

UNIVERSIDADE ESTADUAL DO MARANHÃO

Samanta Kelley Sá

I-PATRI : A APLICABILIDADE DJANGO PARA CONTROLE PATRIMONIAL DE  
SECRETARIAS ESTADUAIS

São Luís - MA

2025

Samanta Kelley Sá

I-PATRI : A APLICABILIDADE DJANGO PARA CONTROLE PATRIMONIAL DE  
SECRETARIAS ESTADUAIS

Trabalho de Conclusão de Curso apresentado à Banca Examinadora do Curso de Engenharia de Computação, da Universidade Estadual do Maranhão, como pré-requisito para a obtenção de título de Bacharel em Engenharia de Computação

Orientador: Prof<sup>o</sup>. Esp. Edilson Carlos Silva  
Lima

São Luís - MA

2025

SAMANTA KELLEY SÁ

I-PATRI : A APLICABILIDADE DJANGO PARA CONTROLE PATRIMONIAL DE  
SECRETARIAS ESTADUAIS

Trabalho de Conclusão de Curso apresentado à Banca Examinadora do Curso de Engenharia de Computação, da Universidade Estadual do Maranhão, como pré-requisito para a obtenção de título de Bacharel em Engenharia de Computação

São Luís (MA), 31 de julho de 2025

BANCA EXAMINADORA

---

Prof<sup>a</sup>. Esp. Edilson Carlos Silva Lima (Orientador)  
Universidade Estadual do Maranhão (UEMA)

---

Prof. Me. Yonara Costa Magalhães  
Universidade Estadual do Maranhão (UEMA)

---

Prof. Dr. Jonathan Araújo Queiroz  
Universidade Estadual do Maranhão (UEMA)



Dedico este trabalho a Dra Jana Cristina e toda a discografia da Taylor Swift.

## AGRADECIMENTOS

Dedico esse espaço para agradecer por cada pessoa que me deu apoio, empurrões, dicas e paciência durante a construção do meu TCC. Principalmente àquelas que torceram de verdade para que eu me formasse — e entenderam que perguntar todo mês se eu já tinha terminado só aumentava o nível de ansiedade interno.

Primeiramente, à minha sonserina favorita, Lícia Oliveira. Não tenho palavras suficientes para agradecer todo o apoio, as sugestões nas escritas, os palpites certos nos trechos mais complicados e burocráticos, e principalmente, a princesa Morgana, que como é filha sua eu tenho que agradecer a você, por todas as vezes que me permitiu recarregar as energias visitando você para segurar ela para lhe ajudar.

Agradeço também ao meu ambiente de trabalho, que foi essencial para o objeto de estudo deste projeto. Ter contato direto com os sistemas, fluxos e realidades da rotina pública me deu uma visão prática que nenhum PDF sozinho traria.

Aos professores que cruzaram meu caminho, tanto os presenciais quanto os das telas — incluindo os cursos online e os do Senac —, muito obrigado. Cada aula foi mais uma peça que me ajudou a confirmar que minhas escolhas de linguagem e banco de dados não foram aleatórias, mas estratégias conscientes para o meu futuro.

Não posso deixar de mencionar cada membro da minha família e cada amigo que, mesmo sem entender direito o que eu estava fazendo, mandou boas energias, orações e apoio sincero. Vocês fizeram toda diferença, eu realmente senti as boas vibrações.

E, por fim (mas com carinho especial), meu agradecimento vai para todos os fones de ouvido que sobreviveram até aqui — e às playlists que embalaram meu foco, me salvaram de crises, e me acompanharam nas madrugadas silenciosas de escrita e muito pop.

Obrigada a todos.



"Só sei que nada sei." (Sócrates)

## RESUMO

O presente trabalho tem como principal foco a otimização do controle do patrimônio em secretarias estaduais através do desenvolvimento de um sistema informatizado, visando a melhoria da gestão de bens públicos. O objetivo geral é analisar e desenvolver uma aplicabilidade Python com Django para o controle do patrimônio de secretarias estaduais, almejando melhorar e promover uma gestão mais eficiente e transparente dos principais ativos públicos. Para tanto, o estudo contextualiza o framework Django, sua arquitetura Model-View-Template (MVT), conceitua a gestão patrimonial e detalha a implementação de um sistema funcional e acessível, o i-Patri. Abordar a gestão patrimonial por meio de soluções tecnológicas é relevante, pois garante a transparência, rastreabilidade e eficiência necessárias para a administração pública em um cenário de constante modernização. O presente estudo consiste em pesquisa aplicada de caráter exploratório, sob a forma de um estudo de caso, com resultados tratados de maneira qualitativa, obtidos a partir de observação e entrevistas, além da coleta de informações em fontes secundárias. A metodologia ágil utilizada seguiu os princípios da *Lean*, com foco na entrega incremental e no feedback contínuo. A partir da condução do processo de pesquisa e desenvolvimento, foi possível concluir que a aplicação i-Patri, desenvolvida com Django, proporciona maior controle e rastreabilidade dos bens públicos, facilita auditorias, melhora a transparência administrativa e se destaca por oferecer maior eficiência, melhor escalabilidade e usabilidade aprimorada em comparação com sistemas anteriores. A implementação de um banco de dados relacional (MySQL) garantiu a segurança e a integridade das informações, reforçando a importância da adoção de soluções tecnológicas para a modernização e eficiência da gestão pública.

**Palavras-chave:** Django Framework; Gestão Patrimonial; Lean.

## **ABSTRACT**

This paper focuses on optimizing asset control in state secretariats through the development of a computerized system, aiming to improve the management of public assets. The main objective is to analyze and develop a Python application using Django for managing the patrimony of state secretariats, seeking to enhance and promote more efficient and transparent management of key public assets. To this end, the study contextualizes the Django framework, its Model-View-Template (MVT) architecture, conceptualizes asset management, and details the implementation of a functional and accessible system named i-Patri. Addressing asset management through technological solutions is relevant as it ensures the transparency, traceability, and efficiency required for public administration in a constantly modernizing scenario. This work is an applied, exploratory study, conducted in the form of a case study with results treated qualitatively, based on observation and interviews, in addition to data collected from secondary sources. The adopted approach followed the principles of Lean methodologies, focusing on incremental delivery and continuous feedback. As a result of the research and development process, it was concluded that the i-Patri application, developed using Django, provides optimized control and traceability of public assets, facilitates audits, improves administrative transparency, and stands out for offering greater efficiency, better scalability, and enhanced usability compared to previous systems. The implementation of a relational database (MySQL) ensured information security and integrity, reinforcing the importance of adopting technological solutions for the modernization and efficiency of public management.

**Keywords:** Django Framework; Asset Management; Lean.

## LISTA DE ILUSTRAÇÕES

Figura 1 — Ciclo PDCA da Metodologia Lean	21
Figura 2 — Algumas figuras usadas do diagrama	24
Figura 3 — Exemplo de Diagrama de Atividade	26
Figura 4 — Relacionamento MVT	35
Figura 5 — Ciclo Lean Aplicado	52
Figura 6 — Diagrama Caso de Uso Aplicado	53
Figura 7 — Factory Aplicado	54
Figura 8 — Parte da árvore de arquivos da aplicação	55
Figura 9 — POO aplicado no Django com Abstract e Herança	57
Figura 10 — Diagrama de Classes Aplicado	58
Figura 11 — Diagrama De Atividade	60
Figura 12 — Diagrama de classe focado em Orgao e Setor	61
Figura 13 — Tabelas para auditoria do banco	63
Figura 14 — Gráfico do site mostrando gastos por tipo	64
Figura 15 — Página de Visitas do iPatri	67

## LISTA DE ABREVIATURAS E SIGLAS

API	Application Programming Interface
CAPEC	Casa de Apoio às Pessoas com Câncer
MER	Modelo Entidade Relacionamento
MVC	Modelo Visão Controle
MVT	Modelo Visualização Template
NoSQL	Not Only Structured Query Language
ONG	Organização Não Governamental
ORM	Object-Relational Mapping
PCASP	Plano de Contas Aplicadas ao Setor Público
PDCA	Plan Do Check Act
POO	Programação Orientada a Objetos
SGBD	Sistema Gerenciador de Banco de Dados
SMART	Specific Measurable Achievable Relevant and Time-bound
SQL	Structured Query Language
UML	Unified Modeling Language
URL	Uniform Resource Locator

## SUMÁRIO

1	<b>INTRODUÇÃO</b>	12
1.1	OBJETIVOS	13
1.1.1	<b>Objetivo Geral</b>	13
1.1.2	<b>Objetivos Específicos</b>	13
1.2	TRABALHOS RELACIONADOS	13
1.3	METODOLOGIA CIENTÍFICA	15
1.3.1	<b>Caracterização da Pesquisa</b>	15
1.4	ESTRUTURA DO TRABALHO	17
2	<b>FUNDAMENTAÇÃO TEÓRICA</b>	19
2.1	METODOLOGIA ÁGIL	19
2.1.1	<b>Metodologia Lean</b>	20
2.2	UML	23
2.2.1	<b>Diagrama de Caso de Uso</b>	23
2.2.2	<b>Diagrama de Classe</b>	25
2.2.3	<b>Diagrama de Atividade</b>	25
2.3	PADRÕES DE PROJETO	28
2.3.1	<b>Considerações na Escolha do Padrão</b>	29
2.3.2	<b>Mapeamento para Bancos de Dados Relacionais</b>	30
2.3.3	<b>Integração e Iteração</b>	30
2.4	LINGUAGEM DE PROGRAMAÇÃO PYTHON	31
2.4.1	<b>Framework Django</b>	32
2.4.2	<b>Padrão MVT</b>	34
2.5	POO	36
2.5.1	<b>Abstração</b>	36
2.5.2	<b>Herança</b>	37
2.5.3	<b>Polimorfismo</b>	38
2.5.4	<b>Encapsulamento</b>	38
2.6	BANCO DE DADOS	40
2.6.1	<b>MySQL</b>	42
2.7	ADMINISTRAÇÃO PÚBLICA ESTADUAL PATRIMONIAL	43
2.8	CONCEITOS DE PATRIMÔNIO PÚBLICO	45
2.8.1	<b>Gestão Patrimonial</b>	47
2.8.2	<b>Gestão de Estoque</b>	47
3	<b>DESENVOLVIMENTO</b>	49
3.1	CENÁRIO ENCONTRADO	49
3.1.1	<b>Coleta e Análise de Dados</b>	51
3.1.2	<b>Relatório dos Resultados</b>	52
3.2	ARQUITETURA DA SOLUÇÃO	53
3.3	BANCO DE DADOS NO IPATRI	61
4	<b>RESULTADOS E DISCUSSÕES</b>	64
4.1	COMPARATIVO GERAL	65

5	<b>CONCLUSÃO</b>	69
	<b>REFERÊNCIAS</b>	71
	<b>GLOSSÁRIO</b>	74
	APÊNDICE A — Subtítulo do apêndice	75
	ANEXO A — Subtítulo do anexo	76

## 1 INTRODUÇÃO

Quem trabalha em uma secretaria do Governo do Estado, no setor responsável pelo controle patrimonial, enfrenta desafios significativos ao não possuir um controle adequado dos bens da secretaria, especialmente diante das transformações que exigem uma gestão mais eficiente. Sem um sistema eficaz, há riscos de perda, mau uso ou desperdício de recursos públicos, dificultando a transparência e a prestação de contas. Portanto, é fundamental implementar mecanismos modernos e estruturados de controle patrimonial para garantir o acompanhamento, a preservação e a gestão correta dos bens públicos, assegurando a eficiência administrativa e o cumprimento das normas legais.

Nesse contexto de transformações estruturais e desafios complexos, torna-se fundamental a capacitação contínua dos profissionais que atuam na área patrimonial e no setor público. Paralelamente, são cruciais o desenvolvimento e a atualização de sistemas e projetos *web* voltados para uma gestão mais transparente, exigindo que os servidores estejam preparados para integrar e utilizar novas tecnologias de maneira eficaz. A utilização do *framework*<sup>1</sup> Django, por exemplo, representa uma solução moderna e eficiente que não só atende às crescentes demandas das secretarias estaduais, mas também fortalece a transparência e a eficácia da administração pública, promovendo uma gestão patrimonial mais produtiva e responsiva às necessidades da sociedade.

Discutir como as secretarias lidam com seu controle patrimonial é relevante, pois envolve a participação da alta e média gestão, permitindo uma visão atualizada e em tempo real das áreas que necessitam de maior atenção. Isso capacita os executivos a tomarem decisões mais informadas. A utilização de aplicações *web* para o controle patrimonial das secretarias tem o potencial de melhorar a colaboração entre as equipes responsáveis pela gestão dos bens públicos, oferecendo escalabilidade robusta, seja para pequenas ou grandes quantidades de informações. Assim, a implementação de um *software* adequado atenderá a essas necessidades, conforme será detalhado nas seções seguintes.

---

<sup>1</sup>Framework, em desenvolvimento de software, é uma abstração que une códigos comuns entre vários projetos de software provendo uma funcionalidade genérica. Um framework pode atingir uma funcionalidade específica, por configuração, durante a programação de uma aplicação.

## 1.1 OBJETIVOS

A seguir é apresentado o objetivo geral deste trabalho e também os objetivos específicos.

### 1.1.1 Objetivo Geral

Analisar e Desenvolver uma aplicabilidade *Python* com *Django* para o controle do patrimônio de secretarias estaduais, almejando melhorar e promover uma gestão mais eficiente e transparente dos principais ativos públicos.

### 1.1.2 Objetivos Específicos

- Contextualizar a linguagem *Python*, com ênfase em sua associação ao *framework Django*;
- Conceituar o controle patrimonial no contexto da gestão pública;
- Implementar uma aplicação intuitiva e eficiente que atenda às necessidades dos usuários responsáveis pelo controle patrimonial em secretarias estaduais;
- Apresentar os impactos e benefícios da utilização do Python com Django na otimização do controle patrimonial de uma secretaria estadual.

## 1.2 TRABALHOS RELACIONADOS

Nessa seção serão apresentados alguns trabalhos científicos encontrados que englobam os conceitos de bens, patrimônio público e de aplicações web voltadas ao auxílio do mesmo.

Em Scott (2021), é enfatizada a importância da análise dos bens móveis patrimoniais na administração pública. Para a autora, essa análise é crucial para garantir a gestão adequada do patrimônio público, que deve estar em conformidade com leis e regulamentos. Scott (2021) argumenta que a identificação, reconhecimento, mensuração e evidenciação dos bens móveis são fundamentais para o controle patrimonial eficaz. Através de entrevistas e análise de documentos, ela conclui que a documentação existente contribui significativamente para a gestão e controle desses bens, além de fornecer sugestões para melhorar a aplicação dos aspectos

patrimoniais. O trabalho destaca que a gestão patrimonial não é apenas uma obrigação legal, mas também uma prática necessária para assegurar a transparência e a eficiência no uso dos recursos públicos, o que, por sua vez, reforça a responsabilidade da administração pública perante a sociedade.

Vico (2021) destaca a importância de se ter um sistema de controle de estoque eficiente para a CAPEC, uma ONG que apoia pessoas carentes com câncer. O autor enfatiza que a implementação de uma solução personalizada, utilizando a plataforma *Salesforce*, é crucial para otimizar a gestão de recursos da ONG, que atualmente enfrenta desafios significativos devido ao uso de um sistema obsoleto que carece de visibilidade das informações essenciais. Com a nova ferramenta, espera-se aumentar a produtividade dos responsáveis pelo estoque, garantindo que as informações sobre validade e quantidade de produtos estejam acessíveis a todos os gestores. Isso não só melhorará a eficiência operacional da ONG, mas também permitirá que mais famílias recebam assistência de forma mais organizada e eficaz.

Por sua vez, em Teixeira (2023), o autor apresenta a criação de um protótipo de aplicação web, que visou resolver problemas específicos de controle de estoque patrimonial, proporcionando um sistema funcional que atende às necessidades dos usuários, resultando em maior precisão nos registros, redução de erros e aumento da satisfação. Teixeira (2023) defende que uma gestão eficiente do inventário, inclui registros precisos de entradas e saídas, rastreamento de produtos e monitoramento das responsabilidades dos colaboradores, pode ser significativamente aprimorada com o uso de Tecnologias da Informação e Comunicação, e quando essas tecnologias são aplicadas corretamente, elas não apenas aumentam a produtividade dos usuários, mas também minimizam erros e insatisfações, que são comuns na ausência de sistemas estruturados.

Após conhecer alguns exemplos na área, podemos pontuar que esta pesquisa se diferencia por sua busca em compreender as experiências e percepções dos participantes em um contexto específico, onde, junto à união de diversas metodologias posteriormente mencionadas, promove uma melhor eficiência e melhoria contínua no desenvolvimento de *software* para controle patrimonial em ambientes governamentais. O estudo destaca a importância da indexação eficiente de dados e da implementação de ciclos curtos de *feedbacks* para atender às expectativas dos usuários, oferecendo uma aplicação prática e atualizada das teorias em um contexto contemporâneo.

### 1.3 METODOLOGIA CIENTÍFICA

A produção científica ocorre por meio de um processo estruturado, que envolve a formulação de hipóteses, a coleta de dados e a análise crítica das informações obtidas. As abordagens requerem uma compreensão empática levando em consideração as estruturas sociais que moldam as ações e significados atribuídos pelos indivíduos (GIL, 2008).

Dito isto, o presente estudo consiste em pesquisa aplicada de caráter exploratório, configurando um estudo de caso, que visa analisar e desenvolver uma aplicabilidade Python com Django para o controle do patrimônio de secretarias estaduais, almejando melhorar e promover uma gestão mais eficiente e transparente dos principais ativos públicos.

