **regiao:** registra os municípios presentes na plataforma, com atributos como código IBGE, nome do município e estado.
- **valor:** armazena os dados efetivos dos indicadores para cada cidade, ano e modelo. Inclui os valores brutos e os valores normalizados, permitindo a comparação entre municípios ao longo do tempo.

Essa estrutura permite maior controle sobre o processo de ETL, reduz o risco de erros e facilita a escalabilidade e a manutenção do sistema de dados no futuro. O modelo também favorece a integração de novas fontes de dados com esforço reduzido, e está alinhado com as necessidades de visualização e filtragem temáticas utilizadas pela plataforma. O modelo criado por MARIA (2024) pode ser observado na Figura 30.

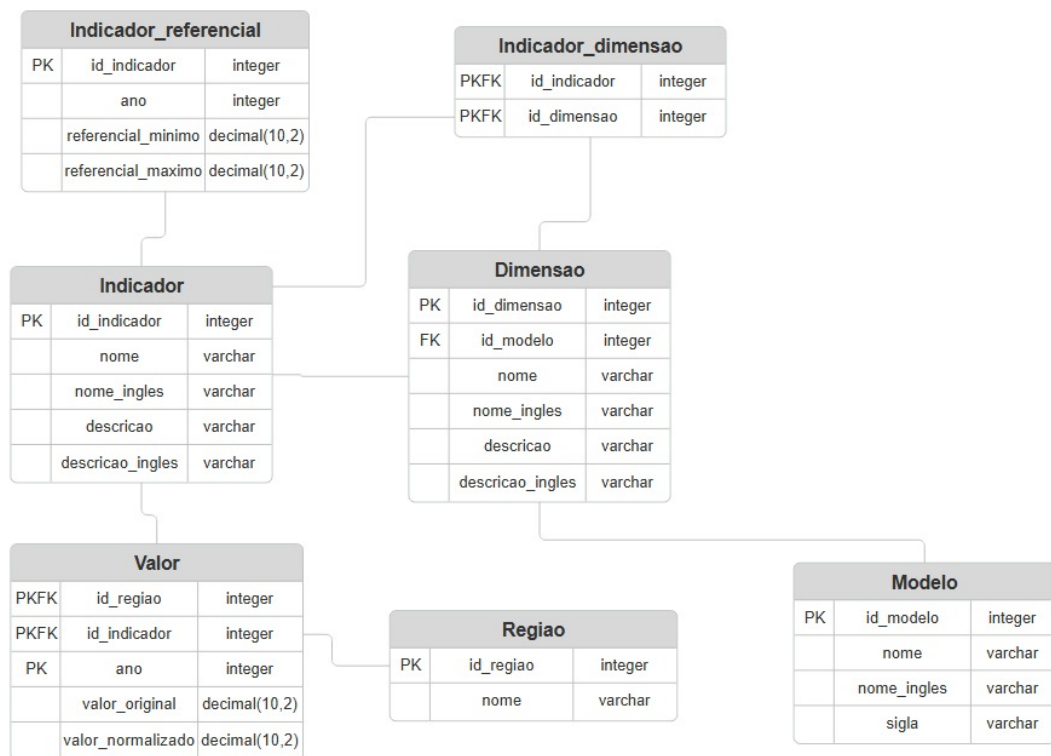


Figura 30 – Modelo de dados proposto para a plataforma. Fonte: MARIA (2024).

A aplicação mobile utilizará essa estrutura como base, sendo alimentada por meio do consumo de uma API externa que faz a interface direta com o banco de dados PostgreSQL. Os dados serão organizados de modo compatível com a plataforma web, porém otimizados para o consumo e processamento local por dispositivos móveis. Esse modelo de sincronização permitirá que os dados embarcados realizem:

- Consulta e comparação entre cidades;
- Visualização de gráficos (como radar) e tabelas associadas;
- Análise temporal de indicadores por cidade;
- Navegação orientada por filtros temáticos e geográficos;
- Escolha entre os modelos conceituais já estabelecidos.

A adoção de uma estrutura validada por projetos anteriores garante compatibilidade com o histórico da plataforma e assegura uma base sólida para análises urbanas consistentes e reproduzíveis. Além disso, a reutilização e adaptação da modelagem proposta por MARIA (2024) acelera o desenvolvimento mobile e assegura continuidade metodológica entre as diferentes versões da aplicação.

4 IMPLEMENTAÇÃO DA PROPOSTA DE SOLUÇÃO

Neste capítulo, será detalhada a implementação da solução mobile desenvolvida para a aplicação do projeto Cidades Inteligentes. Serão apresentados aspectos relacionados ao desenvolvimento da aplicação, incluindo as bibliotecas utilizadas, a estrutura do projeto e a organização da codificação.

Para o desenvolvimento da aplicação mobile, foi utilizado o framework Flutter com a linguagem Dart, mantendo a integração com o banco de dados PostgreSQL por meio da comunicação com a API. O ambiente de desenvolvimento utilizado foi o Visual Studio, escolhido por oferecer ferramentas integradas para depuração, emulação de dispositivos e gerenciamento do projeto Flutter, contribuindo para um fluxo de desenvolvimento mais eficiente.

A interface da aplicação foi desenvolvida utilizando os componentes nativos disponibilizados pelo framework Flutter. Durante o desenvolvimento, foi adotado um padrão de nomenclatura e organização de arquivos com o objetivo de facilitar a reutilização de métodos e chamadas, a manutenção do código e a escalabilidade da aplicação.

4.1 FERRAMENTAS E BIBLIOTECAS UTILIZADAS

Esta seção apresenta as principais ferramentas, frameworks e bibliotecas utilizadas no desenvolvimento da aplicação mobile para o projeto Cidades Inteligentes. As tecnologias adotadas foram fundamentais para possibilitar a adaptação da plataforma web para o contexto mobile, proporcionando uma experiência de uso mais acessível, responsiva e eficiente em dispositivos móveis.

- Flutter: Framework principal utilizado no desenvolvimento da aplicação mobile. O Flutter possibilitou a criação de interfaces responsivas e multiplataforma a partir de uma única base de código, permitindo compatibilidade com dispositivos Android e iOS.

- Dart: Linguagem de programação utilizada em conjunto com o Flutter para o desenvolvimento da aplicação. Sua integração nativa com o framework contribuiu para maior produtividade e organização do código.

- API REST: Utilizada como camada de comunicação entre a aplicação mobile e o backend da plataforma original. Por meio das requisições HTTP, a aplicação realiza consultas e obtém os dados necessários para exibição dos indicadores, modelos e demais informações presentes no sistema.

- PostgreSQL: Banco de dados utilizado pela aplicação backend para armazenamento das informações relacionadas aos indicadores, dimensões, modelos e demais dados do sistema. O acesso ao banco de dados ocorre indiretamente por meio da API utilizada pela aplicação

mobile.

- HTTP: Biblioteca utilizada para realizar as requisições HTTP a API, permitindo o consumo dos endpoints responsáveis pelo fornecimento de dados para aplicação mobile.
- Flutter Material Design: Conjunto de componentes visuais utilizado para construção da interface da aplicação, garantindo padronização visual, responsividade e melhor experiência de navegação em dispositivos móveis.
- fl_chart: Biblioteca utilizada para geração do gráfico de radar dentro da aplicação, permitindo a visualização dos indicadores de forma mais intuitiva e adaptada ao ambiente mobile.
- Visual Studio Code com extensões Flutter e Dart: Ambiente de desenvolvimento utilizado na implementação da aplicação. As extensões forneceram suporte à linguagem Dart, execução da aplicação, depuração, gerenciamento de dependências e integração com emuladores e dispositivos físicos, contribuindo para um fluxo de desenvolvimento mais eficiente.

A Figura 31 apresenta o diagrama com a arquitetura da aplicação mobile. O modelo adotado é baseado na separação entre interface, lógica de aplicação e consumo de serviços externos por meio de API REST. O fluxo de informações ocorre entre o usuário, a interface mobile desenvolvida em Flutter, a camada de serviços responsável pelas requisições HTTP e o backend da aplicação, que realiza o acesso ao banco de dados PostgreSQL.

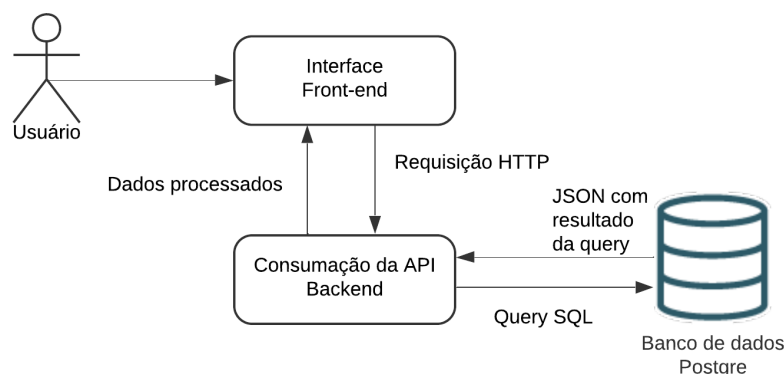


Figura 31 – Diagrama com a ideia geral da aplicação.

Fonte: O autor.

4.2 ESTRUTURA DA APLICAÇÃO

A aplicação foi desenvolvida utilizando uma arquitetura cliente-servidor baseada em comunicação via HTTP e integração com banco de dados PostgreSQL. Diferentemente da versão anterior do sistema, desenvolvida para ambiente web utilizando o padrão MVC (Model-View-Controller) com Django, a nova versão concentra a interface e a interação com o usuário inteiramente na aplicação, enquanto o servidor é responsável pela execução das consultas no

banco de dados e retorno das informações ao aplicativo.