Nesse sentido, os resultados serão apresentados de forma qualitativa, a partir da coleta de informações de fontes primárias e secundárias, incluindo leis, decretos, manuais, contratos, sistemas informatizados, e dados obtidos por meio de entrevistas e observação direta dos processos.

#### 1.3.1 Caracterização da Pesquisa

Tendo como objetivo apresentar impactos de uma aplicação web na Gestão Patrimonial em uma instituição governamental estadual de forma a analisar e implementar um novo método de usabilidade, esta pesquisa tem natureza aplicada, pois tem referencial teórico com sua base na análise da realidade estudada, além de ter como motivação básica a solução de problemas concretos, práticos e operacionais.

A abordagem deste trabalho caracteriza-se como qualitativa, porque o envolvimento na pesquisa será levado à reflexão sobre ações e as consequências dos procedimentos que poderão ser modificadoras para a realidade na qual está inserido.

Sampieri, Collado e Lucio (2013), afirma que a abordagem qualitativa utiliza o ambiente como fonte direta para a obtenção dos dados:

O enfoque qualitativo é selecionado quando buscamos compreender a perspectiva dos participantes (indivíduos ou grupos pequenos de pessoas que serão pesquisados) sobre os fenômenos que os rodeiam, aprofundar em suas experiências, pontos de vista, opiniões e significados, isto é, a forma como os participantes percebem subjetivamente sua realidade.

Os objetos de estudo foram leis, decretos, manuais, contratos, sistemas informatizados, entrevistas e outros envolvidos na pesquisa, que serviram como base para a elaboração do projeto de pesquisa. Foi adotado a configuração exploratória, porque envolve levantamento documental, entrevistas não padronizadas e estudos de caso, na qual Gil (2008), afirma que sua finalidade principal é desenvolver, esclarecer e modificar conceitos e ideias, acerca de determinado fato.

Os níveis de manifestação do estudo são identificados a partir dos objetivos específicos propostos. Por isso seguindo os preceitos de Gil (2008) fora feito a divisão dos níveis de pesquisa em: estudos exploratórios, descritivos e explicativos. As pesquisas desenvolvidas foram feitas por meio de consulta bibliográfica, através de diagnóstico baseado na literatura, por relatos de outros pesquisadores na área, e mediante coleta de informações, por condução de anotações e estudo profundo da aplicação web, que são caracterizados como pesquisa exploratória, que neste caso, casou-se sincronicamente com este estudo.

Sampieri, Collado e Lucio (2013) relata que para obtermos os dados necessários para a elaboração da pesquisa, se faz necessário traçar um plano.

Seis elementos são fundamentais para formular um problema qualitativo: objetivos de pesquisa, perguntas de pesquisa, justificativa da pesquisa, sua viabilidade, avaliação das deficiências no conhecimento do problema e definição inicial do ambiente ou contexto.

Para fins do delineamento da pesquisa, definiu-se caracterizá-la também como bibliográfica e documental por ser desenvolvida a partir de documentos disponíveis no órgão, constituídos por ordens de serviço, acesso privativo ao sistema, termo de referência, contratos, organogramas e outros que possam fornecer embasamento para a pesquisa.

Para a pesquisa valerão das chamadas fontes de papel, usadas para a pesquisa bibliográfica e para a pesquisa documental, cujo esta última é semelhante à pesquisa bibliográfica, pois segundo Gil (2008) tem como única diferença entre elas, a natureza das fontes. Enquanto a pesquisa bibliográfica é fundamentada nas contribuições dos diversos autores sobre determinados assuntos, a pesquisa documental utiliza de fontes de materiais que ainda não receberam qualquer tratamento analítico, ou ainda podem ser reelaborados segundo a finalidade da pesquisa.

A planificação da pesquisa inclui, em primeiro lugar, o levantamento dos dados secundários, para posterior contato com as fontes primárias, a fim de promover a coleta

de dados em campo. Serão aplicados os seguintes instrumentos de pesquisa: observação e entrevistas. Os instrumentos de pesquisa serão aplicados de maneira planejada, com entrevistas semiestruturadas conduzidas com servidores públicos e gestores diretamente envolvidos com o controle patrimonial nas secretarias estaduais, focando na identificação de desafios, fluxos de trabalho e requisitos para aprimoramento da gestão. A observação será realizada nos ambientes de trabalho para compreender o uso atual de sistemas e a rotina de controle de ativos.

Para o projeto deste estudo, foi utilizada a metodologia *Lean*, que visa a eliminação de desperdícios e a maximização do valor entregue ao cliente no desenvolvimento de software. Entre suas vantagens, destaca-se a eficiência nos processos, a redução de custos e a melhoria contínua, possibilitando entregas ágeis e com qualidade superior. A metodologia também incentiva a colaboração e a simplificação de atividades, garantindo maior agilidade na adaptação às mudanças. Os detalhes e impactos dessa abordagem serão discutidos em maior profundidade nos próximos capítulos.

#### 1.4 ESTRUTURA DO TRABALHO

Este Trabalho investiga a aplicabilidade de uma ferramenta web voltada ao auxílio de secretarias estaduais no controle, organização e visualização dos bens patrimoniais.

No Capítulo 2, são discutidos os conceitos fundamentais de Administração Pública e Patrimônio Público, fornecendo a base teórica essencial para o desenvolvimento do estudo. Além disso, são explorados os princípios da Metodologia Ágil, da Programação Orientada a Objetos e da Linguagem de Modelagem Unificada. O capítulo ainda examina o padrão de arquitetura de software MVC, os principais padrões de projeto, o *framework* Django e o banco de dados MySQL, que formam o alicerce técnico desta pesquisa.

No Capítulo 3, é apresentada a metodologia adotada, incluindo o cenário inicial, as arquiteturas escolhidas para solucionar os desafios propostos e o banco de dados utilizado para a organização das métricas.

O Capítulo 4 expõe os resultados e discussões sobre a aplicação da ferramenta em dois cenários distintos, comparando-os com as análises estratégicas anuais e os

levantamentos setoriais realizados. Por fim, o Capítulo 5 apresenta as considerações finais do estudo, além de propostas para trabalhos futuros.

## 2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo, o objetivo é apresentar e discutir os conceitos e teorias mais relevantes sobre o tema, fornecendo uma base teórica sólida para apoiar as análises do estudo conceituando os aspectos essenciais do tema para garantir uma abordagem coerente e fundamentada ao longo do trabalho.

### 2.1 METODOLOGIA ÁGIL

A metodologia ágil surgiu como uma alternativa ao modelo tradicional em cascata, que seguia uma abordagem sequencial e rígida para o desenvolvimento de software. No modelo em cascata, o tempo era um grande empecilho, pois cada etapa precisava ser concluída antes de seguir para a próxima, além de ser inadequado para mudanças de requisitos ao longo do projeto. Em resposta a essas limitações, surgiu o Manifesto Ágil, que introduziu uma abordagem mais flexível e colaborativa, enfatizando a colaboração contínua, a entrega incremental de software e a flexibilidade para mudanças ao longo do desenvolvimento.

Segundo Kulshrestha (2019), o desenvolvimento ágil envolve equipes multifuncionais e autogeridas, que trabalham de forma colaborativa com os clientes para adaptar os requisitos e soluções conforme o projeto evolui. O foco está em entregas incrementais e rápidas, com o software funcional sendo entregue regularmente, permitindo ajustes com base no *feedback* contínuo. A filosofia do ágil é centrada em quatro valores fundamentais: interações humanas acima de processos, software funcional acima de documentação, colaboração com o cliente acima de negociações contratuais, e adaptação a mudanças acima de seguir um plano rígido.

Os projetos ágeis são divididos em pequenos ciclos chamados *sprints*, geralmente com duração de duas a quatro semanas. Cada *sprint* entrega uma parte funcional do produto final, permitindo que o time colete *feedback* contínuo e faça ajustes conforme necessário, aumentando a capacidade de adaptação do projeto às necessidades em evolução do cliente. Entre os benefícios da metodologia ágil, destaca-se a entrega contínua de funcionalidades em ciclos curtos, permitindo que o cliente veja o progresso do projeto de forma constante e ofereça *feedback*, mesmo nas fases avançadas de desenvolvimento.

A comunicação diária entre desenvolvedores e *stakeholders* garante que o produto atenda às expectativas e que mudanças sejam rapidamente incorporadas. A metodologia ágil promove uma comunicação constante entre as equipes de desenvolvimento e as partes interessadas, melhorando a transparência e a eficiência. A implementação da metodologia ágil pode ser feita por meio de *frameworks* como *Scrum*, *Kanban* e *Extreme Programming*, que utilizam os princípios ágeis para ajudar as equipes a gerenciarem e organizar seu trabalho de maneira mais eficaz. Todos compartilham a essência de desenvolver software de forma iterativa, rápida e adaptável, garantindo o foco na entrega de valor ao cliente. (SILVA; GRACIANO, 2022)

### 2.1.1 Metodologia *Lean*

Segundo Gomes (2010), a metodologia *Lean*, adaptada para o desenvolvimento de software, oferece uma abordagem focada em otimizar processos e eliminar desperdícios, resultando em maior eficiência e agilidade no desenvolvimento de sistemas. Inspirada no *Lean Manufacturing*, essa metodologia visa maximizar o valor entregue ao cliente com o menor esforço possível.

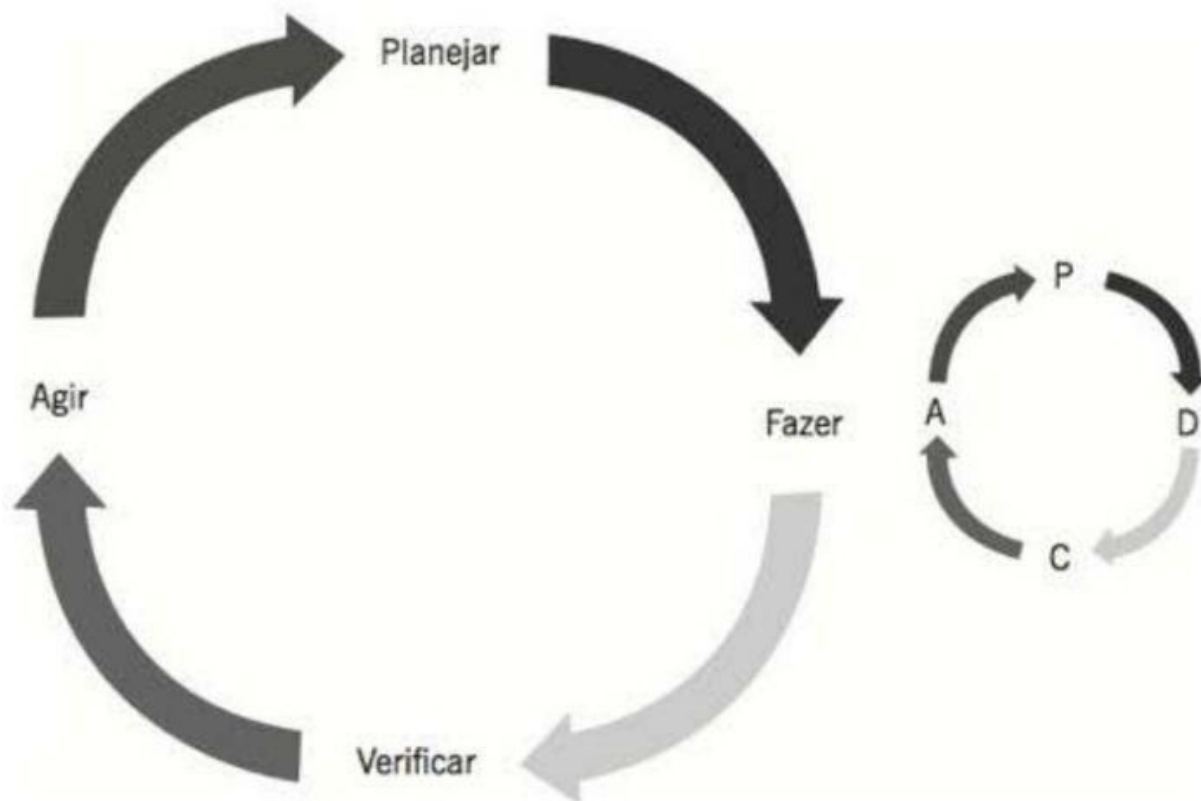
Levando em consideração os conceitos já apresentados sobre o assunto, pode-se também entender que, segundo Dennis (2008), a aplicação dos princípios *Lean* ao desenvolvimento de software complementa as práticas ágeis, focando na criação de valor e eliminação de desperdícios, semelhante ao Sistema Toyota de Produção na manufatura. Esse enfoque permite que as equipes entreguem software de forma incremental, obtendo *feedback* rápido e implementando melhorias contínuas.

Vale ressaltar que, entre as principais vantagens do *Lean* no desenvolvimento de software está a melhoria contínua, como mostrado na figura 1, que promove ajustes frequentes ao longo do ciclo de desenvolvimento. Para Dennis (2008), esse enfoque permite que equipes identifiquem problemas rapidamente e ajustem processos, garantindo que o projeto atenda às necessidades dos clientes de maneira eficiente.

Nesse contexto, o ciclo de desenvolvimento de produto *Lean*, fundamentado no método PDCA (Planejar, Fazer, Verificar e Agir), como mostrado na figura 1, representa uma abordagem sistemática essencial para otimizar processos, eliminar desperdícios e, principalmente, maximizar o valor entregue ao cliente. Amplamente utilizado em ambientes de desenvolvimento ágil, como na fabricação e

desenvolvimento de software, o modelo PDCA oferece uma estrutura cíclica para a melhoria contínua.

Figura 1 — Ciclo PDCA da Metodologia Lean



Fonte: Dennis (2008).

A fase de **Planejar** concentra-se na compreensão profunda da situação atual e na definição clara de objetivos. Inicialmente, é crucial entender o contexto, identificando os problemas existentes e sua relação com os objetivos organizacionais. Em seguida, estabelecem-se metas *SMART* (específicas, mensuráveis, alcançáveis, relevantes e temporais) que direcionam o planejamento estratégico. Este, por sua vez, define o caminho a ser percorrido pela equipe, utilizando ferramentas como *nemawashi* (busca por consenso prévio) e *catchball* (troca de ideias) para facilitar a comunicação e o alinhamento. A criação de um plano abrangente, detalhando os "Cinco Ws e um H" (quem, o quê, quando, onde, por quê e como), garante uma abordagem completa. Adicionalmente, a identificação de objetivos de ruptura, aqueles que podem gerar melhorias significativas no processo, é crucial para impulsionar a inovação.

Na etapa de **Fazer**, o plano é colocado em prática. A execução do plano deve seguir rigorosamente o que foi estabelecido, com a realização de um piloto para validar

a abordagem antes da implementação completa, minimizando riscos e retrabalhos. A avaliação da competência da equipe e o preenchimento de lacunas, garantindo que todos estejam aptos para executar as tarefas, são essenciais para o sucesso. A documentação clara dos processos permite uma análise futura mais precisa e facilita a identificação de oportunidades de melhoria.

A fase de **Verificar** é dedicada à análise dos resultados. A medição dos resultados, utilizando métricas e sistemas de medição, permite comparar o desempenho real com o que foi planejado, identificando possíveis desvios. A análise das discrepâncias entre o que aconteceu e o que deveria ter acontecido, buscando entender as causas dos problemas, é fundamental para a correção de rumo. A implementação de um sistema de feedback contínuo possibilita ajustes imediatos e garante que o processo se mantenha alinhado com os objetivos.

Por fim, a etapa de **Agir** concentra-se na implementação de melhorias com base nas análises realizadas. Os ajustes necessários no processo são feitos com base nos *feedbacks* e nos resultados da verificação. O ciclo de aprendizagem é essencial, promovendo um ambiente onde as lições aprendidas são documentadas e compartilhadas com a equipe, fomentando o aprimoramento contínuo. A revisão do plano de ação e dos objetivos, com base nas lições aprendidas e nas novas condições do mercado ou do projeto, garante que o ciclo PDCA seja dinâmico e responsivo às mudanças.

Em conclusão, o ciclo PDCA no desenvolvimento de produtos Lean transcende a mera aplicação de técnicas, representando uma filosofia que fortalece a cultura organizacional e aprimora a eficiência e a eficácia dos processos. A prática contínua desses ciclos promove um ambiente de trabalho colaborativo e focado na entrega de valor para o cliente, impulsionando a organização em direção à excelência.

## 2.2 UML

A UML, também conhecida por *Unified Modeling Language*, é uma linguagem padrão amplamente utilizada no desenvolvimento de software para a visualização, especificação, construção e documentação de sistemas complexos. Ela oferece um conjunto de ferramentas e diagramas que facilitam a comunicação entre desenvolvedores, projetistas e outros *stakeholders*, garantindo que todos compartilhem

uma visão consistente do projeto, aumentando a eficiência na coordenação do desenvolvimento. (PENDER, 2003)

Entre seus principais benefícios, destaca-se a capacidade de representar sistemas orientados a objetos de maneira clara e estruturada, permitindo a modelagem tanto de software quanto de processos empresariais. A UML é eficaz na decomposição do sistema em componentes fundamentais, como classes, objetos e interações, fornecendo uma representação gráfica precisa das relações e comportamentos. Ela facilita a análise dos requisitos e a definição de arquiteturas complexas, simplificando a compreensão do sistema em desenvolvimento.

Segundo Freitas (2008), a flexibilidade da UML também se reflete em sua ampla aplicabilidade. Por ser uma linguagem padrão, ela oferece compatibilidade com diferentes ferramentas e plataformas, promovendo uma padronização nos projetos de software. Além disso, desempenha um papel importante ao longo do ciclo de vida do desenvolvimento, desde a fase inicial de levantamento de requisitos até a implementação e manutenção, garantindo consistência na documentação.

Com o apoio da UML, a comunicação entre as equipes é otimizada, pois seus diagramas ajudam a estabelecer uma linguagem comum, evitando ambiguidade na interpretação de requisitos e *design*. A modularidade promovida pela linguagem de modularidade unificada permite que sistemas complexos sejam mais bem compreendidos e gerenciados, proporcionando maior controle sobre mudanças e adaptações ao longo do tempo. (GUEDES, 2018)

### 2.2.1 Diagrama de Caso de Uso

Segundo Ribeiro (2021), os diagramas de caso de uso são uma parte fundamental da UML, servindo como uma ferramenta eficaz para capturar e representar as interações entre os atores e o sistema. Eles ilustram de forma clara como os usuários ou sistemas externos interagem com as funcionalidades oferecidas pelo sistema, descrevendo cenários de uso que envolvem o comportamento do sistema sob diferentes condições.

Pode-se também entender que, para Booch e Jacobson (2012), os diagramas de caso de uso permitem representar de maneira clara como os atores, como mostrados na figura 2, interagem com o sistema. Através de casos de uso, o diagrama captura funcionalidades específicas do sistema, descrevendo como ele se comporta

em diferentes cenários. Assim, o diagrama oferece uma visão do comportamento do sistema sob a perspectiva de quem o utiliza.

Figura 2 — Algumas figuras usadas do diagrama



Fonte: Ribeiro (2021).

Vale ressaltar que a elaboração de diagramas de caso de uso envolve alguns desafios, como garantir que os requisitos do sistema estejam corretamente representados. Para Guedes (2018), o principal objetivo dos diagramas de caso de uso é melhorar a comunicação entre os envolvidos no projeto, garantindo que todos compreendam de forma precisa os requisitos funcionais.

Sendo assim, Ribeiro (2012) afirma que os diagramas são amplamente usados na fase de levantamento de requisitos e ajudam a criar um entendimento comum sobre como o sistema deve se comportar, facilitando a implementação posterior. Faz-se necessário, portanto, entender os conceitos relacionados aos diagramas de caso de uso, pois eles são essenciais para a comunicação clara entre desenvolvedores, analistas e clientes, assegurando uma representação fiel das necessidades dos usuários e das funcionalidades do sistema.

### 2.2.2 Diagrama de Classe

Em Guedes (2018), encontramos que os diagramas de classe UML são fundamentais na modelagem de sistemas orientados a objetos, oferecendo uma representação estruturada das classes, seus atributos, operações e as relações entre objetos dentro de um sistema. Utilizados amplamente por engenheiros, esses diagramas ajudam a documentar arquiteturas de software, ilustrando o que deve estar presente no sistema a ser modelado. A estrutura básica de um diagrama de classe é composta por um retângulo dividido em três seções: o nome da classe, seus atributos e seus métodos, permitindo uma visão clara das funcionalidades e interações esperadas dentro do sistema.

Os benefícios dos diagramas de classe são diversos, incluindo a capacidade de ilustrar modelos de dados, facilitar a compreensão dos esquemas de uma aplicação, e expressar visualmente as necessidades específicas de um sistema. Eles proporcionam descrições independentes de implementação, permitindo que os desenvolvedores vejam a estrutura necessária para programação e implementação. Além disso, ao destacar elementos de código específicos que devem ser programados, os diagramas de classe são ferramentas valiosas para a comunicação interna em um projeto, garantindo que todos os *stakeholders* tenham uma compreensão comum da estrutura do sistema. (LUCIDCHART, 2024)

No que diz respeito às interações, em Guedes (2018), os diagramas de classe podem representar heranças, associações bidirecionais e unidirecionais entre classes, oferecendo uma visualização detalhada das relações no sistema. Esses diagramas não apenas facilitam a programação e documentação, mas também são essenciais para o planejamento e análise de sistemas complexos, assegurando que todas as partes do sistema sejam consideradas e compreendidas.

### 2.2.3 Diagrama de Atividade

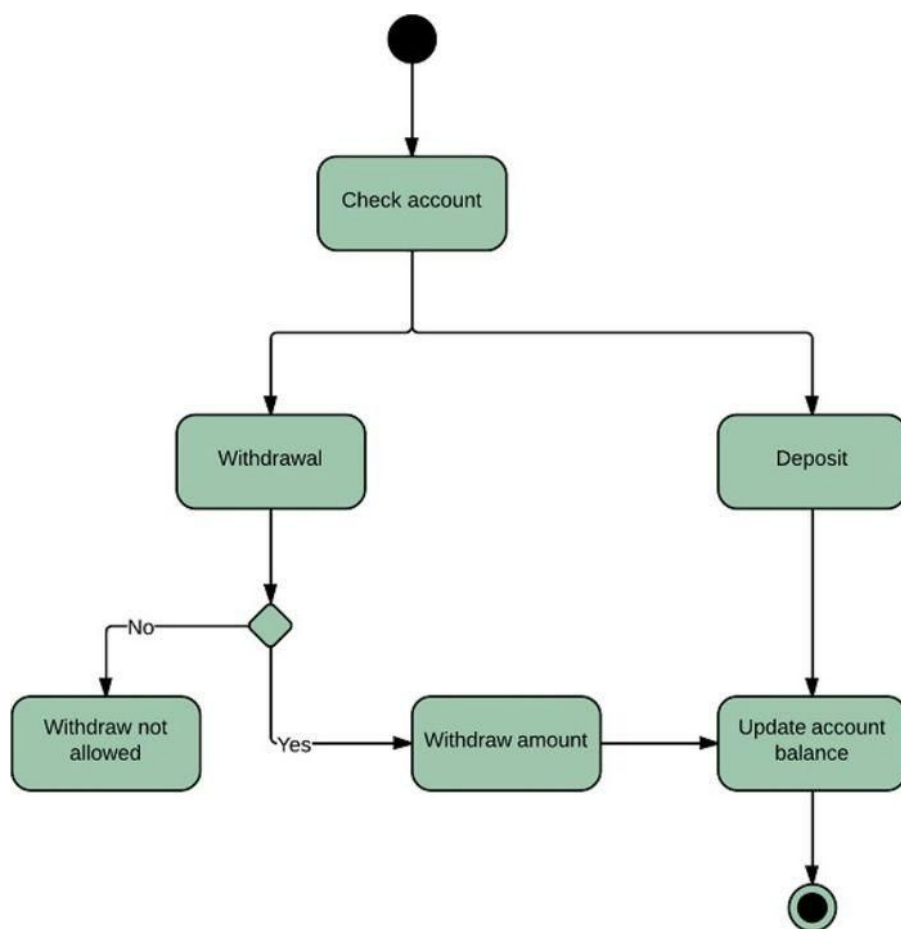
Os diagramas de atividades UML são uma ferramenta essencial para a visualização e documentação de sistemas de software, funcionando como um fluxograma que ilustra as atividades realizadas pelo sistema. Eles são categorizados como diagramas de comportamento, juntamente com diagramas de caso de uso e de máquina de estados, por descreverem os processos que devem ocorrer no sistema modelado. Esses diagramas promovem a clareza e a comunicação entre as áreas de negócios e desenvolvimento, permitindo que todas as partes interessadas compreendam e alinhem suas expectativas sobre o comportamento do sistema. Para criar um diagrama de atividade, utiliza-se um conjunto específico de símbolos que representam o início, término e decisões dentro do fluxo do processo. (LUCIDCHART, 2024a)

Os benefícios dos diagramas de atividades incluem a capacidade de demonstrar a lógica de algoritmos, descrever etapas em casos de uso UML e ilustrar processos de negócios ou fluxos de trabalho entre usuários e sistemas. Eles simplificam a compreensão de processos complexos, esclarecendo casos de uso complicados e modelando elementos de arquitetura de software, como métodos e

operações. Os componentes básicos de um diagrama de atividade incluem ações, nós de decisão, fluxos de controle, e os nós inicial e final, cada um representado por símbolos específicos que facilitam a visualização do fluxo de atividades.

Os diagramas de atividades, segundo Guedes (2018), são aplicáveis em diversos contextos, como no mapeamento do processo de *login* em uma página web ou na representação das operações de saque e depósito em sistemas bancários, o qual é representado na figura 3. Em um diagrama de atividade para uma página de *login*, por exemplo, o fluxo vai desde a entrada de nome de usuário e senha até o acesso bem-sucedido ao sistema, utilizando símbolos para atividades, decisões e notas. Esses diagramas são cruciais para a visualização de fluxos de processos, permitindo identificar áreas de melhoria ou destacar eficiência no processo, contribuindo assim para o planejamento e otimização de sistemas complexos.

Figura 3 — Exemplo de Diagrama de Atividade



Fonte: lucidchart.

Com base na análise realizada, o diagrama de atividade desempenha um papel fundamental na modelagem de sistemas, oferecendo uma representação visual clara e

estruturada dos fluxos de processos. Sua capacidade de detalhar a lógica de algoritmos, descrever casos de uso e ilustrar interações entre usuários e sistemas torna essa ferramenta indispensável tanto para o desenvolvimento quanto para o alinhamento entre áreas técnicas e de negócio. Além de facilitar a comunicação entre os envolvidos no projeto, os diagramas de atividade contribuem significativamente para a identificação de gargalos, a melhoria contínua e o aprimoramento da eficiência dos sistemas. Dessa forma, consolidam-se como um recurso valioso na engenharia de software, especialmente em contextos que exigem precisão, clareza e documentação eficaz dos comportamentos esperados no sistema.

### 2.3 PADRÕES DE PROJETO

Os padrões de projeto, ou *Design Patterns*, segundo (GAMMA et al., 2008), são soluções recorrentes para problemas comuns que surgem durante o desenvolvimento de software. Eles são descritos como uma forma de documentar a experiência de projetistas, oferecendo uma linguagem comum para discutir abordagens de design. Os padrões ajudam a formalizar e nomear as decisões de design, facilitando a comunicação entre desenvolvedores e promovendo a reutilização de soluções eficazes. Em vez de reinventar a roda a cada novo projeto, os desenvolvedores podem aplicar essas soluções testadas e comprovadas, economizando tempo e esforço.

*Design Patterns* são mais abstratos do que *frameworks*, que são implementações concretas em código. Enquanto os *frameworks* podem ser utilizados diretamente, os padrões precisam ser implementados em cada caso específico. Além disso, são menores em escopo, podendo ser aplicados em diversas situações, ao passo que um *framework* é geralmente voltado para um domínio específico. Essa flexibilidade torna os padrões de projeto uma ferramenta vital no desenvolvimento de software, especialmente em ambientes complexos e dinâmicos.

A utilização de padrões de projeto em aplicações web oferece diversas vantagens. Primeiramente, eles promovem a consistência no *design* e na implementação, o que facilita a manutenção e a evolução do software. Isso é crucial em aplicações web, onde as tecnologias e as demandas dos usuários estão em constante mudança. Ao adotar padrões, os desenvolvedores podem garantir que todos os membros da equipe sigam as mesmas diretrizes, reduzindo a probabilidade de erros e retrabalho (GAMMA et al., 2008).

Em segundo lugar, os padrões de projeto ajudam a melhorar a comunicação entre os membros da equipe. Com uma terminologia comum, é mais fácil discutir e justificar decisões de *design*, além de permitir que os novos integrantes da equipe se familiarizem rapidamente com a arquitetura do sistema. Isso é particularmente importante em projetos grandes e colaborativos, onde a clareza e a eficiência são essenciais.

Outra vantagem significativa é a possibilidade de reutilização de código e soluções. Com padrões bem definidos, os desenvolvedores podem aplicar soluções anteriores a novos problemas, economizando tempo e recursos. Isso não apenas acelera o processo de desenvolvimento, mas também contribui para a criação de um software de maior qualidade, já que as soluções foram testadas em situações anteriores. (FOWLER, 2002)

Por fim, a documentação e a catalogação de padrões de projeto promovem uma cultura de aprendizado contínuo dentro das equipes de desenvolvimento. Ao registrar e compartilhar padrões, as equipes podem aprender com as experiências passadas, evitando erros recorrentes e incentivando a inovação. Isso cria um ciclo virtuoso de melhoria contínua, essencial em um campo tão dinâmico quanto o desenvolvimento web.

### 2.3.1 Considerações na Escolha do Padrão

Ao selecionar um padrão de projeto, é fundamental compreender o problema que se deseja resolver. Cada padrão aborda um problema específico em um determinado contexto, e isso deve ser claramente identificado. A descrição do problema deve incluir as condições que precisam ser atendidas para que a aplicação do padrão faça sentido, ajudando a guiar a escolha do padrão mais adequado e garantindo que a solução proposta seja a mais eficaz. (GAMMA et al., 2008)

Após identificar o problema, é necessário avaliar as várias soluções que os padrões de projeto oferecem. Esses padrões não são implementações concretas, mas sim descrições abstratas que incluem elementos, responsabilidades e colaborações para resolver o problema em questão. Além disso, é importante considerar as consequências da aplicação de um padrão, incluindo vantagens e desvantagens, para

entender os *trade-offs*<sup>2</sup> envolvidos e como eles impactam a estrutura e a flexibilidade do sistema.

Ao decidir qual padrão utilizar, segundo (FOWLER, 2002), é importante também avaliar a complexidade da lógica de domínio. Um modelo de domínio é recomendado quando essa complexidade ultrapassa certos limites, embora não haja uma métrica clara para medi-la. Em geral, deve-se buscar profissionais experientes para uma análise inicial dos requisitos do projeto, permitindo uma escolha mais informada. A familiaridade da equipe com o padrão escolhido pode influenciar na eficácia do desenvolvimento, reduzindo custos e tempo de implementação.

Por fim, ao escolher um padrão de projeto, deve-se considerar sua aplicabilidade em diferentes situações e sua compatibilidade com as tecnologias e linguagens de programação utilizadas. Refletir sobre como o padrão se encaixa no ciclo de vida do desenvolvimento de software e como ele pode facilitar a reutilização e a manutenção do código é crucial. Os padrões de projeto devem ser adaptativos, permitindo que os desenvolvedores organizem e refaçam seus sistemas de maneira mais eficiente ao longo do tempo.

### 2.3.2 Mapeamento para Bancos de Dados Relacionais

O mapeamento para bancos de dados relacionais pode ser abordado de diferentes maneiras, dependendo se o esquema é criado do zero ou se um esquema existente deve ser utilizado. Quando o esquema é definido pelo desenvolvedor, pode-se optar por uma abordagem mais simples, utilizando *Row Data Gateway* ou *Table Data Gateway*. No entanto, ao trabalhar com um *Domain Model*, é essencial evitar que o *design* do modelo de domínio se assemelhe ao design do banco de dados, garantindo que a lógica de domínio permaneça independente da estrutura de armazenamento como afirma Gamma et al. (2008)

### 2.3.3 Integração e Iteração

A integração e a iteração em projetos de software são processos interligados que garantem o funcionamento coeso e o aprimoramento contínuo de um sistema. A

---

<sup>2</sup> *trade-offs* – Em projetos de software refere-se à necessidade de fazer escolhas entre duas ou mais opções, onde uma escolha pode levar a benefícios em um aspecto, mas a custos em outro. Essa dinâmica é comum na arquitetura de software e no design de sistemas, onde diferentes abordagens podem impactar desempenho, manutenção, escalabilidade e complexidade.

integração refere-se à maneira como diferentes componentes ou sistemas se unem para operar como um todo, enquanto a iteração envolve o desenvolvimento repetido e a melhoria contínua do sistema. O conhecimento dos padrões de projeto é essencial para facilitar ambos os processos, pois fornecem uma linguagem comum e soluções para problemas recorrentes, tornando a integração mais fluida e a iteração mais eficaz.

Ao aplicar padrões de projeto, os desenvolvedores, mesmo os menos experientes, podem evitar confusões associadas à herança e trabalhar de forma mais eficiente, similar a profissionais experientes. Durante a iteração, esses padrões guiam as decisões dos projetistas, permitindo que eles apliquem soluções previamente testadas e documentadas, economizando tempo e esforço. A documentação de padrões registra experiências passadas, promovendo a aprendizagem e a melhoria contínua, crucial para a evolução do software. (GAMMA et al., 2008)

Segundo Fowler (2002), a integração do modelo de domínio com o banco de dados deve ocorrer de maneira iterativa, construindo partes do modelo e do banco ao longo do desenvolvimento. Isso permite *feedback* contínuo e evita problemas de desempenho que podem surgir de um modelo de domínio mal projetado. Ao mapear para um esquema existente, as classes de *Row Data Gateway* ou *Table Data Gateway* devem ser criadas para refletir a estrutura do banco, garantindo que a lógica de domínio possa ser aplicada sobre essa base de dados. Essa abordagem gradativa ajuda a mitigar riscos e a criar uma fundação sólida para a aplicação.

Os benefícios combinados da integração e iteração incluem uma melhor compreensão entre as equipes, eficiência no desenvolvimento, flexibilidade para se adaptar a novas necessidades e uma melhoria contínua do software. Além disso, esses processos reduzem riscos ao identificar problemas potenciais em estágios iniciais e aumentam a qualidade do design final através de revisões iterativas. Em última análise, a integração e a iteração usando padrões de projeto são fundamentais para o sucesso em ambientes de desenvolvimento de software orientado a objetos, onde a complexidade pode ser um desafio significativo.

#### 2.3.4 Padrão de design Método Factory

Ao abordar o *Factory*, também conhecido como Método Fábrica, observa-se uma solução elegante para a criação de objetos. Esse padrão define uma interface genérica para a criação, mas permite que as subclasses assumam a responsabilidade

de decidir qual classe específica deve ser instanciada. Essa abordagem confere uma flexibilidade notável ao design de *software*, especialmente em cenários onde o tipo exato do objeto a ser criado não pode ser determinado previamente. Essencialmente, o padrão em questão, possibilita que uma classe delegue a tarefa de instanciar objetos a suas subclasses, estabelecendo uma clara separação entre a criação e a utilização dos objetos.