Nessa arquitetura, o Flutter atua como camada de apresentação da aplicação, sendo responsável pela interface gráfica, captura das entradas do usuário e exibição dinâmica dos dados recebidos. A comunicação entre o aplicativo e o servidor ocorre através de requisições HTTP, nas quais são enviadas as consultas SQL ou apenas o nome da tabela da qual queremos obter os dados. Após o processamento das consultas no banco de dados PostgreSQL, os resultados retornados pela API são enviados em formato JSON e apresentados na interface do aplicativo.

A arquitetura da aplicação pode ser dividida em três camadas principais:

- **Camada de Apresentação:** Desenvolvida em Flutter, é responsável pela interface gráfica da aplicação e pela interação com o usuário. Essa camada permite a seleção de modelos, indicadores, cidades e períodos de análise, além da exibição dos resultados em cartões, tabelas e gráficos interativos, como o gráfico radar utilizado nas comparações entre cidades.
- **Camada de Comunicação:** Responsável pelo envio e recebimento das requisições HTTP entre o aplicativo e o servidor. Essa camada realiza a troca de dados de forma assíncrona, permitindo atualização dinâmica das informações exibidas ao usuário.
- **Camada de Dados:** Composta pelo banco de dados PostgreSQL, responsável pelo armazenamento e gerenciamento das informações utilizadas pela aplicação.

As principais entidades utilizadas no banco de dados incluem:

- **Indicador:** tabela que contém os indicadores disponíveis para análise, contendo atributos como identificador e nome.
- **Dimensão:** representa os agrupamentos temáticos dos indicadores dentro de cada modelo de cidade inteligente.
- **Modelo:** define os diferentes modelos de cidades inteligentes utilizados nas comparações realizadas pela aplicação.

Além das tabelas estruturais, o banco de dados PostgreSQL da aplicação possui tabelas dinâmicas de indicadores, identificadas pelo prefixo `db_`. Essas tabelas armazenam os dados utilizados nas análises comparativas entre cidades e podem ser adicionadas ou modificadas diretamente no banco de dados, sem a necessidade de alterações no código-fonte da aplicação, desde que sua estrutura siga o padrão esperado pelo sistema.

O fluxo de funcionamento da aplicação ocorre da seguinte forma:

- O usuário interage com a interface do aplicativo Flutter, realizando seleções de cidades, modelos, indicadores e ano de análise.
- O aplicativo processa essas entradas e envia requisições HTTP contendo os parâmetros necessários para consulta dos dados, incluindo a consulta SQL utilizada para recuperação das informações no banco de dados.

- O servidor recebe a requisição, executa a consulta SQL correspondente no banco de dados PostgreSQL e processa os resultados obtidos.
- Após o processamento, os dados são retornados ao aplicativo em formato JSON.
- O Flutter interpreta os dados recebidos e os apresenta ao usuário de forma visual por meio de cartões informativos, tabelas e gráfico de radar.

Essa arquitetura permite separar a camada de apresentação da camada de armazenamento dos dados, proporcionando maior flexibilidade e escalabilidade para a aplicação. Além disso, o uso de comunicação via HTTP e dados em formato JSON facilita futuras integrações e atualizações tanto na interface móvel quanto na estrutura dos dados armazenados no PostgreSQL.

4.3 AJUSTES EM COMPARAÇÃO A APLICAÇÃO WEB

A principal proposta deste trabalho foi o desenvolvimento de uma aplicação mobile utilizando Flutter, esta que deverá ser como uma extensão da plataforma web já existente. Enquanto anteriormente o sistema estava disponível apenas em ambiente web, a nova solução possibilita o acesso às funcionalidades também por dispositivos móveis, ampliando a acessibilidade na utilização da plataforma. Podemos listar que as principais mudanças e adaptações, em relação a plataforma web, foram:

- Utilização de uma API para comunicação com o banco de dados: Diferentemente da aplicação web, que realiza acesso direto ao banco de dados, a aplicação mobile passou a utilizar uma API exclusiva como intermediária para obtenção das informações. Essa camada de comunicação foi desenvolvida com o objetivo de centralizar o acesso aos dados, organizar as requisições realizadas pelo aplicativo e garantir maior segurança e desacoplamento entre a aplicação mobile e o banco de dados.
- Adaptação da plataforma para dispositivos móveis: Durante o desenvolvimento da aplicação mobile, buscou-se manter a mesma estrutura funcional e organizacional já presente na plataforma web moderna. Dessa forma, as telas e funcionalidades permaneceram, em síntese, equivalentes às da versão web, permitindo que os usuários tivessem familiaridade com a navegação e utilização do sistema também em dispositivos móveis.
- Adequação da interface para diferentes tamanhos de Tela: Embora a organização geral das telas tenha sido mantida, foram necessárias adequações na interface para garantir compatibilidade com dispositivos móveis. Elementos visuais, espaçamentos e componentes de navegação foram ajustados para proporcionar melhor visualização e usabilidade em *smartphones*, respeitando as limitações de espaço das telas menores.
- Reaproveitamento das funcionalidades da plataforma Web: A aplicação mobile foi desenvolvida com base nas funcionalidades já existentes na plataforma web, preservando as regras

de negócio e o comportamento do sistema. Essa abordagem permitiu manter consistência entre as versões da plataforma, além de reduzir retrabalho durante o processo de desenvolvimento.

A aplicação mobile foi desenvolvida mantendo compatibilidade com a mesma estrutura de banco de dados utilizada pela plataforma web desenvolvida por MARIA (2024) preservando o modelo relacional já existente. Dessa forma, não foram necessárias alterações estruturais no banco de dados para suportar a versão mobile da aplicação, mantendo inalteradas as regras relacionadas à inclusão de novas tabelas e alteração ou inclusão de novos dados, garantindo integração e consistência entre as diferentes versões da plataforma.

4.4 ATUALIZAÇÃO DE BIBLIOTECAS E CARREGAMENTO DA APLICAÇÃO

Para executar um projeto *Flutter*, é necessário garantir que o ambiente de desenvolvimento esteja corretamente configurado, incluindo a instalação do *Flutter SDK*. A verificação da instalação pode ser realizada por meio do comando `flutter doctor`, que identifica possíveis problemas no ambiente e indica as ações necessárias para sua correção.

Após a configuração do ambiente, o repositório deve ser clonado para a máquina local utilizando o comando `git clone <URL_DO_REPOSITORIO>`. Em seguida, é necessário acessar o diretório raiz da aplicação por meio do comando `cd nome_do_projeto`.

Com o projeto aberto, as dependências devem ser instaladas utilizando o comando `flutter pub get`. Esse processo realiza o download de todas as bibliotecas definidas no arquivo `pubspec.yaml`, garantindo o funcionamento da aplicação.

Caso seja necessário atualizar as dependências para versões mais recentes, pode-se utilizar o comando `flutter pub upgrade`, responsável por atualizar os pacotes de acordo com as versões disponíveis e compatíveis com o projeto.

Por fim, a execução da aplicação em um emulador ou dispositivo físico conectado pode ser realizada por meio do comando `flutter run`, que compila e inicia a aplicação.

4.5 IMPLEMENTAÇÃO DA APLICAÇÃO MOBILE

A escolha por Flutter permitiu estruturar a interface em componentes reutilizáveis, manter uma base única de código para diferentes plataformas e integrar recursos visuais, navegação e comunicação com serviços externos em uma mesma arquitetura de desenvolvimento. No projeto, a organização do código foi separada em camadas funcionais: as telas da aplicação estão concentradas no diretório `screens`, a lógica de acesso e processamento de dados está no diretório `service`, a autenticação está em `security`, os modelos de dados em `models` e a padronização visual em `theme`.

Essa organização buscou separar responsabilidades entre apresentação e regra de negócio. As telas ficaram responsáveis pela interação com o usuário, exibição dos dados e navegação entre funcionalidades. Os `services` concentraram as requisições HTTP, a construção das consultas e o tratamento dos dados retornados pela API. Dessa forma, o fluxo da aplicação parte da interface, aciona um `service` específico e atualiza o estado da tela conforme os dados são carregados.

4.5.1 Inicialização e autenticação

O ponto de entrada da aplicação realiza a preparação inicial antes da exibição da interface principal. Ao iniciar o aplicativo, o `service` de autenticação é instanciado e executa a obtenção do token de acesso. Após essa etapa, o Flutter renderiza o componente principal da aplicação e também a tela inicial de abertura.

Algoritmo 1 – Inicialização da aplicação e preparação do token de acesso.

```
1 void main() async {  
2   final authService = AuthService();  
3   await authService.fetchAndStoreToken();  
4  
5   runApp(MyApp());  
6 }
```

O `service` de autenticação utiliza uma requisição HTTP do tipo *POST* para obter um token junto ao endpoint de login. Quando a resposta é bem-sucedida, o token retornado é armazenado localmente por meio da biblioteca `shared_preferences`. Essa estratégia evita que cada tela precise repetir o processo de login, pois os demais serviços podem recuperar o token salvo e utilizá-lo no cabeçalho `Authorization` das consultas seguintes. No trecho a seguir, as credenciais reais foram omitidas por segurança.

Algoritmo 2 – Armazenamento local do token de autenticação.

```
1 final response = await http.post(  
2   Uri.parse(url),  
3   headers: {'Content-Type': 'application/json'},  
4   body: jsonEncode({  
5     'email': '<email>',  
6     'password': '<senha>',  
7   })),  
8 );  
9  
10 if (response.statusCode == 200) {  
11   final data = jsonDecode(response.body);  
12   final token = data['token'];  
13  
14   if (token != null) {  
15     final prefs = await SharedPreferences.getInstance();  
16     await prefs.setString('auth_token', 'Bearer_$token');  
17   }  
18 }
```

Esse procedimento define uma etapa central para a comunicação com a base de dados, pois as consultas posteriores dependem do token salvo para serem autorizadas pela API.

4.5.2 Estrutura visual e navegação

A interface foi construída com componentes do Material Design e com uma identidade visual definida no arquivo de tema da aplicação. Nesse arquivo foram centralizadas cores, gradientes, estilos de texto e configurações globais do *ThemeData*. A centralização desses elementos contribuiu para manter consistência visual entre as diferentes telas, como tela inicial, listagem de indicadores, seleção de cidades e resultado comparativo.

Após a inicialização, a aplicação apresenta uma tela de abertura com um *loading* e em seguida, a navegação é direcionada para a tela inicial, que concentra os principais caminhos de uso: exploração de indicadores, comparação de cidades e acesso a informações sobre o projeto. Essa estrutura atua como menu principal e organiza as funcionalidades conforme a finalidade de uso.

4.5.3 Consulta de modelos, indicadores e dimensões

A funcionalidade de exploração de indicadores tem como objetivo apresentar os modelos disponíveis na base de dados e permitir que o usuário visualize suas respectivas dimensões. Para isso, a tela de indicadores executa uma consulta ao endpoint de consultas da API, recuperando os registros da tabela de modelos. A consulta exclui o modelo de identificador 4, mantendo apenas os modelos utilizados pela interface.

Algoritmo 3 – Consulta dos modelos disponíveis para exibição na tela de indicadores.

```
1 final response = await http.post(  
2   url ,  
3   headers: headers ,  
4   body: 'SELECT * FROM sys_modelo WHERE id_modelo != 4',  
5 );  
6  
7 if (response.statusCode == 200) {  
8   setState(() {  
9     dados = json.decode(response.body);  
10    isLoading = false;  
11  });  
12 }
```

Os resultados são apresentados em uma lista de cartões. Cada cartão representa um modelo, exibindo nome, nome em inglês e sigla. Ao selecionar um modelo, a aplicação navega para a tela de dimensões, repassando o identificador do modelo selecionado. A tela de dimensões, por sua vez, executa uma nova consulta filtrada por *id_modelo*, retornando as dimensões associadas àquele modelo.

Algoritmo 4 – Consulta de dimensões associadas ao modelo selecionado.

```
1 final response = await http.post(  
2   url ,  
3   headers: headers ,  
4   body: 'SELECT * FROM sys_dimensao WHERE id_modelo = ${widget.idModelo}',  
5 );
```

Essa implementação permite que a estrutura conceitual dos modelos seja explorada de forma progressiva: primeiro o usuário escolhe o modelo de análise, depois visualiza suas dimensões e descrições. Assim, a aplicação também cumpre uma função informativa, pois apresenta a organização dos indicadores antes da comparação numérica entre municípios.

4.5.4 Seleção de cidades para comparação

A funcionalidade principal da aplicação é o comparativo de cidades. O fluxo inicia na tela *ComparativoDeCidadesPage*, na qual o usuário seleciona o modelo de análise, o ano de referência e os municípios que deseja comparar. Os modelos disponíveis nessa tela são definidos como uma lista local contendo três opções: Sistema de Capitais, Cidades Inteligentes e Objetivos do Desenvolvimento Sustentável.

Algoritmo 5 – Modelos de análise disponíveis no comparativo.

```
1 final modelos = [  
2   {"id": "1", "nome": "Sistema de Capitais"},  
3   {"id": "2", "nome": "Cidades Inteligentes"},
```

```

4   {"id": "3", "nome": "Objetivos do Desenvolvimento Sustentavel"},
5 ];

```

A seleção do ano é gerada dinamicamente a partir de 1989 até o ano atual do dispositivo. Essa decisão evita a manutenção manual da lista de anos e permite que a aplicação continue atualizada conforme novos anos se tornam disponíveis.

Algoritmo 6 – Geração dinâmica da lista de anos.

```

1 List<String> gerarListaAnos() {
2     final anoAtual = DateTime.now().year;
3     return List.generate(
4         (anoAtual - 1989) + 1,
5         (index) => (1989 + index).toString(),
6     );
7 }

```

Para selecionar municípios, a aplicação consulta a tabela *sys_regiao* por meio do serviço *Utils*. O retorno é usado para preencher uma janela inferior com campo de busca, permitindo filtrar os municípios pelo nome. O usuário pode selecionar de dois a cinco municípios, regra validada antes da navegação para a tela de resultado.

Algoritmo 7 – Validação antes de iniciar o comparativo.

```

1 void irResultado() {
2     if (cidadesSelecionadas.length < 2) {
3         mostrarErro(context, "Selecione pelo menos 2 cidades.");
4         return;
5     }
6     if (cidadesSelecionadas.length > 5) {
7         mostrarErro(context, "Voce pode selecionar no maximo 5 cidades.");
8         return;
9     }
10    if (modeloSelecionado == null || anoSelecionado == null) {
11        mostrarErro(context, "Selecione um modelo e um ano.");
12        return;
13    }
14 }

```

Essa validação é importante para garantir que o comparativo tenha sentido analítico, pois exige ao menos duas cidades e limita a quantidade máxima para preservar a legibilidade dos gráficos e listas de resultados.

4.5.5 Serviço de comparação e processamento dos dados

A regra de negócio do comparativo está concentrada no serviço *ComparativoResultadoService*. Esse serviço é responsável por consultar a configuração do modelo selecionado, re-

cuperar as dimensões e indicadores relacionados, filtrar os dados das cidades no ano escolhido, agrupar os resultados por dimensão e calcular as médias normalizadas.

O método principal recebe três parâmetros: a lista de cidades selecionadas, o modelo escolhido e o ano de referência. A partir deles, o serviço carrega a configuração das dimensões do modelo e monta estruturas auxiliares que relacionam dimensões e indicadores.

Algoritmo 8 – Assinatura do método principal do comparativo.

```
1 static Future<Map<String , dynamic>> gerarResultadoComparativo({
2     required List<Map<String , dynamic>> cidadesSelecionadas ,
3     required String modeloSelecionado ,
4     required String anoSelecionado ,
5 }) async {
6     final configuracaoDimensoes =
7         await carregarConfiguracaoDimensoes(modeloSelecionado);
8
9     final dimensoes = List<Map<String , dynamic>>.from(
10         configuracaoDimensoes["dimensoes"] as List ,
11     );
12
13     final todosIndicadores = List<dynamic>.from(
14         configuracaoDimensoes["todos_indicadores"] as List ,
15     );
16 }
```

A configuração é obtida a partir da relação entre dimensões e indicadores. A consulta une as tabelas *sys_dimensao* e *sys_indicadordimensao*, filtrando pelo modelo selecionado. Com o resultado, o serviço monta quatro estruturas: lista de dimensões, indicadores por dimensão, dimensões por indicador e conjunto total de indicadores.

Algoritmo 9 – Consulta da configuração de dimensões e indicadores do modelo.

```
1 final relacoes = await _postQuery(" "
2     SELECT d.id_dimensao ,
3           d.nome AS dimensao_nome ,
4           i.id_indicador
5     FROM sys_dimensao d
6     JOIN sys_indicadordimensao i
7         ON i.id_dimensao = d.id_dimensao
8     WHERE d.id_modelo = ${_toSqlString(modeloSelecionado)}
9     ORDER BY d.nome, i.id_indicador
10 " " );
```

Após identificar quais indicadores pertencem ao modelo, o serviço consulta as tabelas da base que seguem o padrão *db_%*. Em seguida, cada tabela é filtrada conforme as cidades selecionadas, os indicadores relacionados e o ano escolhido. O resultado de cada consulta contém o código IBGE, o ano, o valor normalizado e o identificador do indicador.

Algoritmo 10 – Filtro aplicado aos dados de cada tabela.

```
1 final lista = await _postQuery("""
2     SELECT DISTINCT
3         codigo_ibge ,
4         ano ,
5         valor_normalizado ,
6         indicador
7 FROM $nomeTabela
8 WHERE codigo_ibge IN ($cidades)
9     AND indicador IN ($indicadores)
10    AND ano = '$anoSelecionado'
11 """);
```

Depois da consulta, os dados são enriquecidos com informações do município e com o nome do indicador. Para isso, o serviço carrega previamente as regiões das cidades selecionadas e os nomes dos indicadores. Essa etapa transforma os registros numéricos em dados mais adequados para exibição, pois a tela pode apresentar nomes de municípios, estados e indicadores em vez de apenas códigos.

O agrupamento por dimensão ocorre a partir da relação entre indicador e dimensão. Para cada dado retornado da base, o serviço verifica a quais dimensões aquele indicador pertence e adiciona o registro ao grupo correspondente.

Algoritmo 11 – Agrupamento dos registros conforme a dimensão do indicador.

```
1 for (final dado in dados) {
2     final dimensoes = dimensoesPorIndicador[dado["indicador"].toString()];
3
4     if (dimensoes == null) {
5         continue;
6     }
7
8     for (final dimensao in dimensoes) {
9         final idDimensao = dimensao["id_dimensao"].toString();
10        agrupados.putIfAbsent(idDimensao, () => <Map<String, dynamic>>[]);
11        agrupados[idDimensao]!.add(dado);
12    }
13 }
```

Por fim, o serviço calcula a média dos valores normalizados por cidade e por dimensão. Para isso, cada registro contribui com seu *valor_normalizado* para uma chave composta por município, identificador da dimensão e nome da dimensão. Ao final, a soma acumulada é dividida pela quantidade de indicadores considerados.