A aplicação do Método *Factory*, segundo Fowler (2002), se manifesta em duas variações principais. A primeira envolve uma classe *Creator* abstrata, que não fornece uma implementação para o método fábrica, exigindo que as subclasses o implementem. A segunda variação permite que a classe *Creator* seja concreta, oferecendo uma implementação padrão do método fábrica. Isso é particularmente útil quando a classe do objeto a ser criado é desconhecida, pois as subclasses podem redefinir o processo de criação, assegurando que a lógica de construção permaneça flexível e adaptável. Além disso, o Método *Factory* pode ser estendido para incluir métodos parametrizados, que aceitam argumentos para identificar o tipo de objeto a ser criado. Essa estratégia é vantajosa ao se desejar criar diferentes variantes de um produto a partir de uma única fábrica, o que facilita a extensão e a manutenção do código. Assim, as subclasses podem modificar o comportamento da fábrica sem afetar diretamente a classe base, resultando em uma arquitetura de software mais limpa e organizada.

Entre os benefícios do *Factory*, destaca-se a redução da dependência entre classes, uma vez que os clientes interagem apenas com interfaces abstratas, e não com classes concretas. Isso também simplifica a troca de implementações, pois novos tipos de produtos podem ser incorporados sem a necessidade de alterar o código do cliente. Contudo, uma desvantagem inerente é que a proliferação de subclasses pode introduzir uma complexidade desnecessária ao sistema, especialmente quando um grande número de variações de produtos se faz necessário. Em suma, tal padrão de projeto, se revela uma ferramenta poderosa para desenvolvedores que buscam construir sistemas flexíveis e escaláveis, permitindo que a lógica de criação de objetos seja facilmente adaptada às necessidades em constante evolução de um projeto.

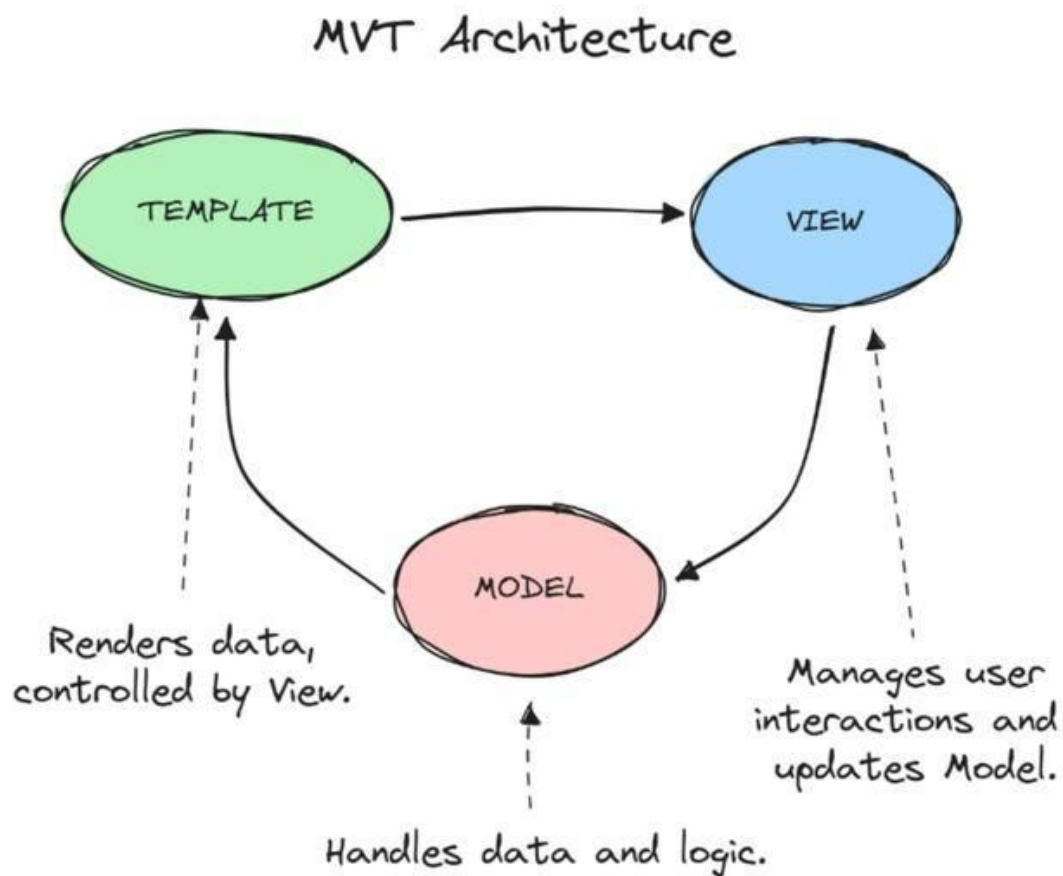
### 2.3.5 Padrão MVT

O padrão MVT, ou *Model-View-Template*, apresenta várias vantagens que o tornam uma escolha popular no desenvolvimento de aplicações web. Primeiramente, a separação clara de responsabilidades entre o modelo, a visualização e o *template* permite uma melhor organização do código. Isso facilita a manutenção e a escalabilidade da aplicação, pois mudanças em uma camada não afetam diretamente as outras. Além disso, a reutilização de componentes é incentivada, permitindo que desenvolvedores possam criar interfaces consistentes sem duplicar o código.

Outra vantagem significativa do MVT é a facilidade de teste. Como o modelo e a visualização estão desacoplados, é mais simples testar a lógica de negócios sem a necessidade de interagir com a interface do usuário. Essa abordagem resulta em uma base de código mais robusta e menos propensa a erros, já que a lógica pode ser validada independentemente da apresentação. Isso é especialmente benéfico em projetos de maior complexidade, onde a integração de várias partes do sistema pode introduzir falhas sutis. A incorporação de padrões de projeto traz várias vantagens, como a reusabilidade de código e soluções, permitindo aos desenvolvedores aplicarem métodos comprovados a problemas recorrentes, o que economiza tempo e esforço. (FOWLER, 2002)

Por fim, o padrão MVT, representado na figura 4, também se destaca por sua flexibilidade. Ele permite que desenvolvedores criem diferentes apresentações da mesma lógica de negócios, facilitando a adaptação a novos requisitos ou mudanças nas preferências dos usuários. Isso é vital em um ambiente de desenvolvimento ágil, onde as necessidades dos usuários podem evoluir rapidamente. A capacidade de alterar a camada de apresentação sem reescrever a lógica do modelo representa uma economia de tempo significativa e um aumento na eficiência do desenvolvimento. Por outro lado, Yahya (2023) reconhece que a implementação de padrões também pode apresentar desafios, como a complexidade adicional que eles podem introduzir no código, especialmente para desenvolvedores menos experientes.

Figura 4 — Relacionamento MVT



Fonte: Yahya (2023).

Em conclusão, os padrões de projeto são ferramentas valiosas que, quando usados de forma consciente e apropriada, podem transformar significativamente a maneira como aplicações são desenvolvidas. Eles não só promovem boas práticas e padrões da indústria, mas também facilitam a comunicação e colaboração entre equipes, garantindo que o desenvolvimento de software seja mais eficiente e alinhado com as melhores práticas.

## 2.4 LINGUAGEM DE PROGRAMAÇÃO PYTHON

Python, segundo Alura (2024), destaca-se como uma linguagem de programação versátil, caracterizada por ser interpretada, interativa e orientada a objetos, com uma sintaxe simples e clara. Essa linguagem suporta múltiplos paradigmas de programação, como os paradigmas procedural e funcional, o que a torna adequada para uma ampla gama de aplicações, desde scripts simples até

sistemas complexos. A vasta gama de módulos e bibliotecas disponíveis permite aos desenvolvedores implementar funcionalidades de maneira eficaz, abrangendo áreas como processamento de strings, protocolos de internet e interfaces de sistema operacional.

Uma das principais vantagens de Python é sua facilidade de aprendizado, especialmente para iniciantes. A sintaxe intuitiva e a tipagem dinâmica facilitam a compreensão e a implementação de conceitos de programação, permitindo que novos programadores se concentrem em habilidades essenciais, como decomposição de problemas e design de dados, sem se perder em complexidades de linguagem, comuns em linguagens estaticamente tipadas. Python é acompanhada por uma extensa biblioteca padrão que cobre uma ampla gama de funcionalidades, incluindo manipulação de arquivos, testes de unidade e comunicação via protocolos de internet. Essa biblioteca rica possibilita que desenvolvedores realizem tarefas complexas com menos código, aumentando a produtividade. Além disso, muitas extensões de terceiros estão disponíveis, ampliando ainda mais as capacidades da linguagem. (DOWNEY, 2010)

A comunidade Python é uma das mais ativas e envolventes no mundo da programação, oferecendo uma rica oferta de recursos, como tutoriais, documentação e fóruns de discussão. O suporte da comunidade facilita o aprendizado e a resolução de problemas, além de promover a troca de conhecimento entre desenvolvedores de diferentes níveis de experiências tornando-a uma escolha popular em ambientes corporativos e acadêmicos, onde a interoperabilidade é essencial.

#### 2.4.1 **Framework Django**

O Django é um *framework* web de alto nível desenvolvido em Python, que facilita a criação de aplicações web robustas e interativas. Projetado para promover um desenvolvimento limpo e ágil, ele permite que os desenvolvedores se concentrem nas regras de negócio da aplicação, evitando a complexidade técnica excessiva. É amplamente utilizado para construir desde sites simples até sistemas complexos de gerenciamento de dados, beneficiando-se de sua estrutura modular e flexível. Além disso, o Django é uma escolha poderosa para desenvolvedores que buscam um framework que não apenas simplifica o desenvolvimento web, mas também garante segurança, escalabilidade e flexibilidade.

O Django adota o padrão arquitetural *Model-View-Template* (MVT), que separa a lógica de negócios da apresentação e do acesso a dados, promovendo uma organização clara do código. Ele oferece uma interface administrativa pronta para uso, simplificando a gestão de dados e usuários. Em termos de segurança, o Django incorpora diversas proteções contra-ataques comuns, como *Cross Site Scripting* e *SQL Injection*, assegurando a integridade das aplicações. O *framework* é projetado para suportar o aumento de carga, tornando-se adequado para projetos de diversos tamanhos. A capacidade de reutilização de componentes permite a criação de aplicativos modulares aplicáveis em diferentes projetos, enquanto o suporte à internacionalização facilita a tradução de aplicações em vários idiomas.

O uso do Django para desenvolvimento web proporciona várias vantagens. Ele oferece ferramentas e bibliotecas que aceleram o desenvolvimento, permitindo a criação rápida e eficiente de aplicações. A comunidade ativa de desenvolvedores contribui para a evolução contínua do *framework*, oferecendo suporte e recursos adicionais. O Django é flexível, adequado para uma ampla variedade de aplicações, facilita a criação de testes automatizados, garantindo que as aplicações funcionem corretamente à medida que evoluem. A estrutura organizada do Django simplifica a manutenção das aplicações, com uma clara separação entre componentes.

O sistema de mapeamento objeto-relacional (ORM) do Django assegura que as operações de banco de dados sejam realizadas de forma segura e consistente. O *Django REST Framework* complementa essa capacidade, permitindo a construção de *APIs REST-ful* e facilitando a integração com outras aplicações e serviços. O Django é altamente personalizável, permitindo que os desenvolvedores adaptem suas funcionalidades às necessidades específicas de cada projeto. A ênfase no princípio “Don’t Repeat Yourself” melhora a reutilização do código, evitando repetições e promovendo um código modular. (ALCHIN, 2009)

Em resumo, o Django é um *framework* poderoso que simplifica o desenvolvimento web, cuidando de aspectos complexos como a interface administrativa e a autenticação de usuários. Sua abordagem modular e a forte ênfase na reutilização de código tornam-no uma escolha ideal para desenvolvedores que buscam criar desde pequenos sites até grandes aplicações mais robustas. A documentação abrangente e a comunidade ativa são recursos valiosos que apoiam os desenvolvedores na compreensão e aplicação das melhores práticas do Django.

## 2.5 POO

A Programação Orientada a Objetos (POO) tem como objetivo estruturar o código ao redor de "objetos", para representar tanto entidades do mundo real quanto conceitos abstratos. Essa abordagem organiza o sistema de forma modular e intuitiva, permitindo que o código seja reutilizado e mantido com mais facilidade. Em Python, as classes são os elementos fundamentais para a criação de objetos, sendo compostas por atributos que caracterizam as propriedades desses e por métodos que descrevem seus comportamentos. Essa organização possibilita a união de dados e comportamentos de forma coesa, reforça a integridade do código utilizando, por exemplo, classes abstratas e interfaces, que estabelecem contratos de como as classes derivadas devem seguir. (DOWNEY, 2010)

### 2.5.1 Abstração

A abstração é um dos conceitos fundamentais da programação orientada a objetos, permitindo que os programadores ocultem detalhes complexos e exponham apenas as funcionalidades essenciais. Em Python, a abstração facilita o desenvolvimento de programas mais simples e gerenciáveis, permitindo que os desenvolvedores manipulem objetos sem se preocupar com suas implementações internas. Por meio do uso de classes e interfaces, a abstração ajuda a criar sistemas que são mais fáceis de entender e manter, pois os detalhes da implementação são encapsulados dentro das classes.

No Python, a abstração é frequentemente implementada através do uso de classes abstratas e métodos abstratos, que fornecem uma estrutura para classes derivadas. As classes abstratas definem uma interface que outras classes devem seguir, mas não podem ser instanciadas diretamente. Isso garante que as classes derivadas implementem os métodos definidos, promovendo uma arquitetura de código mais robusta e flexível. Por exemplo, ao criar uma classe `Animal` com métodos abstratos como `fazersom`, qualquer classe que herde de `Animal`, como `Cachorro` ou `Gato`, deve implementar esse método, assegurando que todos os animais tenham uma

forma de fazer som, mas de maneira específica para cada tipo (LABAKI; WOISHI, 2016).

Além disso, a abstração em Python promove reuso de código, uma vez que permite que os desenvolvedores criem soluções genéricas que podem ser adaptadas para diferentes contextos. Isso é particularmente útil em projetos grandes, onde a manutenção e a escalabilidade são essenciais. Através da abstração, o código se torna mais modular, permitindo que partes do sistema sejam alteradas ou atualizadas sem impactar outras, o que é crucial para a evolução de software ao longo do tempo.

Esta abordagem não apenas melhora a legibilidade do código, mas também contribui para a redução de erros, já que os desenvolvedores podem focar nas interações entre objetos em vez de se perderem em detalhes de implementação. Em resumo, a abstração é uma ferramenta poderosa na programação orientada a objetos em Python, que ajuda a simplificar a complexidade e a criar sistemas mais eficientes e de fácil manutenção.

### 2.5.2 Herança

A herança é um dos princípios fundamentais da programação orientada a objetos em Python. Ela permite que uma classe herde atributos e métodos de outra classe. Isso promove a reutilização de código e facilita a criação de hierarquias de classes. Por exemplo, se tivermos uma classe `Animal` que define características comuns a todos os animais, podemos criar classes específicas como `Cachorros` e `Galinhas` que herdam essas características, evitando a duplicação de código e melhorando a organização do programa (LABAKI; WOISHI, 2016).

Em Python, a herança é implementada especificando a classe pai entre parênteses ao definir a classe filha. Isso permite que a subclasse acesse os métodos e atributos da superclasse. Por exemplo, uma subclasse pode sobrescrever um método da superclasse para fornecer uma implementação mais específica. Além disso, Python suporta herança múltipla, onde uma classe pode herdar mais de uma classe. Isso é útil quando uma classe precisa combinar comportamentos de diferentes classes, mas deve ser usado com cautela para evitar complexidade e ambiguidade.

### 2.5.3 Polimorfismo

O polimorfismo é um conceito fundamental na programação orientada a objetos, que permite que objetos de diferentes classes sejam tratados como objetos da mesma classe através de uma interface comum. No contexto do Python, o polimorfismo pode ser observado através do "duck typing", que se baseia na ideia de que um objeto pode ser utilizado de forma intercambiável com outro se ele possui os métodos e atributos necessários, independentemente de sua classe real. Isso significa que, desde que um objeto possua a estrutura necessária para responder a uma chamada de método, ele pode ser utilizado como se fosse de um tipo específico, promovendo flexibilidade no código e facilitando a reutilização de componentes.

Um exemplo prático de polimorfismo em Python é a utilização de métodos com o mesmo nome em diferentes classes. Quando uma classe herda de outras classes, como mostrado no exemplo com as classes A, B e C, o método `run()` pode ser implementado de maneira diferente em cada classe. Quando um objeto da classe C é criado, ele pode chamar o método `run()`, que, devido à herança múltipla, pode buscar a implementação mais apropriada nas superclasses. Isso demonstra como o polimorfismo permite que diferentes tipos de objetos respondam a chamadas de métodos de maneira específica, dependendo de sua implementação. (CRUZ, 2022)

Além disso, o polimorfismo em Python é suportado pela capacidade de sobrescrever métodos. Isso permite que cada classe defina sua própria representação em formato de *String*, proporcionando uma maneira de personalizar a saída de objetos de acordo com suas características. Essa personalização é uma aplicação prática do polimorfismo, permitindo que métodos comuns se comportem de forma diferente dependendo do objeto que os invoca, enriquecendo a experiência de desenvolvimento e a legibilidade do código.