Algoritmo 12 – Cálculo da média normalizada por cidade e dimensão.

```
1 final chave =
```



```

2      "${dado["municipio"]}|${dim["id_dimensao"]}|${dim["nome"]}";
3
4      resultados.putIfAbsent(
5          chave,
6          () => {"soma": 0.0, "qtd": 0},
7      );
8
9      resultados[chave]["soma"] +=
10         double.tryParse(dado["valor_normalizado"].toString()) ?? 0;
11      resultados[chave]["qtd"] += 1;

```

O retorno do processamento é composto por duas estruturas principais: *dados_por_dimensao*, contendo os registros organizados para apresentação detalhada, e *resultados_medias*, contendo as médias utilizadas nos gráficos e na listagem comparativa.

4.5.6 Apresentação do resultado comparativo

A tela *ComparativoResultadoPage* é responsável por transformar o resultado produzido pelo serviço em uma visualização compreensível para o usuário. Ao ser criada, a tela executa o carregamento dos dados no método *initState*. Durante esse processo, apresenta uma interface de carregamento e, após a resposta do serviço, atualiza o estado com as dimensões, os dados detalhados e as médias calculadas.

Algoritmo 13 – Carregamento dos dados na tela de resultado.

```

1 Future<void> carregarDados() async {
2     final resultado = await ComparativoResultadoService.gerarResultadoComparativo(
3         cidadesSelecionadas: widget.cidades,
4         modeloSelecionado: widget.modeloId.toString(),
5         anoSelecionado: widget.ano,
6     );
7
8     setState(() {
9         dadosPorDimensao = List<Map<String, dynamic>>.from(
10             resultado["dados_por_dimensao"],
11         );
12         medias = Map<String, double>.from(resultado["resultados_medias"]);
13         loading = false;
14     });
15 }

```

A apresentação dos resultados foi dividida em seções. A primeira seção exibe gráficos e médias gerais; as demais seções correspondem às dimensões retornadas pelo modelo. Um menu lateral permite alternar entre a visualização geral e o detalhamento de cada dimensão. Essa solução evita que todos os dados sejam exibidos de uma só vez e facilita a navegação em comparativos com várias dimensões.

Na visualização geral, foi utilizada a biblioteca *fl_chart* para construir um gráfico radar. Cada cidade selecionada é representada por uma cor, e cada eixo do gráfico corresponde a uma dimensão. Os valores plotados são as médias normalizadas calculadas pelo serviço.

Algoritmo 14 – Construção dos dados utilizados no gráfico radar.

```
1 for (var cidade in widget.cidades) {
2   final nomeCidade = cidade["nome"];
3   List<double> valores = [];
4
5   for (var dim in dimensoes) {
6     final item = medias.entries.where((e) {
7       final split = e.key.split("|");
8       return split[0] == nomeCidade && split[2] == dim;
9     }).toList();
10
11     valores.add(item.isNotEmpty ? item.first.value : 0);
12   }
13
14   lista.add(
15     RadarDataSet(
16       dataEntries: valores.map((e) => RadarEntry(value: e)).toList(),
17     ),
18   );
19 }
```

A tela também oferece controle de zoom para o gráfico radar, permitindo ampliar ou reduzir a visualização. Esse recurso foi implementado com *InteractiveViewer* e *TransformationController*, melhorando a leitura dos eixos quando há muitas dimensões ou nomes extensos.

No detalhamento por dimensão, cada cidade é apresentada em um componente expansível. Ao expandir o item, o usuário visualiza a média da cidade naquela dimensão e os indicadores que compuseram o cálculo, juntamente com seus valores normalizados. Assim, a aplicação não se limita a mostrar uma pontuação final, mas também permite verificar quais dados sustentam o resultado apresentado.

Algoritmo 15 – Exibição detalhada dos indicadores de uma dimensão.

```
1 return Card(  
2   child: ExpansionTile(  
3     title: Text(cidade),  
4     subtitle: Text(nomeDimensao),  
5     children: [  
6       ListTile(title: Text("Media: ${item.value.toStringAsFixed(4)}"),  
7         ... dadosDimensao  
8         .where((d) => d["municipio"] == cidade)  
9         .map((dado) => ListTile(  
10           title: Text(dado["indicador_nome"]),  
11           subtitle: Text("Valor: ${dado["valor_normalizado"]}"),  
12         )),  
13     ],  
14   ),  
15 );
```

Essa forma de apresentação aproxima a leitura técnica da leitura visual: o gráfico radar oferece uma comparação sintética entre cidades, enquanto as seções por dimensão permitem observar a composição dos valores de maneira mais analítica.

4.5.7 Funcionalidades implementadas

Com base na implementação realizada, as funcionalidades principais da aplicação podem ser sintetizadas da seguinte forma:

- autenticação automática na API e armazenamento local do token de acesso;
- tela de abertura com animação e transição para a página inicial;
- menu inicial com acesso às áreas de exploração de indicadores, comparativo de cidades e informações do projeto;
- consulta e exibição dos modelos de indicadores disponíveis;
- consulta das dimensões associadas a cada modelo;
- seleção de modelo, ano e municípios para comparação;
- busca de municípios a partir da tabela de regiões;
- validação da quantidade de cidades selecionadas, com mínimo de duas e máximo de cinco;
- processamento dos dados normalizados por ano, cidade, indicador e dimensão;
- cálculo de médias normalizadas por cidade e dimensão;

- exibição comparativa em gráfico radar;
- listagem das médias por dimensão;
- detalhamento dos indicadores que compõem cada média;
- padronização visual por meio de tema, cores, gradientes e componentes reutilizáveis.

De modo geral, a implementação buscou transformar dados tabulares da base do Observatório em uma experiência mobile orientada à análise. O usuário seleciona parâmetros de comparação, a aplicação recupera e organiza os dados por meio dos serviços e, por fim, apresenta os resultados em visualizações sintéticas e detalhadas. Assim, o aplicativo atua como uma interface de apoio à leitura de indicadores urbanos, permitindo comparar municípios segundo diferentes modelos de avaliação e dimensões de análise.

4.6 CASOS DE USO DA APLICAÇÃO

Os casos de uso da aplicação representam as principais funcionalidades disponibilizadas ao usuário para consulta e análise dos dados urbanos presentes na plataforma. A interface foi desenvolvida para permitir o acesso direto a diferentes recursos, possibilitando tanto a exploração da estrutura dos indicadores quanto a realização de análises comparativas entre municípios.

A Figura 32 reúne as interfaces referentes à abertura da aplicação, tela inicial, exploração de indicadores e informações institucionais. Ao iniciar a aplicação, o usuário visualiza uma tela de abertura (Tela A indicada Figura 32) responsável pela inicialização dos recursos necessários para o funcionamento do sistema. Após esse processo, é apresentada a tela inicial da plataforma, que centraliza o acesso às principais funcionalidades disponíveis.

A tela inicial (Tela B indicada na Figura 32) reúne os principais recursos oferecidos pela plataforma, permitindo ao usuário acessar as funcionalidades de exploração dos indicadores, comparação entre municípios e consulta de informações institucionais sobre o projeto.

Uma dessas funcionalidades consiste na exploração dos indicadores cadastrados na plataforma. Para isso, a aplicação disponibiliza uma tela (Tela C indicada na Figura 32) que apresenta os modelos conceituais disponíveis na base de dados. Cada modelo é exibido em formato de cartão contendo informações como nome, sigla e descrição, permitindo ao usuário conhecer a estrutura conceitual utilizada pelo sistema.

Além da consulta aos modelos, a aplicação permite visualizar as dimensões associadas a cada estrutura conceitual (Tela D indicada na Figura 32). Nessa funcionalidade são apresentados o identificador, o nome e a descrição de cada dimensão, permitindo compreender como os indicadores estão organizados dentro de cada modelo e facilitando a interpretação dos resultados posteriormente utilizados nas análises comparativas.

Complementando esse fluxo exploratório, a aplicação apresenta uma tela contendo uma descrição geral da proposta do City Living Lab, seus objetivos e orientações sobre a utilização da plataforma (Tela E indicada na Figura 32). Essa funcionalidade contribui para contextualizar o propósito do sistema e sua aplicação no apoio à análise de indicadores urbanos.



Figura 32 – Telas de abertura, página inicial, indicadores, dimensões e informações sobre o projeto. Fonte: Geração própria.

O fluxo correspondente à comparação de municípios e a apresentação de seus resultados está consolidado na Figura 33. Para realizar essa análise, a aplicação disponibiliza uma interface na qual o usuário define o modelo conceitual de interesse, o ano de referência e os municípios que participarão da comparação. O sistema permite selecionar entre duas e cinco cidades, garantindo condições adequadas para a análise dos resultados e para a visualização das informações apresentadas (Tela A indicada na Figura 33).

Como parte desse processo, a aplicação oferece um mecanismo de busca para seleção de municípios. A funcionalidade disponibiliza consulta à lista de regiões e um campo de pesquisa que permite localizar cidades pelo nome, facilitando a composição do conjunto de municípios

que serão analisados (Tela B indicada na Figura 33). Além disso, os municípios selecionados podem ser removidos ou substituídos antes da geração do comparativo.

Após a definição dos parâmetros necessários, a aplicação processa os dados e apresenta os resultados da comparação. Inicialmente, é exibida uma área de carregamento enquanto as informações são obtidas e tratadas. Em seguida, são apresentados os dados comparativos referentes ao modelo, ao ano selecionado e aos municípios participantes da análise.

A principal visualização disponibilizada consiste em um gráfico radar, no qual cada eixo representa uma dimensão do modelo conceitual escolhido. As cidades analisadas são diferenciadas por cores e os valores apresentados correspondem às médias normalizadas calculadas para cada dimensão, permitindo uma comparação visual entre os municípios (Tela C indicada na Figura 33). A visualização gráfica também oferece recursos de ampliação, redução e recentralização, contribuindo para a análise de modelos que possuem maior quantidade de dimensões ou rótulos mais extensos. Complementando o gráfico, a aplicação apresenta uma legenda contendo a identificação das cidades analisadas.

Além da representação gráfica, a aplicação disponibiliza uma listagem das médias normalizadas por dimensão. Essa funcionalidade permite consultar diretamente os valores numéricos utilizados na comparação, fornecendo uma visão complementar à análise visual realizada por meio do gráfico radar (Tela D indicada na Figura 33).

A aplicação também oferece uma funcionalidade de detalhamento dos resultados por dimensão. Nessa visualização, cada município é apresentado juntamente com sua média na dimensão selecionada e com os indicadores que contribuíram para o cálculo desse resultado (Tela E indicada na Figura 33). Os indicadores são exibidos com seus respectivos valores normalizados, permitindo uma análise mais aprofundada da composição das médias apresentadas. Esse detalhamento permite compreender os fatores que influenciaram os resultados observados na comparação entre municípios, ampliando a capacidade de interpretação dos dados apresentados pela plataforma.

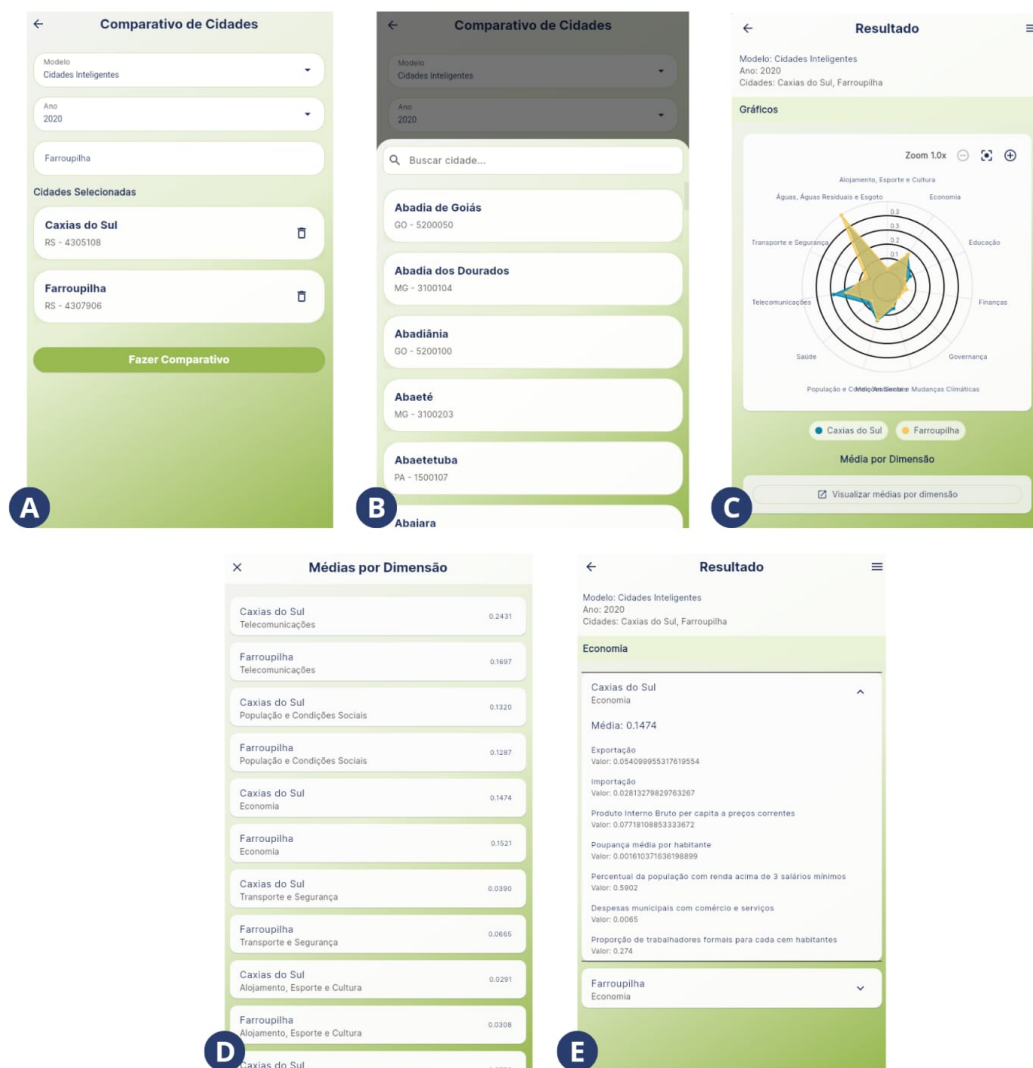


Figura 33 – Fluxo de comparação: seleção de cidades, gráfico radar, médias e detalhamento por dimensão. Fonte: Geração própria.

Os casos de uso apresentados demonstram as principais funcionalidades disponibilizadas pela aplicação. Por meio delas, o usuário pode consultar os modelos conceituais e suas respectivas dimensões, realizar análises comparativas entre municípios utilizando diferentes critérios de seleção e acessar informações relacionadas ao projeto. Essas funcionalidades contribuem para tornar a plataforma uma ferramenta de apoio à exploração e interpretação dos indicadores urbanos disponibilizados pelo City Living Lab.

4.6.1 Comparativo dos dados com a aplicação web

Com o objetivo de validar os resultados apresentados pela aplicação mobile desenvolvida, realizou-se um comparativo entre os dados exibidos na versão web da plataforma e aqueles gerados no aplicativo. Essa comparação permite verificar se as informações recuperadas, organizadas e apresentadas no ambiente mobile mantêm consistência em relação à aplicação já

existente, especialmente no que se refere aos indicadores, valores e formas de visualização dos dados.

A Figura 34 apresenta o gráfico gerado pela aplicação web, enquanto a Figura 35 demonstra o gráfico equivalente produzido pela aplicação mobile. Observa-se que ambas as versões apresentam os dados de forma compatível, permitindo ao usuário realizar a análise comparativa entre cidades a partir dos mesmos indicadores. A adaptação para o ambiente mobile, entretanto, exige uma organização visual mais compacta e responsiva, considerando as limitações de espaço da tela e a necessidade de preservar a legibilidade das informações.

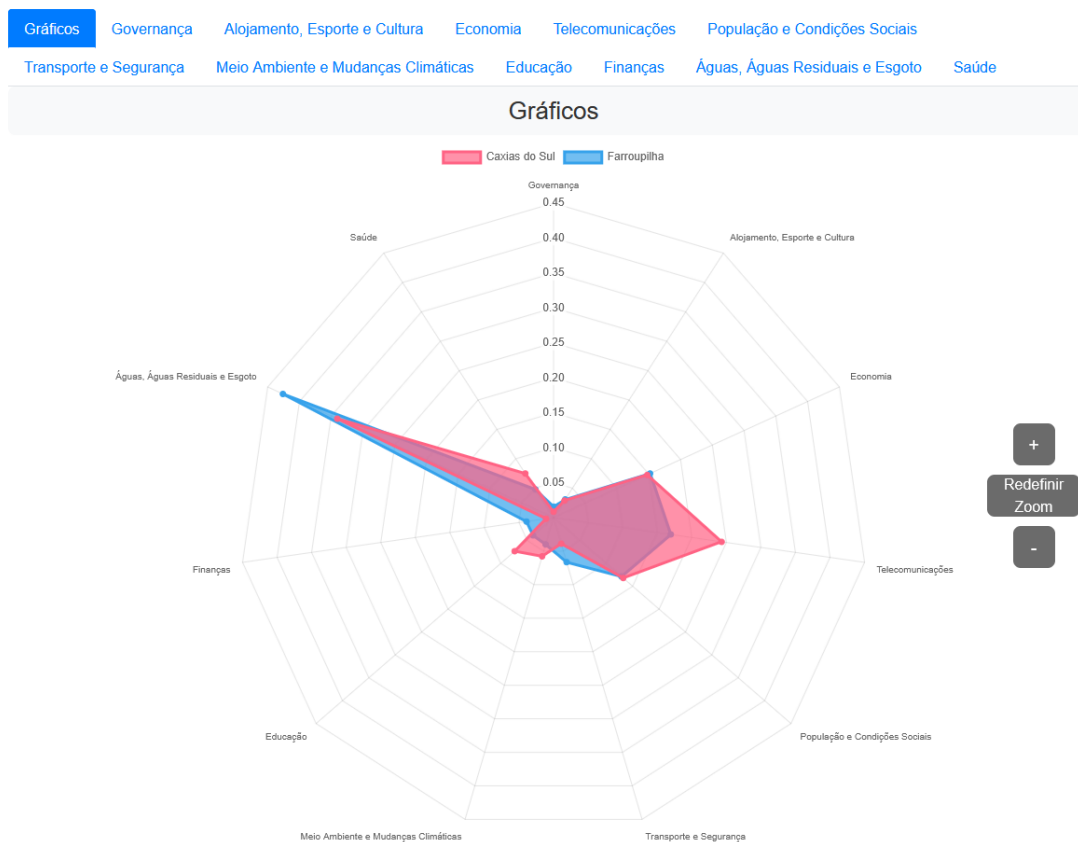


Figura 34 – Gráfico gerado pela aplicação web. Fonte: <https://cidadesinteligentes.agrocitylivinglab.com.br/comparar_cidades_resultado/>.