#### 2.5.4 Encapsulamento

O encapsulamento é um dos princípios fundamentais da programação orientada a objetos, e no Python, ele é implementado de maneira específica. Em Cruz (2022), é enfatizado que, em Python, não existem modificadores de acesso como *public* ou *private* encontrados em outras linguagens, como Java. Em vez disso, o encapsulamento é alcançado através da convenção de usar um sublinhado, para indicar que um atributo ou método deve ser tratado como não público. Essa abordagem é uma forma de sugerir que esses elementos não devem ser acessados diretamente

fora da classe, embora essa restrição não seja rigorosa, já que Python é uma linguagem dinâmica e flexível.

Uma das práticas recomendadas para implementar o encapsulamento em Python é o uso de métodos chamados *getters* e *setters*. Por exemplo, ao criar uma classe, pode-se definir um método que retorna um atributo privado, permitindo que o acesso a esse dado seja controlado. O uso do decorador *@property* é uma maneira elegante de expor atributos privados, permitindo que sejam acessados como se fossem públicos, sem expor diretamente a implementação subjacente. Isso proporciona uma interface mais limpa e mantém a integridade dos dados, já que a lógica de acesso pode ser encapsulada em métodos. (CRUZ, 2022)

Além disso, o autor menciona que o encapsulamento em Python se alinha com o princípio "Explícito é melhor que implícito", presente no Zen da linguagem em questão. Isso significa que, ao definir claramente como os atributos e métodos devem ser acessados, os desenvolvedores podem criar códigos mais legíveis e manuteníveis. O uso do encapsulamento não apenas protege os dados, mas também facilita a manutenção e a evolução do código, permitindo que mudanças internas sejam feitas sem impactar diretamente as partes do programa que dependem desses dados.

## 2.6 BANCO DE DADOS

Um banco de dados é uma coleção organizada de informações ou dados estruturados, armazenados eletronicamente em um sistema de computador. Geralmente gerido por um Sistema de Gerenciamento de Banco de Dados (SGBD), um banco de dados compreende os dados, o SGBD e os aplicativos associados, formando um sistema de banco de dados. Os dados são organizados de maneira a facilitar a busca e recuperação por um computador, comumente modelados em tabelas de linhas e colunas para otimizar o processamento e a consulta de dados de forma eficiente (Elmasri e Navathe, 2006). A interação e manipulação dos dados são frequentemente realizadas utilizando a Linguagem de Consulta Estruturada (SQL).

Ao longo do tempo, os bancos de dados evoluíram, abrangendo diferentes tipos, como bancos de dados hierárquicos, de rede, relacionais, orientados a objetos e NoSQL, cada um utilizando modelos distintos para organizar as informações. Por exemplo, segundo Elmasri e Navathe (2006), os bancos de dados hierárquicos utilizam uma estrutura em árvore, enquanto os bancos orientados a objetos armazenam dados

complexos em objetos que podem herdar propriedades de classes superiores na hierarquia. Com o aumento da coleta de dados, os bancos de dados tornaram-se essenciais para as empresas na análise e tomada de decisões. Organizações estão adotando tecnologias inovadoras, como bancos de dados autônomos, baseados em nuvem e com *Machine Learning*, para automatizar processos, garantir segurança, melhorar o desempenho e proporcionar agilidade no gerenciamento de dados em meio aos desafios de escala e complexidade.

A maioria dos sistemas desenvolvidos envolve bancos de dados e, embora o foco muitas vezes esteja no código-fonte, a base de dados, para Elmasri e Navathe (2006), sua projeção e gestão representam o "núcleo" do sistema em termos de valor agregado ao negócio, devido a fatores como segurança, performance e políticas organizacionais, que tornam essa "camada" das aplicações extremamente relevante, merecendo atenção especial dos desenvolvedores.

A maneira como um banco de dados é projetada, organizada e otimizada pode impactar diretamente na eficiência e segurança do sistema. Desde a escolha correta dos tipos de dados, passando pela utilização de índices e a implementação de *procedures* e *views*, até práticas mais avançadas como o monitoramento de *queries* e o uso de *caching*, cada decisão pode influenciar o desempenho da aplicação (OLIVEIRA, 2010).

Portanto, é essencial que os desenvolvedores dediquem tempo à modelagem correta dos bancos de dados, aplicando as melhores práticas para garantir que as informações sejam armazenadas, acessadas e manipuladas de forma eficiente. Um banco de dados bem projetado constitui um dos pilares fundamentais para o sucesso de qualquer sistema, seja ele uma pequena aplicação ou uma solução empresarial em larga escala.

### 2.6.1 MySQL

O MySQL, um dos sistemas gerenciadores de banco de dados relacionais mais populares e utilizados globalmente, destaca-se como uma ferramenta essencial no desenvolvimento de aplicações web e empresariais modernas. Sua reputação é construída sobre pilares como eficiência, confiabilidade e facilidade de uso, características que o consagraram como a escolha de muitos desenvolvedores que buscam uma solução robusta para o gerenciamento de dados. Conforme ressalta

MySQL (2024), "o MySQL oferece uma combinação única de simplicidade e poder, permitindo que desenvolvedores implementem soluções de banco de dados de forma rápida e eficaz". Essa combinação de fatores o torna um componente fundamental na arquitetura de sistemas contemporâneos, facilitando a interação entre aplicações e dados de maneira otimizada.

A importância dele transcende sua popularidade, abrangendo um conjunto abrangente de vantagens que o diferenciam no mercado. Sua eficiência e confiabilidade, amplamente reconhecidas na comunidade de desenvolvedores, proporcionam uma base sólida para a implementação de soluções robustas e escaláveis, capazes de atender às demandas de aplicações web modernas. Sua facilidade de uso, evidenciada por sua interface amigável e simplicidade na gestão de dados e execução de consultas, democratiza o acesso ao gerenciamento de banco de dados, permitindo que desenvolvedores de diferentes níveis de experiência o utilizem com eficácia. A segurança, um pilar fundamental na era digital, é tratada com a devida importância no MySQL, que incorpora funcionalidades avançadas para proteger os dados contra acessos não autorizados e ataques maliciosos, garantindo a integridade das aplicações.

A flexibilidade e escalabilidade do mesmo são características cruciais para aplicações web que precisam se adaptar a diferentes cenários de uso e acompanhar o crescimento da demanda. Desde pequenos *websites* pessoais até grandes plataformas de *e-commerce*, demonstrando sua capacidade de se ajustar a diferentes escalas, garantindo que o desempenho se mantenha otimizado mesmo com o aumento do volume de dados. O suporte a diversos tipos de dados, incluindo funcionalidades avançadas como replicação e *clustering*, permite que, segundo Carvalho (2022), o MySQL atenda às necessidades de aplicações que exigem alta disponibilidade e desempenho. A comunidade ativa de desenvolvedores, que contribui continuamente para a evolução do SGBD em questão, oferece suporte, recursos adicionais e facilita a troca de conhecimentos, enriquecendo o ecossistema e impulsionando a inovação.

Além dos aspectos técnicos, o sistema de gerenciamento de banco de dados em questão também se destaca por sua capacidade de reduzir a redundância de dados, otimizar a organização e melhorar a qualidade geral dos sistemas de dados. Sua integração facilitada com outros sistemas permite que as aplicações *web* troquem dados de forma fluida entre diferentes plataformas, promovendo a interoperabilidade e a eficiência. O suporte à internacionalização, que facilita a tradução de aplicações para

diversos idiomas, é um diferencial importante para empresas que atuam em mercados globais. E, não menos importante, o custo acessível do MySQL, como uma ferramenta gratuita e de código aberto, o torna uma opção atrativa para desenvolvedores e empresas de todos os portes, permitindo que utilizem um sistema de gerenciamento de banco de dados de alta qualidade sem custos adicionais. (CARVALHO, 2022)

Em suma, é uma escolha preferencial para o desenvolvimento de aplicações web, proporcionando uma base sólida para a gestão eficiente e segura de dados. Sua combinação de eficiência, confiabilidade, facilidade de uso, segurança, flexibilidade, escalabilidade e custo-benefício o torna uma ferramenta indispensável para desenvolvedores que buscam construir aplicações web modernas e robustas. A constante evolução do MySQL, impulsionada por uma comunidade ativa e engajada, garante que ele continue a ser uma solução líder no mercado de gerenciamento de banco de dados, acompanhando as últimas tendências e necessidades do mundo digital.

## 2.7 ADMINISTRAÇÃO PÚBLICA ESTADUAL PATRIMONIAL

A Administração Pública Estadual Patrimonial refere-se à gestão de bens e recursos públicos dentro do contexto de um estado. Tal setor é responsável por assegurar que os bens móveis e imóveis sejam utilizados de forma eficaz e em conformidade com as normas legais. A prática de controle patrimonial é fundamental, pois garante a transparência e a *accountability*<sup>3</sup> na utilização dos recursos públicos, promovendo uma administração ética e responsável. Profissionais dessa área devem estar aptos a entender e aplicar as leis pertinentes, além de gerenciar adequadamente os ativos do estado em suas diversas categorias, como bens duráveis, descartáveis e recicláveis (SCOTT, 2021).

O controle patrimonial envolve a aplicação de regras e ações institucionais que são apoiadas por recursos humanos qualificados e tecnologias adequadas. Isso inclui a competência de registrar e reportar informações patrimoniais na contabilidade pública, assegurando que todos os atos administrativos sejam devidamente documentados e auditáveis. Essa estrutura organizacional é essencial para que os gestores públicos

---

<sup>3</sup> *Accountability* – no contexto da administração pública e da gestão patrimonial, refere-se à responsabilidade e à transparência na administração dos recursos públicos. Esse conceito implica que os gestores devem prestar contas sobre suas ações e decisões, garantindo que a utilização dos bens públicos esteja em conformidade com as normas e leis vigentes.

possam monitorar e avaliar a situação dos bens sob sua responsabilidade, evitando desperdícios e irregularidades.

Além disso, a classificação dos bens patrimoniais é uma prática importante na gestão pública. Os bens podem ser categorizados como duráveis, com uma vida útil estimada de vários anos; descartáveis, que têm uma vida útil limitada; ou danificados, que requerem reparos. A correta classificação permite um gerenciamento mais eficiente e facilita a tomada de decisões sobre a manutenção, aquisição ou descarte desses bens. A transparência na administração dos bens estaduais também é crucial para a confiança pública e a integridade das instituições governamentais.

Por fim, a Administração Pública Estadual Patrimonial não apenas trata da gestão de bens, mas também da formação e aprimoramento contínuo dos profissionais envolvidos. A necessidade de conhecimentos atualizados e práticas eficazes é um desafio constante, especialmente em um ambiente em que as exigências legais e sociais estão em constante evolução. A capacitação e a educação contínua dos funcionários públicos são essenciais para garantir que a administração patrimonial atenda às expectativas da sociedade e cumpra seu papel de forma responsável e eficaz.

## 2.8 CONCEITOS DE PATRIMÔNIO PÚBLICO

A presente seção tem como foco discorrer sobre a gestão patrimonial no setor público e sua importância para assegurar a eficiência e transparência na utilização dos bens públicos. Tal abordagem é necessária para que se compreenda a relevância de normas e regulamentos que visam garantir a correta administração dos bens públicos, evitando extravios e uso inadequado. O patrimônio público, compreendido como o conjunto de bens, direitos e valores que pertencem ao Estado e são destinados à prestação de serviços à população, abrange tanto bens tangíveis, como imóveis e equipamentos, quanto intangíveis, como direitos autorais e marcas. A gestão eficaz desse patrimônio é de crucial importância para assegurar a transparência e a eficiência na administração pública, elementos indispensáveis para a consolidação da confiança da sociedade nas instituições estatais.

Com base (SCOTT, 2021), é possível notar como o tema vem sendo abordado, em produções científicas, de forma que enfatiza a necessidade de controle sistemático e a implementação de procedimentos contábeis adequados para assegurar que os

registros patrimoniais sejam precisos e reflitam a realidade dos bens públicos. Também é possível perceber que o assunto apresenta uma relação direta com a responsabilidade fiscal e os princípios constitucionais que orientam a administração pública.

O patrimônio público se distingue por características específicas que delineiam sua natureza e importância. Entre elas, destacam-se os bens tangíveis, que englobam ativos físicos como imóveis, veículos, máquinas e equipamentos, que servem de suporte material para a realização das atividades estatais. Há também os bens intangíveis, que compreendem ativos não físicos, como direitos autorais, patentes, marcas, softwares e outros elementos de propriedade intelectual, que representam um valor significativo para o Estado.

Os bens que integram o patrimônio público são destinados ao uso e benefício de toda a coletividade, visando atender às necessidades e demandas da população. A gestão do patrimônio público exige a implementação de sistemas de controle rigorosos, como a gestão patrimonial, que garantam a utilização eficiente e responsável dos bens, bem como a prevenção de desvios e irregularidades. O patrimônio público é regido por leis e normas específicas que estabelecem os critérios para a sua administração, utilização e alienação, assegurando a sua correta gestão e preservação.

Os gestores públicos possuem a responsabilidade de zelar pelo bom uso e conservação do patrimônio público, adotando medidas preventivas e corretivas para evitar danos, perdas ou desvios. A gestão do patrimônio público deve ser transparente, permitindo o acompanhamento e a fiscalização por parte da sociedade, como forma de garantir a sua correta aplicação e evitar o uso indevido dos recursos públicos.

O tombamento é um processo administrativo que visa registrar bens de valor histórico, cultural ou artístico, protegendo-os de eventuais ameaças de destruição ou degradação, e preservando-os para as futuras gerações.

Ainda no que diz respeito à formação e treinamento dos servidores responsáveis pela gestão patrimonial, é necessário que esses profissionais compreendam as normas locais e a estrutura administrativa da entidade, contribuindo efetivamente para a administração dos bens públicos.

A interação entre as áreas envolvidas na gestão patrimonial pode resultar em uma gestão mais eficiente e transparente, assegurando que os recursos públicos sejam utilizados de maneira responsável e em benefício da coletividade. A administração eficiente do patrimônio público desempenha um papel fundamental na concretização

dos objetivos do Estado, sendo essencial para a prestação de serviços públicos de qualidade à população, nas áreas de saúde, educação, segurança, transporte, entre outras.

Uma gestão eficiente do patrimônio público contribui para o fortalecimento do controle interno na administração pública, garantindo que os bens sejam utilizados de forma eficiente e responsável, em conformidade com os princípios da legalidade, economicidade e eficiência. A gestão transparente e rigorosa do patrimônio público minimiza os riscos de corrupção, desvio de recursos e outras irregularidades, fortalecendo a integridade da administração pública e a confiança da sociedade. O patrimônio público abrange também bens de valor cultural e histórico, cuja preservação é fundamental para a manutenção da identidade e da memória do país, bem como para o desenvolvimento do turismo e da cultura. Machado (2013)

Fica claro, portanto, que a gestão patrimonial no setor público não se resume apenas ao controle e registro de bens, mas também envolve a otimização de sua utilização, respeitando os princípios constitucionais e promovendo a responsabilidade fiscal.

### **2.8.1 Gestão Patrimonial**

A Gestão Patrimonial na administração pública é um pilar fundamental para assegurar que os bens públicos sejam administrados em estrita conformidade com as leis e normas vigentes. Nesse contexto, o estudo de Scott (2021) se destaca ao analisar os aspectos patrimoniais relacionados aos bens móveis na Prefeitura Municipal de Gravataí, enfatizando a importância do tombamento, bem como a identificação, reconhecimento, mensuração e evidenciação desses bens. Essa gestão não apenas busca garantir a integridade dos ativos, mas também almeja aprimorar o controle patrimonial e a transparência das informações, refletindo um compromisso inegociável com a responsabilidade fiscal e a boa governança. O tombamento, nesse cenário, configura-se como um ato administrativo de suma relevância, direcionado à proteção de elementos com valor histórico, cultural ou ambiental, assegurando tanto sua preservação quanto seu uso apropriado. Essa ferramenta é vital para o controle patrimonial, permitindo a identificação e classificação dos bens e, conseqüentemente, que sua utilização esteja alinhada às normas legais e administrativas.

Scott (2021) reforça que a gestão patrimonial é uma obrigação e exige o cumprimento de procedimentos específicos para o levantamento físico dos bens, sua avaliação e reavaliação. Para tal, os órgãos públicos precisam dispor de meios adequados que garantam a conformidade com as normatizações e a execução das atividades que envolvem a gestão do patrimônio, evitando assim, a instauração de processos administrativos disciplinares. O processo de tombamento, em particular, engloba etapas cruciais como a identificação, reconhecimento, mensuração e evidenciação dos bens móveis, todas rigidamente pautadas em leis e regulamentos específicos. Uma vez tombados, esses bens necessitam de um registro com número de identificação único, facilitando o controle e a gestão, e seu registro é fundamentado no valor de aquisição e outras características relevantes. A pesquisa de Scott (2021) ainda sugere que a interação entre diferentes áreas administrativas e o treinamento adequado podem impulsionar uma evolução significativa na gestão e no controle dos bens patrimoniais. A eficácia desses processos é, portanto, diretamente proporcional à capacitação dos servidores envolvidos, que devem compreender as etapas e a legislação pertinente para garantir a conformidade e a eficiência das ações.

Contudo, é imperativo reconhecer que a área da gestão patrimonial não está isenta de desafios. Um dos pontos mais críticos é a necessidade urgente de melhorias nos sistemas utilizados para registrar e controlar os bens, com o objetivo de mitigar a duplicidade de registros. Estudos de caso, como os conduzidos na Prefeitura Municipal de Gravataí, evidenciam como a documentação e os procedimentos adequados são essenciais para um controle efetivo dos bens patrimoniais. Diante desse panorama, torna-se claro que a administração pública deve implementar ações contínuas para aprimorar a aplicação dos aspectos patrimoniais. A formalização de normas claras e a utilização de tecnologia adequada são elementos cruciais para a consecução de um controle efetivo dos bens móveis, contribuindo de forma expressiva para a eficiência da gestão pública como um todo.