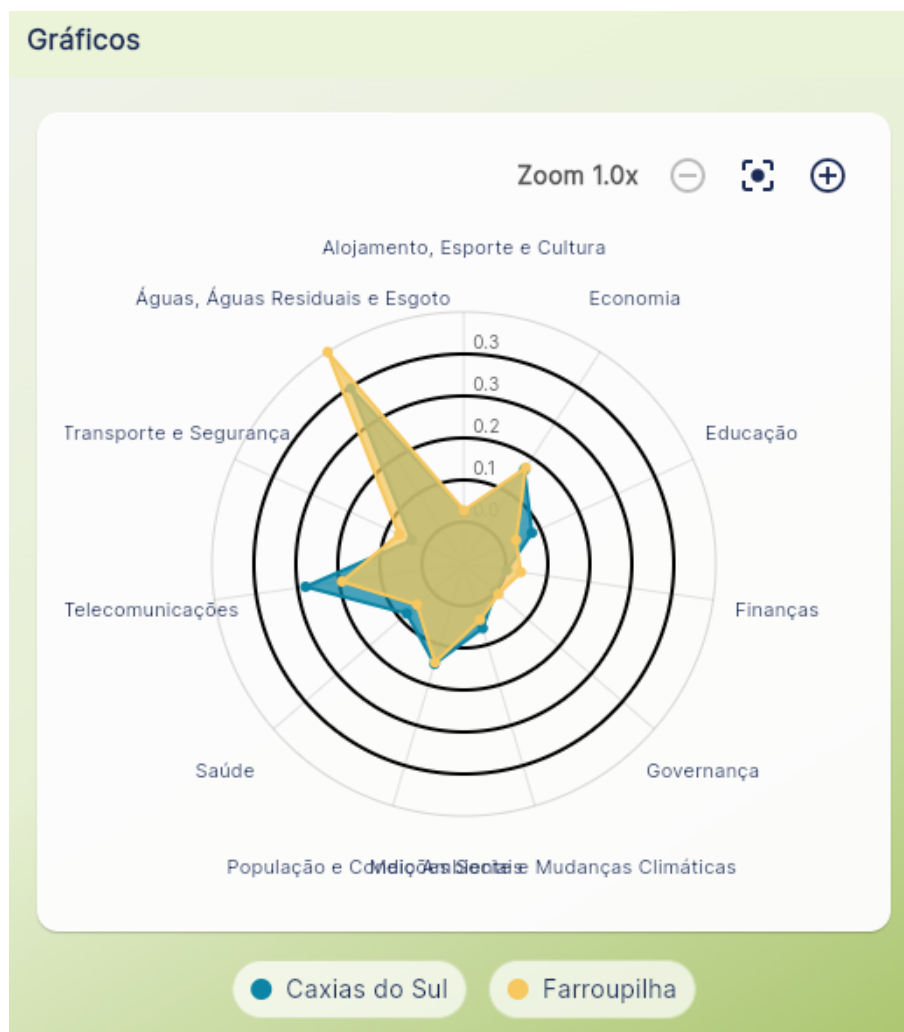


Figura 35 – Gráfico gerado pela aplicação mobile. Fonte: Geração própria.

Além da visualização gráfica, também foram comparados os dados detalhados por dimensão. As Figuras 36 e 37 apresentam os dados referentes à dimensão de telecomunicações nas aplicações web e mobile, respectivamente. A equivalência entre as informações exibidas demonstra que a aplicação mobile consegue reproduzir os dados essenciais da plataforma original, mantendo a estrutura necessária para a interpretação dos indicadores pelo usuário.

Cidade	Dimensão ID	Dimensão Nome	Média
Caxias do Sul	12	Telecomunicações	0.2431
Farroupilha	12	Telecomunicações	0.1697

Figura 36 – Dados de telecomunicações na aplicação web. Fonte: <https://cidadesinteligentes.agrocitylivinglab.com.br/comparar_cidades_resultado/>.

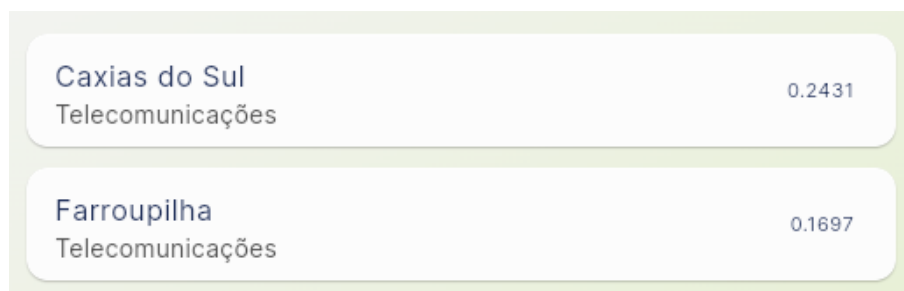


Figura 37 – Dados de telecomunicações na aplicação mobile. Fonte: Geração própria.

De forma semelhante, as Figuras 38 e 39 apresentam os dados relacionados à dimensão de economia. A comparação evidencia que a aplicação mobile preserva a coerência dos dados apresentados na versão web, ao mesmo tempo em que reorganiza a interface para favorecer a navegação em dispositivos móveis. Esse resultado reforça que a versão mobile atende ao propósito de disponibilizar as informações da plataforma de maneira acessível, mantendo a confiabilidade dos dados utilizados no processo de análise comparativa.

Cidade	Dimensão ID	Dimensão Nome	Média
Caxias do Sul	11	Economia	0.1474
Farroupilha	11	Economia	0.1521

Figura 38 – Dados de economia na aplicação web. Fonte: <https://cidadesinteligentes.agrocitylivinglab.com.br/comparar_cidades_resultado/>.

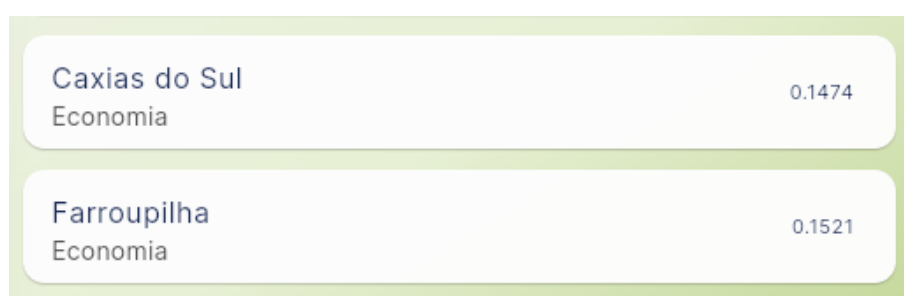


Figura 39 – Dados de economia na aplicação mobile. Fonte: Geração própria.

Assim, a comparação entre as duas aplicações indica que a solução *mobile* desenvolvida é capaz de apresentar os mesmos conjuntos de dados da aplicação web, adaptando-os a uma interface própria para o contexto *mobile*. Dessa forma, a aplicação contribui para ampliar o acesso às informações sobre cidades inteligentes, permitindo que a consulta e a análise dos indicadores sejam realizadas de maneira mais prática e adequada ao contexto de uso em dispositivos móveis.

5 CONSIDERAÇÕES FINAIS

A cidade é parte de um fenômeno moldado por relações humanas e por transformações no território. O modelo de cidades inteligentes foca no uso de tecnologias e na estruturação de dados públicos no espaço urbano, permitindo o monitoramento contínuo de indicadores e o planejamento integrado. O projeto Cidades do Conhecimento, do City Living Lab, atua diretamente na coleta, no tratamento e na visualização dessas métricas. A versão web do projeto agrupou as informações utilizando o Sistema de Capitais, os Objetivos de Desenvolvimento Sustentável (ODS) e o modelo de Cidades Inteligentes, substituindo planilhas isoladas por um sistema centralizado de consulta.

A versão anterior da plataforma não contemplava o uso por dispositivos móveis, nem oferecia uma visualização adaptada para esse formato. Com o objetivo de disponibilizar uma ferramenta aplicável em diferentes contextos e viabilizar a análise de indicadores urbanos, este projeto propôs o desenvolvimento de uma aplicação mobile. O desenvolvimento técnico da solução, bem como as decisões relacionadas à organização visual e ao tratamento dos dados, foi conduzido ao longo da execução do projeto, com base nos requisitos definidos e nas referências conceituais adotadas.

Nesse sentido, além de aprimorar a consulta às informações, a proposta busca fomentar a cultura de dados abertos. Com a popularização dos dispositivos móveis, espera-se que a plataforma facilite o entendimento e o monitoramento de indicadores essenciais para o desenvolvimento sustentável das cidades. Dessa forma, o caráter analítico da aplicação também poderá apoiar projetos, o delineamento de políticas baseadas em evidências e ações de planejamento urbano colaborativo.

A aplicação foi programada em Dart com o framework Flutter, o que permite o funcionamento em sistemas operacionais distintos com um código-base. A ferramenta não será offline pois dependerá de uma API para a comunicação. O aplicativo foi projetado exclusivamente para falar da base de dados externa em PostgreSQL. O sistema atua como consumidor de dados, executando requisições HTTP para obter as métricas processadas no servidor.

A solução desenvolvida possibilitou a consulta dos dados anteriormente disponíveis apenas na versão web, oferecendo uma alternativa adequada ao crescente uso de dispositivos móveis. A aplicação permite a visualização de indicadores, dimensões e modelos de forma organizada e intuitiva, preservando a estrutura das informações já consolidadas pelo projeto Cidades do Conhecimento e ampliando seu alcance para diferentes contextos de utilização. O fluxo de navegação foi estruturado para a execução das etapas de análise, permitindo a seleção dos modelos conceituais, a leitura das dimensões e a escolha de dois a cinco municípios para a realização de comparativos. A apresentação de resultados utiliza gráficos de radar gerados pela

biblioteca `fl_chart` e listas de expansão com as médias calculadas.

Dessa forma, o trabalho contribuiu para reduzir as limitações identificadas na plataforma existente, disponibilizando uma ferramenta que favorece a mobilidade, a praticidade e a disseminação dos dados produzidos pelo projeto. Ao retomar o problema de pesquisa proposto neste trabalho, que consistia em ampliar as possibilidades de visualização dos dados do projeto Cidades do Conhecimento em dispositivos móveis, mantendo a organização, a clareza e a eficiência na apresentação dos indicadores, conclui-se que o objetivo foi alcançado por meio do desenvolvimento da plataforma mobile. Assim, considera-se que a pesquisa respondeu satisfatoriamente à questão proposta, demonstrando que a adaptação da plataforma para o ambiente mobile constitui uma alternativa viável para a consulta dos indicadores do projeto.

5.1 ADEQUAÇÕES DA INTERFACE DA VERSÃO MOBILE EM RELAÇÃO À VERSÃO WEB

Embora a versão mobile preserve as funcionalidades presentes na plataforma web, foram necessárias diversas adequações na interface para atender às limitações de espaço e às características de interação dos dispositivos móveis. As alterações realizadas buscaram manter o fluxo de utilização da plataforma, ao mesmo tempo em que proporcionaram uma melhor experiência de navegação em telas reduzidas.

A primeira tela analisada corresponde à visualização das dimensões do modelo. Conforme apresentado na Figura 40, essa tela possui caráter informativo, apresentando ao usuário as dimensões e suas respectivas descrições. Na versão mobile, os componentes foram reorganizados para uma disposição vertical, feito por meio de *cards* enquanto na versão *web* é feito por meio de uma tabela, adequando-se às dimensões reduzidas da tela sem alterar o conteúdo disponibilizado pela versão web.

Dimensões do Modelo		
ID Dimensão	Dimensão	Descrição do Indicador
0	Geral	Dados sem dimensão específica, geralmente associados à população e/ou área do município
1	Identidade	Inclui os elementos formais e informais de capital que contribuíram e/ou contribuem para determinar a individualidade, clareza e diferenciação de uma cidade.
2	Inteligência	Capacidade dos sistemas de sentir, fazer sentido, e responder a agentes externos e eventos que são significativos para o bem-estar da cidade (ou seja, as agências de planejamento estratégico da cidade, futuros centros urbanos públicos e privados, estudos
3	Relacional	Capacidade de desenvolver interações de qualidade com todos os agentes significativos, tanto internos quanto externos. Qualidade da interação entre os agentes internos significativos da cidade, bem como entre a cidade e seus agentes externos significativo
4	Humano Individual	Capacidade de criar condições para o pleno desenvolvimento biológico e psicológico dos moradores. Capacidade de geração de valor de indivíduos.
5	Humano Coletivo	Capacidade de aumentar o potencial de realização do objetivo de suas comunidades constituintes. Capacidades coletivas e de equipe para gerar valor.

(a) Versão web.

Dimensões	
0 - Geral	Dados sem dimensão específica, geralmente associados à população e/ou área do município
1 - Identidade	Inclui os elementos formais e informais de capital que contribuíram e/ou contribuem para determinar a individualidade, clareza e diferenciação de uma cidade.
2 - Inteligência	Capacidade dos sistemas de sentir, fazer sentido, e responder a agentes externos e eventos que são significativos para o bem-estar da cidade (ou seja, as agências de planejamento estratégico da cidade, futuros centros urbanos públicos e privados, estudos
3 - Relacional	Capacidade de desenvolver interações de qualidade com todos os agentes significativos, tanto internos quanto externos. Qualidade da interação entre os agentes internos significativos da cidade, bem como entre a cidade e seus agentes externos significativo

(b) Versão mobile.

Figura 40 – Comparação da tela de dimensões entre as versões web e mobile. Fonte: Geração própria.

Outra funcionalidade disponibilizada pela plataforma é a tela de comparativos, apresentada na Figura 41. Na versão *web*, os filtros e informações são exibidos simultaneamente em uma única página. Para a versão *mobile*, optou-se pela utilização de abas e componentes do tipo *popover*, permitindo segmentar as informações e reduzir a quantidade de conteúdo visível ao mesmo tempo, tornando a navegação mais adequada às limitações de espaço dos dispositivos móveis.

A interface web para 'Comparativo de Cidades' apresenta um formulário com os seguintes elementos:

- Título:** Comparativo de Cidades
- Filtros:**
 - Seleção de Modelo: Sistema de Capitais
 - Seleção de Ano: 1989
- Busca:** Campo de texto com o valor 'São' e ícones de ajuda.
- Botões:** 'Adicionar' e 'Remover'.
- Lista de Resultados:**
 - São Miguel do Guaporé - RO
 - São Felipe D'Oeste - RO
 - São Francisco do Guaporé - RO
 - São Gabriel da Cachoeira - AM

(a) Versão web.

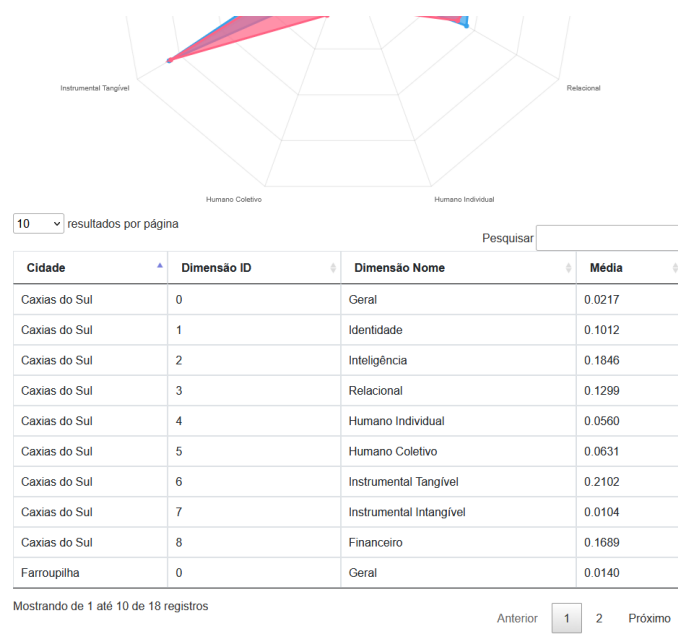
A interface mobile para 'Comparativo de Cidades' utiliza uma estrutura de abas e componentes do tipo *popover*. Os elementos visíveis incluem:

- Título:** Comparativo de Cidades
- Filtros:**
 - Modelo
 - Ano
- Busca:** Campo de texto com o valor 'São'.
- Lista de Resultados:**
 - São Bonifácio (SC - 4215901)
 - São Borja (RS - 4318002)
 - São Brás (AL - 2708204)
 - São Brás do Suaçuí (MG - 3160900)
 - São Braz do Piauí (PI - 2209559)

(b) Versão mobile.

Figura 41 – Comparação da tela de comparativos entre as versões web e mobile. Fonte: Geração própria.

Também houve ajustes na tela de exibição dos dados do comparativo, conforme ilustrado na Figura 42, a versão *web* utiliza uma estrutura tabular para exibição das informações logo abaixo do gráfico carregado. Entretanto, esse modelo se mostrou pouco adequado para dispositivos móveis devido à necessidade de rolagens horizontais e à redução da legibilidade. Por esse motivo, a versão *mobile* passou a utilizar um botão logo abaixo do gráfico que direciona o usuário a uma nova tela para exibição dos dados por meio de componentes do tipo *card*, nos quais as informações são apresentadas de forma individualizada, conforme ilustrado na Figura 43.



(a) Versão web.



(b) Versão mobile.

Figura 42 – Comparação da apresentação tabular dos indicadores entre as versões web e mobile.



Figura 43 – Tela de exibição das médias por dimensão da versão *mobile*. Fonte: Geração própria.

Além da apresentação tabular, a versão *web* desta tela permite a navegação entre as dimensões por meio de abas localizadas na parte superior da página. No entanto, essa abordagem não se mostrou adequada para a aplicação *mobile*, pois a quantidade de dimensões exigia muito deslocamento horizontal, dificultando a localização da dimensão selecionada e das demais opções disponíveis. Para resolver esse problema, a versão *mobile* passou a utilizar um menu do tipo *hamburger*, conforme apresentado na Figura 44, aproveitando melhor o espaço disponível em telas menores e facilitando a navegação entre as dimensões.



Figura 44 – Comparação da navegação pelas dimensões entre as versões web e mobile. Fonte: Geração própria.

5.2 SUGESTÕES DE TRABALHOS FUTUROS

Considerando que a base de dados em PostgreSQL depende de cargas manuais e não possui processos automatizados para a inserção de novos dados ou indicadores, as sugestões para a continuidade do projeto focam em funcionalidades de uso e em novas representações visuais, mantendo a premissa de que toda a consulta mobile seguirá sendo realizada exclusivamente via API:

- Exportação de resultados: Criação de uma funcionalidade para a geração de arquivos em formato PDF com os gráficos de radar e as tabelas de médias geradas, permitindo o arquivamento do comparativo no armazenamento do dispositivo móvel.
- Integração de novas visualizações gráficas: Implementação de mapas de calor (heatmaps) e gráficos de dispersão (scatter plots) na tela de resultados, oferecendo formas adicionais para o cruzamento de indicadores e o detalhamento das dimensões.
- Funcionalidade de favoritos: Inclusão de um recurso para o salvamento de parâmetros das comparações mais executadas, automatizando a seleção do modelo, do ano e dos municípios em consultas de rotina.