### **2.8.2 Gestão de Estoque**

O presente tópico tem como foco discorrer sobre a importância da gestão de estoque nas organizações. Tal abordagem é necessária para que as empresas possam otimizar seus recursos, reduzir custos e atender melhor às demandas dos clientes. Segundo Teixeira (2023), uma gestão eficiente do estoque é crucial para evitar

excessos ou faltas de produtos, que podem impactar diretamente a lucratividade e a satisfação do cliente.

Ainda no que se está relacionado à gestão de estoque, para Scott (2021) a normalização dos procedimentos de controle é essencial para garantir a conformidade com as normas fiscais e regulatórias. Fica claro, portanto, que a falta de um gerenciamento adequado pode resultar em sanções legais e prejuízos financeiros. Tomando por base o objetivo do presente trabalho, que trata das melhores práticas na gestão de estoques, é possível notar que a adoção de tecnologias e metodologias de controle pode trazer vantagens competitivas significativas.

A gestão de estoque, que é um componente importante da gestão patrimonial, envolve o controle dos recursos físicos da empresa. A eficiência nessa área pode trazer os seguintes benefícios:

- a) Redução de Custos: Minimiza os custos de armazenagem e evita a falta ou excesso de produtos;
- b) Melhoria na Tomada de Decisão: Acesso a dados precisos sobre níveis de estoque que auxiliam no planejamento;
- c) Aumento da Produtividade: Processos automatizados ajudam a reduzir o tempo gasto na gestão de estoques;
- d) Otimização do Espaço: Melhora na organização física do estoque para melhor utilização do espaço disponível;
- e) Controle de Ativos: Monitoramento constante dos ativos físicos para evitar perdas e desperdícios;
- f) Planejamento Eficiente: A gestão de estoque permite um planejamento mais eficaz das compras e vendas.

Desse modo, as principais informações a serem tratadas aqui passam por técnicas de previsão de demanda, categorização de produtos, e a utilização de software de gestão. Com base na argumentação apresentada ao longo deste tópico, fica claro que a gestão de estoque não é apenas uma questão operacional, mas um elemento crítico para a estratégia de negócios. E ficam, então, estabelecidas as bases da discussão sobre a implementação de sistemas avançados de controle de estoque, a ser desenvolvida no próximo capítulo.

### 3 DESENVOLVIMENTO DO ESTUDO DE CASO

Neste trabalho, será detalhado um estudo de caso focado no desenvolvimento de uma aplicação, explorando as soluções atuais para as necessidades abordadas em capítulos anteriores, bem como apresentando e embasando cientificamente soluções alternativas ou aprimoradas. Para isso, serão apresentados os seguintes tópicos: o cenário encontrado, a metodologia de coleta e análise de dados, um relatório dos resultados obtidos, as arquiteturas da solução proposta e, por fim, o detalhamento do banco de dados.

#### 3.1 CENÁRIO ENCONTRADO

Em muitas secretarias estaduais, a maior parte dos processos de gestão de estoque ainda é predominantemente manual, baseada em planilhas eletrônicas ou, em alguns casos, registros em papel. Essa abordagem tradicional está suscetível a uma série de erros humanos, incluindo a duplicação de registros, perda acidental de dados e demora para respostas específicas de superiores. Além disso, a atualização desses registros demanda tempo e esforço consideráveis, o que frequentemente resulta em dados desatualizados. Com isso, a ausência de um sistema melhor automatizado não só compromete a eficiência operacional, como também retarda a reposição de materiais essenciais, gerando um impacto negativo na capacidade da secretaria de responder rapidamente a demandas emergentes e prever necessidades futuras de inventário com precisão.

A integração limitada com outros sistemas administrativos e financeiros, principalmente devido à ausência de aplicação das formas nominais nos bancos de dados, que é muito defendida em Oliveira (2010), é outro problema crítico enfrentado por muitos órgãos estaduais. A falta de conexão com outros departamentos resulta em uma coordenação ineficaz, aumentando a complexidade do gerenciamento de informações e a probabilidade de inconsistências nos dados, gerando custos operacionais extras elevados, pois os recursos são frequentemente desperdiçados em tentativas de resolver problemas de comunicação e processamento de dados.

Situações como as anteriores mencionadas foram encontradas pela equipe de desenvolvimento na secretaria estadual em questão, a qual, o setor do patrimônio do mesmo, vide acomodação e relutância de buscar alternativas anteriormente, realiza

tarefas arcaicas, como fazer vistorias anuais nos outros setores para saber se os bens duráveis estão na posse do lugar certo em horários grandes, como um setor por dia, que facilmente poderia ser otimizado com aplicações, as quais, se conectadas à internet, podem gerar relatórios mais precisos para os gestores.

Na secretaria de estudo, observou-se um predomínio de práticas arcaicas no setor de patrimônio, resultado de uma tendência à acomodação e relutância em adotar novas alternativas. As atividades de inspeção anual, realizadas para assegurar a correta alocação de bens duráveis entre os setores, são conduzidas de maneira ineficiente, onde um setor é visitado por dia. Esse método consome tempo significativo e poderia ser facilmente aprimorado através da adoção de aplicações conectadas à Internet, que possibilitariam a geração de relatórios mais precisos aos gestores.

Conseqüentemente, há uma receptividade crescente entre gestores e secretários, tanto da esfera administrativa quanto financeira, para a implementação de ferramentas tecnológicas modernas.

Tais ferramentas têm o potencial de melhorar significativamente a capacidade de geração de relatórios e análises detalhadas, essenciais para a realização de avaliações estratégicas mais eficazes. Essa modernização não apenas otimiza os processos existentes, mas também promove o desenvolvimento de soluções proativas e inovadoras, enfrentando de forma mais eficiente os desafios associados à gestão de estoque.

### **3.1.1 Coleta e Análise de Dados**

A coleta e análise de dados no contexto encontrado revelou várias deficiências críticas ao sistema atual. Um dos problemas mais comuns relatados entre os entrevistados é a ocorrência frequente de erros de registro ou perda de dados. Ao decorrer dos relatos, itens são muitas vezes fisicamente localizados em setores diferentes daqueles indicados nos registros, uma situação que se agrava devido à ineficiência das vistorias anuais, que são lentas e não realizadas com a frequência necessária para garantir a precisão das informações. Este processo demorado contribuiu significativamente para a desatualização dos dados, impactando negativamente a gestão eficiente de estoque, como afirmou a própria equipe do setor.

Outra reclamação é sobre o tempo necessário para realizar uma vistoria completa. Atualmente, a vistoria de estoque é feita a um ritmo de um setor por dia, o

que significa que leva um mês inteiro para completar uma revisão geral. Este método não apenas consome muito tempo, mas, como foi relatado, também retarda a capacidade de resposta do setor de patrimônio às demandas dos gestores (gestão patrimonial). Além disso, a falta de visibilidade em tempo real complica ainda mais as operações diárias. Quando informações sobre o estoque são solicitadas, é necessário gerar relatórios que não estão segmentados por setor, mas sim por órgão, o que exige uma organização posterior e prolonga o tempo de resposta.

Adicionalmente, a ausência de integração com outros sistemas administrativos ou financeiros dificulta a possibilidade de o setor inter-relacionar os dados já existentes com outras ferramentas, uma vez que não existe um fluxo de dados eficiente entre as diferentes áreas. No geral, foi constatado que o sistema atual é amplamente considerado pelos gestores como o pior entre os sistemas governamentais, devido à sua falta de interatividade, dificuldade de organização e incapacidade de fornecer informações de forma clara e detalhada. Esta insatisfação dos entrevistados destaca a urgência de reformular o sistema para atender melhor às necessidades dos usuários e melhorar a eficiência operacional da administração pública matrimonial.

A implementação de uma nova aplicação web é vista como uma solução promissora para superar esses desafios. Espera-se que a nova plataforma inclua funcionalidades avançadas, como a anexação de fotos aos itens e uma melhor organização dos dados, além de fornecer informações detalhadas em tempo real e quantidades por setor. A secretaria demonstra entusiasmo em relação às novas soluções tecnológicas, investindo em reuniões de alinhamento com os setores de desenvolvimento para identificar problemas e desenvolver soluções eficazes. Essa abordagem proativa visa não apenas resolver as falhas atuais, mas também promover uma gestão de estoque mais eficiente e integrada.

### **3.1.2 Relatório dos Resultados**

A realização das visitas de campo revelou diversas inconformidades nos princípios básicos das 5 formas normais de armazenamento de dados em tabelas, como mostrado na Figura 6, juntamente com uma resistência significativa quanto à disponibilização dessas informações através do sistema de gerenciamento atual. Os servidores do almoxarifado expressaram ceticismo sobre a necessidade de novos

sistemas, acreditando que as soluções existentes seriam suficientes para suas operações diárias.

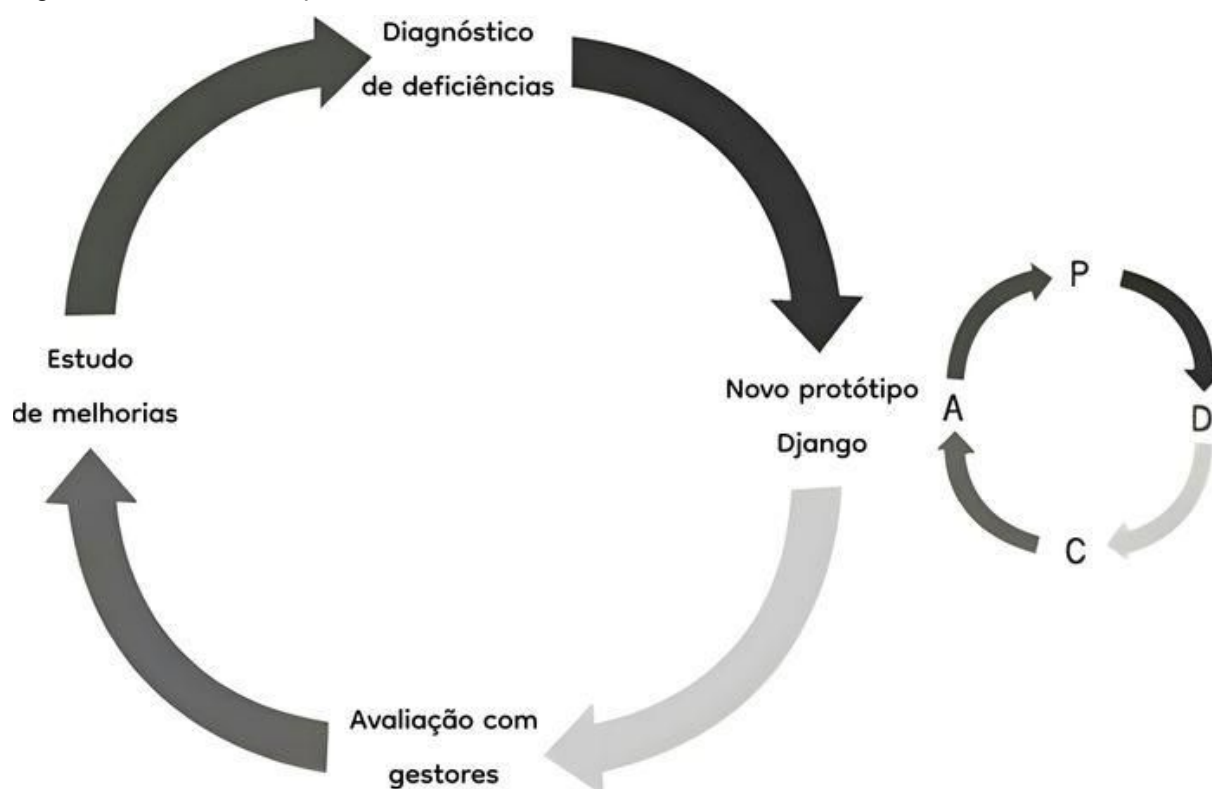
Contrapondo essa visão, gestores de alto escalão destacaram as deficiências do sistema atual, apontando-o como insuficiente em termos de informações necessárias para análises de estoque e decisões de compras. A contribuição desses foi essencial na avaliação da qualidade do sistema iPatri, onde, ao se apresentar as primeiras telas de protótipos desenvolvidas no Figma, as lideranças dialogaram com os servidores inicialmente resistentes, esclarecendo dúvidas e aliviando receios de que novos sistemas pudessem resultar em sobrecarga de trabalho ou impactos negativos em suas rotinas. Essa interação mostrou-se crucial para promover a aceitação e apoio à modernização e integração de novas tecnologias na secretaria.

### 3.2 ARQUITETURA DA SOLUÇÃO

O desenvolvimento do sistema iPatri foi guiado pela metodologia *Lean*, buscando aprimorar a gestão dos bens públicos de forma eficiente e transparente. Inspirados pela visão de entrega rápida focada no cliente, conforme mencionado por Kulshrestha (2019), e com o objetivo de otimizar o todo, foi desenvolvido o primeiro protótipo do iPatri para solucionar os problemas encontrados na gestão do patrimônio. A metodologia ágil adotada permitiu a construção de um protótipo funcional em um curto período, possibilitando ajustes e melhorias contínuas.

O ciclo *Plan-Do-Check-Act*, expressado na figura 7, seguiu uma estrutura bem definida, iniciando-se pela fase de **Planejar**, na qual foram identificadas as deficiências do sistema de gestão patrimonial existente que impactavam negativamente a tomada de decisões. Este diagnóstico serviu como base para a concepção do protótipo do iPatri, com a participação ativa de gestores e servidores para promover a aceitação da mudança e alinhar expectativas. Na etapa de **Fazer**, o primeiro protótipo do sistema foi desenvolvido utilizando o framework Django, com foco na entrega rápida e eficiente para atender às necessidades práticas da gestão de bens públicos.

Figura 5 — Ciclo Lean Aplicado



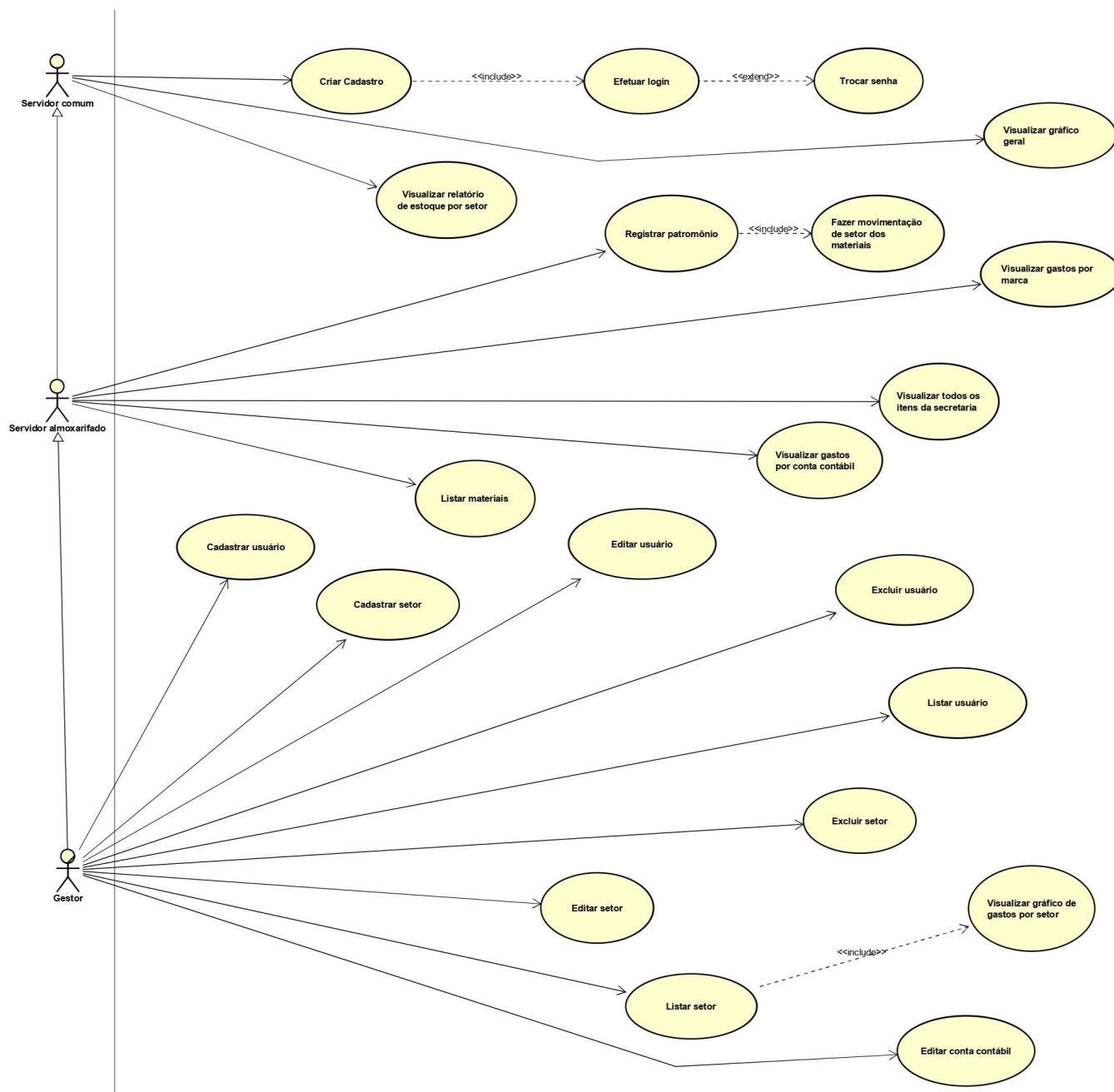
Fonte: O autor (2025).