REFERÊNCIAS

- ALVES, M. A. *et al.* Smart cities no brasil e em portugal: o estado da arte. **Urbe. Revista Brasileira de Gestão Urbana**, v. 11, p. e20190061, 2019. Disponível em: <<https://doi.org/10.1590/2175-3369.011.e20190061>>.
- BACH, B. *et al.* Dashboard design patterns. **IEEE Transactions on Visualization and Computer Graphics**, IEEE, v. 29, n. 1, p. 366–376, 2023. Disponível em: <<https://doi.org/10.1109/TVCG.2022.3209448>>.
- BATTISTELO, R. **Aplicação web para indicadores de cidades do conhecimento**. Caxias do Sul: [s.n.], 2018. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) – Universidade de Caxias do Sul.
- BIKAKIS, N. Big data visualization tools. **arXiv preprint arXiv:1801.08336**, 2018. Disponível em: <<https://arxiv.org/abs/1801.08336>>.
- BREGALDA, V. **Coleta dos indicadores da plataforma de Cidades do Conhecimento**. Caxias do Sul: [s.n.], 2021. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) – Universidade de Caxias do Sul.
- BRUZZEGUEZ, G. A. *et al.* **Autoridade Nacional de Proteção de Dados: radar tecnológico – nº 1: cidades inteligentes**. Brasília, DF: ANPD, 2024.
- BUSE, R. F. Native or cross-platform mobile applications development. **Revista tinerilor economişti**, n. 40, p. 55–62, 2023.
- COVAS, M. D. M. C. D. M.; COVAS, A. M. A. Cidades inteligentes e criativas e smartificação dos territórios: apontamentos para reflexão. **Revista de Estudos Regionais**, n. 17, p. 45–68, 2021.
- DELJOUYI, A.; RAMSIN, R. Mdd4rest: Model-driven methodology for developing restful web services. In: **Proceedings of the 10th International Conference on Model-Driven Engineering and Software Development**. [S.l.]: SCITEPRESS, 2022. p. 156–163.
- FACHINELLI, A. C. *et al.* **Observatório de Cidades 2023**. Caxias do Sul: Corede Serra, 2023. Disponível em: <<https://doi.org/10.18226/9786500628241>>.
- GAHAR, R. M. *et al.* Open research issues and tools for visualization and big data analytics. **arXiv preprint arXiv:2404.12505**, 2024. Disponível em: <<https://arxiv.org/abs/2404.12505>>.
- GALWAY, L. P. *et al.* A scoping review examining governance, co-creation, and social and ecological justice in living labs literature. **Challenges**, MDPI, v. 13, n. 1, p. 1, 2022. Disponível em: <<https://doi.org/10.3390/challe13010001>>.
- GAO, Y. *et al.* Understanding the evolution of open government data research: towards open data sustainability and smartness. **International Review of Administrative Sciences**, v. 89, n. 1, p. 59–75, 2023.
- GUO, Z. *et al.* Type-directed program synthesis for restful apis. **Proceedings of the ACM on Programming Languages**, v. 6, n. PLDI, p. 652–675, 2022.

HABIBIPOUR, A.; STÅHLBRÖST, A. A living lab perspective on information systems development process. In: **International Conference on Information Systems Development**. Cham: Springer International Publishing, 2022.

HAMIDLI, N. Introduction to ui/ux design: key concepts and principles. **Preuzeto**, v. 28, p. 20–24, 2023.

JACOB, S. A. *et al.* Sharing compelling results: Data storytelling and data visualization. In: _____. **Coordinating Divisional and Departmental Student Affairs Assessment**. 1. ed. London: Routledge, 2023. p. 175–194.

JUNG, W. The effect of representational ui design quality of mobile shopping applications on users' intention to shop. **Procedia Computer Science**, Elsevier, v. 121, p. 887–894, 2017. Disponível em: <<https://doi.org/10.1016/j.procs.2017.11.023>>.

KAGINALKAR, A. *et al.* Smartairq: A big data governance framework for urban air quality management in smart cities. **Frontiers in Environmental Science**, Frontiers, v. 10, p. 785129, 2022. Disponível em: <<https://doi.org/10.3389/fenvs.2022.785129>>.

LEE, B. *et al.* (Ed.). **Mobile Data Visualization**. Boca Raton, FL, USA: CRC Press, 2021. ISBN 978-0367255742.

LIU, Z. Evaluating digitalized visualization interfaces: Integrating visual design elements and analytic hierarchy process. **International Journal of Human–Computer Interaction**, v. 41, n. 9, p. 5731–5760, 2025.

LOPES, S. **A Web Mobile: design responsivo e além para uma Web adaptada ao mundo mobile**. São Paulo: Casa do Código, 2014.

MACEDO, D. F.; LEMOS, D. L. D. S. Dados abertos governamentais: iniciativas e desafios na abertura de dados no brasil e outras esferas internacionais. **AtoZ: Novas Práticas em Informação e Conhecimento**, v. 10, n. 2, p. 14–26, 2021. Disponível em: <<https://doi.org/10.5380/atoz.v10i2.77737>>.

MARIA, G. P. D. S. S. **Readequação de plataforma web de indicadores para cidades inteligentes**. 2024. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) – Universidade de Caxias do Sul.

MARTÍNEZ, E. L.; CEBALLOS, C. S. Diseño web adaptativo o responsivo. **Revista Digital Universitaria**, v. 14, n. 6, p. 1–10, 2013.

MONTEIRO, G.; LUCAS, E. R. D. O. Dados científicos abertos: identificando o papel das políticas de gestão e das agências de fomento. **AtoZ: Novas Práticas em Informação e Conhecimento**, v. 8, n. 1, p. 5–14, jun. 2019. Disponível em: <<https://doi.org/10.5380/atoz.v8i1.67253>>.

MOUNIR, S. E. A. Software engineering for mobile applications: A survey on challenges and solutions. **arXiv preprint arXiv:2301.00602**, 2023. Disponível em: <<https://arxiv.org/abs/2301.00602>>.

MUHAMMAD, I. R. D.; PAPUTUNGAN, I. V. Development of backend server based on rest api architecture in e-wallet transfer system. **Jurnal Sains, Nalar, dan Aplikasi Teknologi Informatika**, v. 3, n. 2, p. 79–87, 2024. Disponível em: <<https://doi.org/10.20885/snati.v3.i2.35>>.

- NUNES, V. G. C.; QUEIROZ, P. G. G. **Avaliação de abordagens e definição de diretrizes para o desenvolvimento de aplicativos mobile: um relato de experiência utilizando a abordagem não nativa de desenvolvimento.** Pau dos Ferros: [s.n.], 2021. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) – Universidade Federal Rural do Semi-Árido. Disponível em: <<https://repositorio.ufersa.edu.br/handle/prefix/6917>>.
- NYABUTO, G. M. *et al.* Architectural review of client-server models. **International Journal of Scientific Research and Engineering Trends**, v. 10, n. 1, p. 142–149, 2024.
- PATIDAR, A. K.; SUMAN, U. A survey on mobile app development approaches with the industry perspective. **International Journal of Open Source Software and Processes**, IGI Global, v. 13, n. 1, p. 1–17, 2022.
- RODRIGUES, H. **Virtualização de ferramentas de carga de dados da plataforma Cidades do Conhecimento.** 2023. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) – Universidade de Caxias do Sul.
- SILVA, A. G. D. **Desenvolvimento da automatização da coleta de dados na plataforma de Cidades do Conhecimento.** 2022. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) – Universidade de Caxias do Sul.
- SILVA, F. C. C. D. Visualização de dados: passado, presente e futuro. **Liinc em Revista**, v. 15, n. 2, p. 374–389, 2019.
- SILVA, L. F. D. *et al.* Mobile user interaction design patterns: A systematic mapping study. **Information**, v. 13, n. 5, p. 216, 2022.
- SOEIRO, D. Smart cities and innovative governance systems: a reflection on urban living labs and action research. **Fennia – International Journal of Geography**, v. 199, n. 1, p. 114–126, 2021.
- SOLER, J. a. V. A. *et al.* A relevância da acessibilidade intuitiva: Como o foco na usabilidade no design de interfaces impacta o usuário. **Revista Interface Tecnológica**, v. 18, n. 2, p. 194–207, 2021.
- STEEN, K.; BUEREN, E. V. **Urban Living Labs: A Living Lab Way of Working.** [S.l.]: Amsterdam Institute for Advanced Metropolitan Solutions (AMS), 2017.
- SVALINA, A. *et al.* Assessing the design of interactive radial data visualizations for mobile devices. **Journal of Imaging**, MDPI, v. 9, n. 5, p. 93, 2023.
- SYSTÄ, K.; SAND, A. Experiments comparing cross-platform and native mobile applications. **International Journal of Software Engineering and Computer Systems**, v. 9, n. 1, p. 45–54, 2023.
- TERCANLI, H.; JONGBLOED, B. A systematic review of the literature on living labs in higher education institutions: Potentials and constraints. **Sustainability**, v. 14, n. 19, p. 12234, 2022. Disponível em: <<https://doi.org/10.3390/su141912234>>.
- ULTRAMARI, C. Conceito de cidade: dificuldades e razões para formulá-lo. **Revista Brasileira de Gestão e Desenvolvimento Regional**, v. 15, n. 6, p. 295–310, 2019.
- VASCONCELOS, P. D. A. As metamorfoses do conceito de cidade /// the metamorphoses in the concept of the city. **Revista Mercator**, v. 14, n. 4, p. 39–55, 2015.

WANG, A. *et al.* **The Hitchhiker's Guide to Successful Living Lab Operations**. 2022. ArXiv preprint arXiv:2212.00008. Disponível em: <<https://arxiv.org/abs/2212.00008>>.

WEICHBROTH, P. Usability issues with mobile applications: Insights from practitioners and future research directions. **Information and Software Technology**, Elsevier, v. 132, p. 106514, 2021.

WOOD, D. **Basics Interactive Design: Interface Design – An Introduction to Visual Communication in UI Design**. London: Bloomsbury Academic, 2014. ISBN 978-2940411993.

ZIMMERMANN, O. *et al.* **Patterns for API Design: Simplifying Integration with Loosely Coupled Message Exchanges**. Boston: Addison-Wesley Professional, 2022. ISBN 978-0136504290.