A fase de **Verificar** consistiu na apresentação do protótipo aos gestores e usuários, que puderam avaliar a eficácia do sistema e fornecer feedback valioso sobre a clareza e utilidade das informações. Essa etapa possibilitou identificar os pontos de melhoria e realizar os ajustes necessários. Por fim, a fase de **Agir** compreendeu a documentação das lições aprendidas ao longo do processo e o planejamento de futuras iterações do sistema, visando aprimorá-lo continuamente e adaptá-lo às demandas do setor público, inclusive com a exploração de tecnologias como a inteligência artificial. A arquitetura do sistema foi planejada para suportar a futura implementação de padrões de projeto, o que ampliará ainda mais sua robustez e flexibilidade, garantindo que a solução possa evoluir e se integrar a futuras necessidades e requisitos tecnológicos.

O iPatri destacou-se por sua eficiência e agilidade, principalmente através da remoção de código desnecessário e otimização do desempenho. Logo no início do projeto, foi criado um diagrama de Caso de Uso, exposto na Figura 8, para se localizar as necessidades dos clientes, o qual auxiliou na identificação e eliminação de funcionalidades que não agregavam valor significativo aos usuários, permitindo uma interface mais limpa e objetiva. A simplificação de lógicas complexas ocorreu através

da refatoração, resultando em um código mais legível e fácil de manter, além de minimizar erros potenciais. Essa abordagem não só melhorou a eficiência do desenvolvimento, mas também garantiu que a equipe pudesse focar em funcionalidades realmente relevantes para os usuários.

Figura 6 — Diagrama Caso de Uso Aplicado



Fonte: O autor (2025).

A solução proposta visa não apenas desenvolver, mas também demonstrar as vantagens de se ter o Django como *framework* principal, com o objetivo de promover

uma gestão eficiente e transparente dos ativos públicos. O iPatri foi desenvolvido para melhor reunir as informações das três importantes áreas voltadas aos bens públicos: o setor de patrimônio, os gestores e os servidores que utilizam os bens. A aplicação web desenvolvida foi segmentada em nichos onde cada um pode ter controle e melhor gestão dos bens de seu setor, ou, dependendo do nível de usuário, de toda a secretaria.

No desenvolvimento do iPatri, a priorização da organização do código e da reutilização de componentes é crucial para assegurar a escalabilidade, manutenibilidade e clareza da solução. Nesse cenário, a implementação de padrões de projeto, como o Factory, presente na figura 7, surgiu como uma estratégia eficaz para abstrair e centralizar a criação de objetos complexos com diversas vantagens, como o promover a separação de responsabilidades, o que, por sua vez, aprimora a legibilidade e simplifica a manutenção do código.

Figura 7 — Factory aplicado

```
factories > relatorio_factory.py > RelatorioFactory > _gerar_pdf
1 import xlwt
2 from reportlab.platypus import SimpleDocTemplate, Paragraph, Table, TableStyle
3 from reportlab.lib.pagesizes import A4
4 from reportlab.lib import colors
5 from reportlab.lib.styles import getSampleStyleSheet
6 from django.http import HttpResponse
7 from usuario.models import Usuario
8 from patrimonio.models import Item
9
10 class RelatorioFactory:
11     @staticmethod
12     def gerar_relatorio(usuario_email: str, formato: str):
13         try:
14             usuario = Usuario.objects.get(email=usuario_email)
15             setor = usuario.setor_id_setor
16             itens = Item.objects.filter(setor_id_setor=setor)
17
18             if formato == 'xls':
19                 return RelatorioFactory._gerar_xls(itens)
20             elif formato == 'pdf':
21                 return RelatorioFactory._gerar_pdf(itens, setor.setor_abrev, usuario.
22                                                     nomeusuario)
23             else:
24                 raise ValueError("Formato não suportado.")
25         except Usuario.DoesNotExist:
26             return None
```

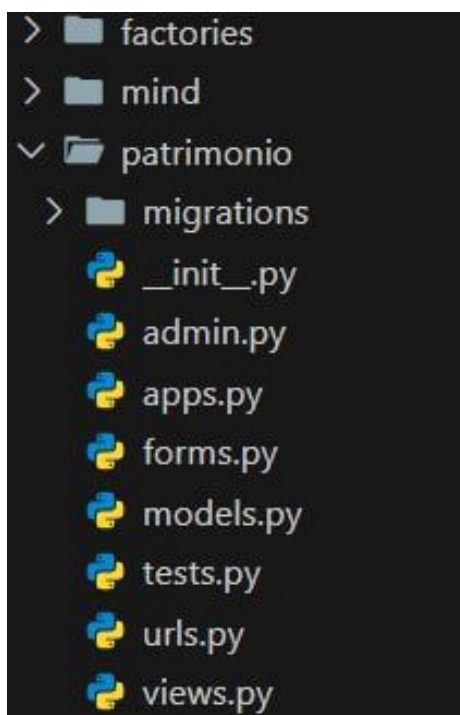
Fonte: O autor (2025)

O padrão Factory, no contexto da aplicação utilizada no estudo de caso, favoreceu a reutilização de métodos de geração de relatórios, permitindo que sejam

aplicados em diferentes contextos da aplicação, sejam eles agendamentos automáticos, integrações externas ou outras views. Por fim, a centralização da lógica de formatação e exportação de dados contribui significativamente para a padronização da saída dos relatórios, independentemente do ponto da aplicação que demande essa funcionalidade. Desse modo, a aplicação do padrão de projeto Factory, a qual se está presente na no processo de geração de relatórios, se configura como uma solução elegante e eficiente, alinhada aos princípios do desenvolvimento orientado a objetos e às boas práticas de arquitetura de software em ambientes Django.

Já a Programação Orientada a Objetos ofereceu uma série de vantagens para o iPatri, incluindo modularidade, organização do código em classes e objetos, reutilização de componentes, facilidade de testes, flexibilidade e manutenibilidade do sistema, além de possibilitar a aplicação de diversos padrões de projeto como o anteriormente mencionado. A própria arquitetura do Django, fundamentada no padrão Model-Template-View, alinha-se intrinsecamente com os princípios da POO. O MVT, com arquivos expostos na figura 8, ao promover uma estrutura clara e organizada no desenvolvimento de aplicações web, destaca-se pela separação de responsabilidades, otimizando a manutenção e a escalabilidade do código.

Figura 8 — Parte da Árvore de Arquivos da aplicação



Fonte: O autor (2025)

A divisão da aplicação em Model, Template e View permite, por exemplo, legibilidade do código, tornando-o mais acessível caso ocorra de ter mais membros na equipe de desenvolvimento. Os *Models*, que representam a estrutura de dados (definidos como classes em *models.py*), permitem a criação de modelos complexos com herança, relacionamentos e métodos personalizados. As *Views* (presentes em *views.py*), que encapsulam a lógica de negócios, podem ser implementadas como classes para melhor organização e reutilização. Os Templates, responsáveis pela apresentação, beneficiam-se da POO ao usar classes e objetos para representar elementos da interface do usuário.

Essa sinergia do MVT com a POO, resultou em um código mais modular e reutilizável, facilitando a implementação de novas funcionalidades sem a necessidade de reescrever grandes partes do sistema. No contexto da modelagem de dados em Django, a aplicação da POO é fundamental para mitigar a duplicidade de código, dificuldades na manutenção e o aumento da complexidade lógica decorrentes de estruturas repetitivas entre tabelas.

No presente projeto, por exemplo, a análise das *models* existentes revelou que diversas classes compartilhavam campos comuns, como nome, sigla, telefone e descrição. Para solucionar essa repetição, foram criadas classes abstratas, presentes por exemplo na figura 9, reutilizáveis que atuam como superclasses. A *EntidadeOrganizacional* foi estabelecida para encapsular nome e sigla, enquanto *ComContato* foi criada para o campo telefone. Dessa forma, *models* como *Setor* passaram a herdar de ambas as superclasses, e *Orgao* de *EntidadeOrganizacional*, otimizando suas definições.

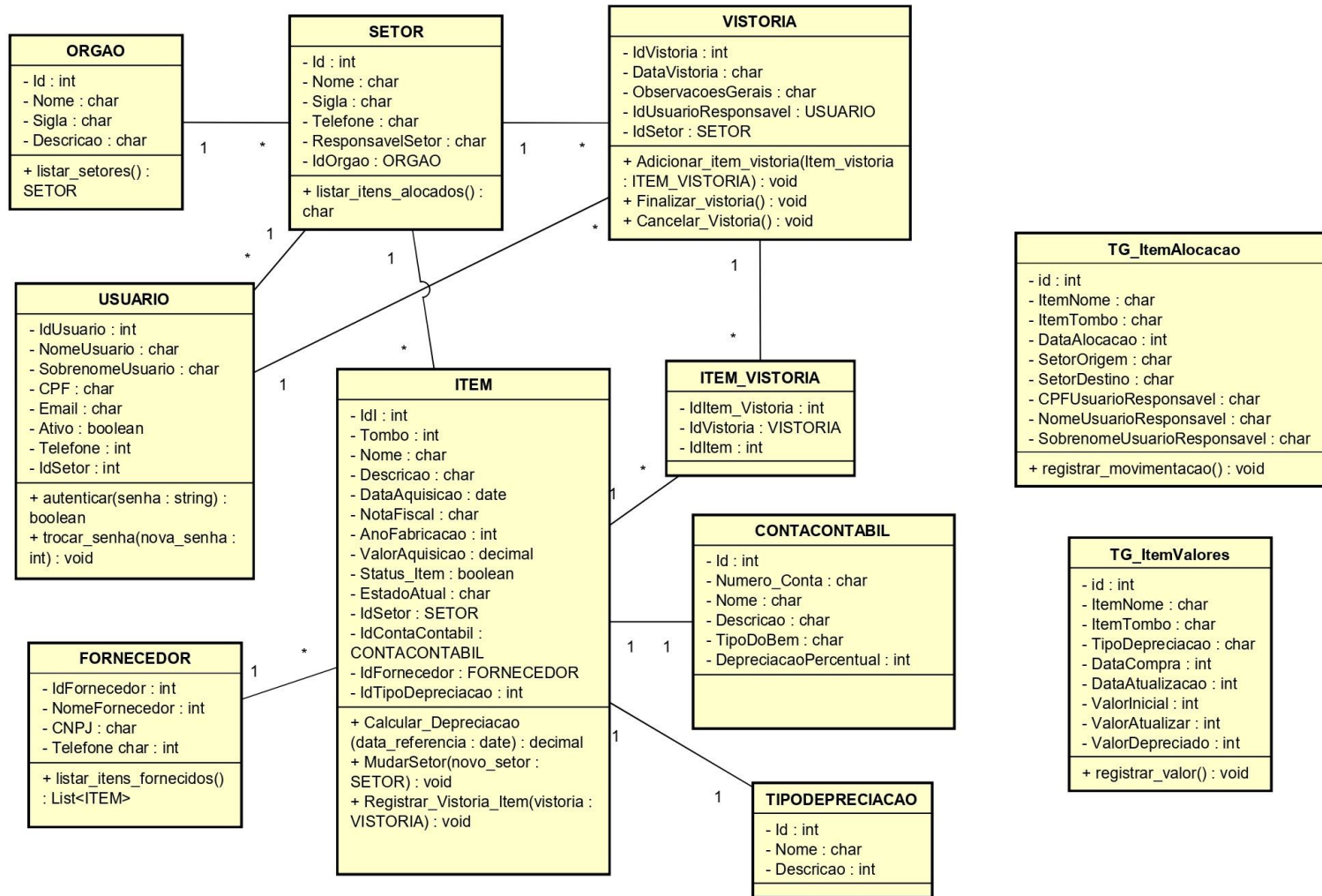
Figura 9 — POO aplicado no Django com Abstract e Herança

```
usuario > models.py > ...
1  from django.db import models
2
3  class EntidadeNomeada(models.Model):
4      nome = models.CharField(max_length=250, blank=True, null=True)
5      sigla = models.CharField(max_length=45)
6
7      class Meta:
8          abstract = True
9
10
11 class ComContato(models.Model):
12     telefone = models.CharField(max_length=45, blank=True, null=True)
13
14     class Meta:
15         abstract = True
16
17
18 class Orgao(EntidadeNomeada):
19     id = models.AutoField(primary_key=True)
20
21     class Meta:
22         managed = False
23         db_table = 'orgao'
24
25 class Setor(EntidadeNomeada, ComContato):
26     id = models.AutoField(primary_key=True)
27     responsavelsetor = models.CharField(max_length=45, blank=True, null=True)
28     idorgao = models.ForeignKey(Orgao, models.DO_NOTHING, db_column='IdOrgao', blank=True,
                                null=True)
```

Fonte: O autor (2025).

Essa reorganização hierárquica, onde todas as tabelas estão expressas no diagrama de classes da figura 10, proporciona uma série de vantagens significativas como eliminar a repetição de atributos, simplificar a manutenção, melhora a organização lógica, padroniza a nomenclatura e os tipos de campo, e aumenta a escalabilidade do sistema. Em suma, a flexibilidade e a extensibilidade proporcionadas pelo MVT e sua integração com a POO são cruciais para o sucesso de projetos complexos, reforçando sua posição como um padrão essencial no desenvolvimento de sistemas web como o iPatri.

Figura 10 — Diagrama de Classes Aplicado

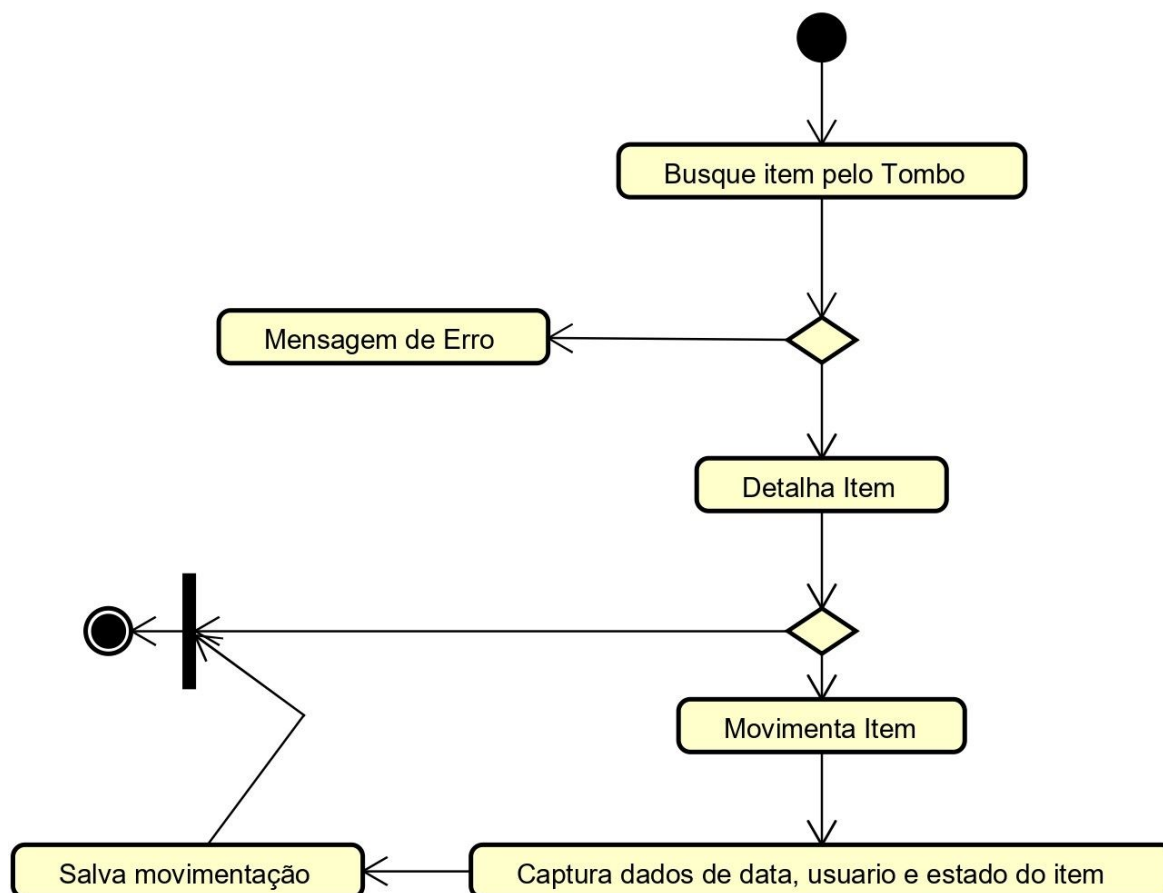


Fonte: O autor (2025)

Com isso, o ponto de partida para o projeto Django residiu na execução do comando `django-admin.py startproject core`. Essa ação estabelece uma estrutura de diretórios que engloba o diretório do projeto e arquivos de configuração cruciais, a saber, `settings.py`, que armazena as configurações do projeto, incluindo banco de dados e aplicativos instalados, `urls.py`, que define as rotas e *URLs* que o projeto responderá, `wsgi.py`, que configura a interface entre o servidor de aplicação e a aplicação Django, e `manage.py`, um utilitário para a execução de comandos de gerenciamento do projeto, tais como migrações e inicialização do servidor.

Subsequentemente, a criação de *apps*, componentes modulares do projeto, é realizada por meio do comando `django-admin.py startapp nomedoapp`, exemplificado por `django-admin.py startapp patrimonio`, o qual possui o seguinte diagrama de atividade, Figura 11, em uma de suas funcionalidades. Cada app pode conter funcionalidades distintas, permitindo a organização do projeto em partes independentes e reutilizáveis. A estrutura de arquivos padrão do mesmo compreende `models.py`, que define os modelos de dados para o banco de dados, `views.py`, que encapsula a lógica da aplicação e a interação com os modelos, `urls.py`, que define *URLs* específicas do app, `admin.py`, que configura o painel administrativo do Django para o app, `apps.py`, que contém a configuração do app, e o diretório `migrations/`, que armazena os arquivos de migração do banco de dados, colaborando para a construção da lógica completa do app, desde a definição dos dados até a apresentação na interface do usuário.

Figura 11 — Diagrama De Atividade



Fonte: O autor (2025).

Por fim, a documentação está sendo estruturada para fornecer informações detalhadas e organizadas sobre a aplicação, facilitando o entendimento e a manutenção do sistema por parte da equipe de desenvolvimento e dos gestores das secretarias. A combinação de uma arquitetura bem definida com uma documentação abrangente assegura que o impacto do uso de Python com Django para controle patrimonial seja amplamente compreendido e aproveitado. Dessa forma, a solução se posiciona como uma ferramenta estratégica para melhorar a eficiência e a transparência na gestão dos principais ativos públicos.

### 3.3 BANCO DE DADOS NO IPATRI

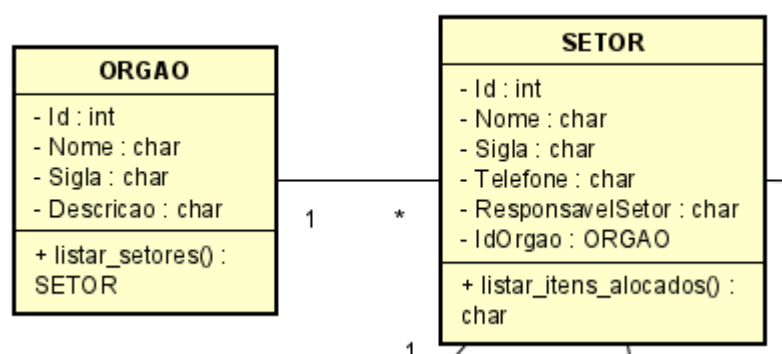
O iPatri utiliza o MySQL para estruturar seu banco de dados, priorizando a eficiência e funcionalidade por meio de um modelo entidade-relacionamento bem planejado. Tal organização facilitou a utilização de princípios de Programação

Orientada a Objetos, como herança e abstração, favorecendo a aplicação de atributos com funcionalidades semelhantes. A concepção da estrutura do banco de dados da aplicação web iPatri foi cuidadosamente elaborada para atender às complexas necessidades da gestão patrimonial em órgãos públicos estaduais. O banco de dados relacional, denominado db\_ipatri, constitui o alicerce sobre o qual toda a lógica da aplicação se apoia, garantindo a integridade e a consistência das informações.

A estrutura do banco teve como início com a criação da tabela ORGAO, como pilar hierárquico, armazenando os órgãos públicos aos quais os setores estão subordinados. Cada registro nesta tabela é identificado por um Id (chave primária), acompanhado de um Nome e uma Sigla. A Sigla, por ser um campo obrigatório, padroniza a identificação abreviada do órgão, contribuindo para a clareza e a uniformidade dos dados.

Dando sequência, a Tabela SETOR, mostrada na figura 11, está diretamente relacionada à tabela ORGAO, representando as divisões administrativas internas. Seus campos incluem Nome, Sigla (também obrigatória), Telefone e ResponsavelSetor. A crucial ligação com a Tabela ORGAO é estabelecida pelo IdOrgao, uma chave estrangeira que formaliza a hierarquia entre órgãos e seus respectivos setores.

Figura 12 — Diagrama de classe focado em Orgao e Setor



Fonte: O autor (2025).

A Tabela USUARIO é fundamental para a gestão de acesso e responsabilidades, representando os usuários da aplicação. Seus principais campos são NomeUsuario, SobrenomeUsuario, CPF e Email (todos obrigatórios), além de Telefone e um indicador binário Ativo, que controla o acesso do usuário. A relação

com o setor ao qual o usuário está vinculado é feita através do IdSetor, uma chave estrangeira que assegura a integridade referencial. Ademais, a Tabela FORNECEDOR é responsável por armazenar os dados das empresas ou entidades que forneceram os bens, facilitando o rastreamento da origem dos bens e a conformidade com contratos públicos. Para gerenciar os diferentes modelos de depreciação, a Tabela TIPODEPRECIACAO foi criada, oferecendo flexibilidade na aplicação das políticas contábeis.

A Tabela ITEM, é a tabela central do banco de dados, representando cada bem patrimonial individualmente. Contém identificadores como Tombo (único) e IdItem, dados detalhados do bem e relacionamentos cruciais com outras tabelas. Essa estrutura permite um controle detalhado de localização, origem, valor atual e histórico de cada item, sendo o coração da gestão patrimonial. Outrossim, a Tabela VISTORIA é dedicada ao armazenamento dos registros de inspeções físicas e administrativas realizadas sobre os bens patrimoniais. Seus campos essenciais, como DataVistoria e IdUsuarioResponsavel, permitem a geração de relatórios e auditorias sobre o estado dos bens em um determinado setor e data. Já a tabela CONTACONTABIL é essencial para a classificação contábil dos bens, alinhando-se aos princípios da administração pública. Contém Numero\_Conta e DepreciacaoPercentual, que representa a taxa anual de depreciação associada à conta, permitindo cálculos automáticos de desvalorização dos bens ao longo do tempo. E estabelecendo um relacionamento muitos-para-muitos, a tabela ITEM\_VISTORIA atua como uma tabela de associação entre ITEM e VISTORIA.

A Tabela TG\_ItemValores, mostrada na Figura 11, auxilia no controle evolutivo dos valores dos bens, com campos como ValorInicial, ValorAtualizado e ValorDepreciado, sendo extremamente útil para auditorias financeiras e estudos de viabilidade de reposição ou alienação de bens.

Figura 13 — Tabelas para auditoria do banco

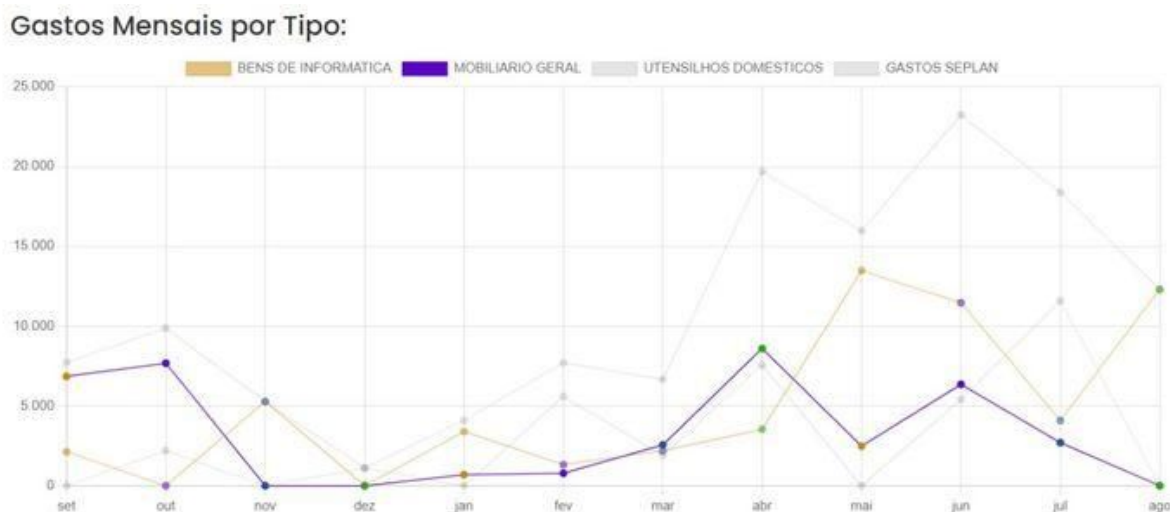
TG_ItemAlocacao	TG_ItemValores
<ul style="list-style-type: none"> <li>- id : int</li> <li>- ItemNome : char</li> <li>- ItemTombo : char</li> <li>- DataAlocacao : int</li> <li>- SetorOrigem : char</li> <li>- SetorDestino : char</li> <li>- CPFUsuarioResponsavel : char</li> <li>- NomeUsuarioResponsavel : char</li> <li>- SobrenomeUsuarioResponsavel : char</li> </ul>	<ul style="list-style-type: none"> <li>- id : int</li> <li>- ItemNome : char</li> <li>- ItemTombo : char</li> <li>- TipoDepreciacao : char</li> <li>- DataCompra : int</li> <li>- DataAtualizacao : int</li> <li>- ValorInicial : int</li> <li>- ValorAtualizar : int</li> <li>- ValorDepreciado : int</li> </ul>
+ registrar_movimentacao() : void	+ registrar_valor() : void

Fonte: O autor (2025).

E Para manter um registro histórico detalhado das movimentações de itens, a Tabela TG\_ItemAlocacao, também presente na figura 11, funciona como um log, registrando cada transação e fornecendo dados essenciais para rastreabilidade patrimonial e auditorias.

O desenvolvimento do sistema iPatri beneficiou-se consideravelmente da otimização das consultas ao banco de dados ao se utilizar o Django ORM de maneira eficiente. Além disso, quando necessário, os dados foram tratados com o MySQL Workbench para uma melhor organização e geração de tabelas. A indexação adequada dos campos frequentemente consultados foi essencial para melhorar a performance das consultas e a geração de gráficos de forma mais organizada, como o presente na Figura 12, garantindo respostas rápidas mesmo sob alta carga de trabalho.

Figura 14 — Gráfico do site mostrando gastos por tipo



Fonte: O autor (2025).

Paralelamente, ciclos curtos de feedback dos usuários foram implementados, permitindo ajustes rápidos e eficazes no desenvolvimento para validar ideias antes de avançar com o desenvolvimento total. Essa abordagem assegurou que as mudanças atendiam efetivamente às expectativas dos usuários e pudessem aumentar a satisfação geral com o produto.

Em suma, a modelagem do banco de dados foi orientada por princípios de normalização e integridade referencial, assegurando que os dados estejam bem organizados, acessíveis e seguros. O uso estratégico de chaves estrangeiras e índices únicos garante a coerência das informações, enquanto a integração com o framework Django permite uma administração robusta e amigável da base de dados. Essa estrutura é fundamental para garantir a transparência na gestão patrimonial pública, além de oferecer mecanismos eficientes de auditoria, rastreamento e valorização dos bens do Estado, culminando em uma gestão de recursos públicos mais eficaz e responsável.

## 4 RESULTADOS E DISCUSSÕES

A transparência e a responsabilidade na gestão dos recursos públicos são pilares fundamentais para a consolidação da confiança da sociedade nas instituições governamentais, permitindo que cidadãos e gestores acompanhem a execução orçamentária e avaliem a eficiência dos serviços. Nesse contexto, a administração pública estadual, com sua intrínseca ligação à gestão patrimonial e de estoque, é crucial para a otimização de recursos. Os resultados deste estudo de caso revelaram, através das entrevistas com supervisores, chefes setoriais e servidores públicos que atuam no controle de patrimônio, e das observações das dinâmicas de trabalho, que a falta de sistemas informatizados adequados é um desafio significativo que afeta a rastreabilidade e a eficácia na administração de bens públicos. As informações coletadas destacaram a urgência de modernização e a necessidade de tecnologias que garantam maior controle e transparência nos processos de gestão patrimonial.

Segundo Barros (2021), a contabilidade pública transcende a mera função de registro, atuando como um instrumento de controle e avaliação, facilitando a prestação de contas e a responsabilização dos gestores. A implementação de sistemas de informação eficientes, como o sistema web proposto, torna-se, portanto, crucial para a coleta e o gerenciamento de dados orçamentários e financeiros. A integração de dados, proporcionada por tais sistemas, permite que a administração pública tenha acesso a informações precisas e atualizadas sobre suas despesas, otimizando a tomada de decisões e alinhando-as às necessidades da sociedade.

A adoção de tecnologias da informação na contabilidade pública não apenas padroniza os registros contábeis, mas também possibilita uma melhor consolidação das contas nacionais. O Plano de Contas Aplicadas ao Setor Público (PCASP) é um exemplo de estrutura que visa atender às peculiaridades do setor, promovendo a transparência e a eficiência na gestão pública. Nesse contexto, a gestão patrimonial e a gestão de estoque assumem um papel central, pois a correta administração dos bens e materiais públicos impacta diretamente na eficiência dos serviços prestados e na responsabilidade fiscal.

A aplicação da metodologia *Lean*, observada nos resultados da pesquisa, reforça a importância da otimização dos processos na administração pública. A metodologia não apenas acelerou o desenvolvimento do sistema, mas também

melhorou significativamente a qualidade e a relevância das funcionalidades entregues. A capacidade de incorporar feedback contínuo dos usuários resultou em uma aplicação mais alinhada às necessidades do público-alvo, melhorando a satisfação do usuário final. A implementação de *dashboards* sincronizados às compras e visualizações de dados dinâmicas equipou os gestores com ferramentas poderosas para análise e decisão, permitindo respostas mais rápidas às mudanças no ambiente de negócios.

Em síntese, a administração pública estadual, ao integrar a gestão patrimonial e a gestão de estoque em um sistema de informação eficiente, impulsiona a transparência, a responsabilidade fiscal e a eficiência na prestação de serviços. A adoção de tecnologias inovadoras, como o sistema web proposto, e a aplicação de metodologias ágeis, como o *Lean*, são ferramentas indispensáveis para a construção de uma administração pública moderna, transparente e alinhada às demandas da sociedade.

#### 4.1 COMPARATIVO GERAL

A implementação do i-Patri representou um avanço significativo no rastreamento e controle dos bens públicos, garantindo maior transparência e eficiência na gestão patrimonial das secretarias estaduais. A digitalização dos registros eliminou falhas comuns em sistemas manuais, como a falta de atualização de inventários, a dificuldade na localização de itens e a imprecisão no controle de transferências entre setores. A aplicação possibilita o registro detalhado de cada bem patrimonial, incluindo número de tombamento, localização, histórico de movimentação e status de uso, reduzindo os riscos de extravio, deterioração indevida e uso inadequado dos recursos públicos. A automação facilita a identificação de bens ociosos ou subutilizados, permitindo uma redistribuição mais eficiente dentro da administração pública.

A aplicação trouxe maior transparência à gestão patrimonial, permitindo que auditorias sejam realizadas com mais eficiência e menor tempo de processamento. O sistema gera relatórios detalhados e personalizados, possibilitando que os gestores públicos tenham acesso instantâneo a informações essenciais sobre os bens registrados. Essa melhoria impacta diretamente a prestação de contas, facilitando o cumprimento das exigências dos órgãos de controle e fiscalização. A

integração de filtros avançados e relatórios automáticos otimiza a tomada de decisão e reduz os riscos de fraudes e inconsistências contábeis.

Figura 15 — Página de Visitas do iPatri

## Busca de Item por Tombo

Tombo no setor:

### Buscados:

Tombo	Nome do Item	Descrição	Setor do Bem
6011398	Roteador	Roteadores Wi-Fi de última geração, garantindo uma conexão de internet rápida e estável em toda a área do escritório, suportando múltiplos dispositivos simultaneamente.	SPLAN
8961339	Armario De Aço	Armários de aço resistentes, com fechamento seguro, ideais para o armazenamento de arquivos e materiais confidenciais em setores administrativos.	SPLAN
4059136	Estante	Estantes de aço robustas, ideais para armazenar documentos e equipamentos em escritórios. Possuem prateleiras ajustáveis para acomodar diferentes tamanhos de itens.	SPLAN

Fonte: O autor (2025).

Uma das principais vantagens do i-Patri é sua capacidade de escalabilidade, tornando-o uma solução viável tanto para pequenos órgãos estaduais quanto para grandes estruturas administrativas. A arquitetura baseada no framework Django permite que o sistema seja expandido conforme a demanda, suportando um aumento no volume de dados sem comprometer o desempenho. A flexibilidade do mesmo demonstra que ele é uma solução adaptável e sustentável para diferentes realidades da administração pública, promovendo gestão eficiente, segurança da informação e modernização contínua dos processos administrativos.

O estudo evidenciou a relevância da modernização da gestão patrimonial no setor público, demonstrando que a adoção de ferramentas digitais baseadas em tecnologias web é fundamental para garantir maior controle, transparência e eficiência na administração de bens públicos. A implementação do i-Patri, desenvolvido com Django e baseado no modelo MVT, permitiu não apenas a otimização dos processos administrativos, mas também a redução de desperdícios e falhas na rastreabilidade dos bens estatais. A aplicação prática do sistema nas secretarias estaduais demonstrou vantagens significativas sobre modelos

tradicionais, tanto em termos de usabilidade e escalabilidade quanto na capacidade de adaptação às necessidades específicas de cada órgão público. O uso de bancos de dados relacionais (MySQL) garantiu a integridade e segurança das informações, enquanto a abordagem fundamentada nas metodologias ágeis e Lean permitiu uma implementação flexível e ajustável de acordo com o retorno dos usuários.

A taxa de adoção e uso da aplicação foi outro indicador de sucesso, demonstrando a relevância e aceitação da ferramenta entre os usuários. A aplicação não só se integrou facilmente com sistemas existentes, mas também se beneficiou de uma organização de dados superior, reduzindo a redundância e melhorando a qualidade organizacional. Essa capacidade de integração facilitada garantiu que os dados pudessem fluir suavemente entre sistemas, um aspecto crucial para a colaboração eficiente entre diferentes departamentos e plataformas tecnológicas.

Um dos objetivos centrais do projeto foi diminuir o tempo de processamento de solicitações nos próximos seis meses, especialmente em processos de vistorias, o que poderia permitir a execução de duas ou mais inspeções por ano. Essa meta específica reforçou a eficiência operacional, pois incentivou a equipe a focar em otimizações críticas, promovendo um ciclo contínuo de aprimoramento e iteração conforme preconizado pela abordagem Lean. A combinação do Django com a metodologia ágil em questão ofereceu um framework poderoso para o desenvolvimento de aplicações web que são não apenas eficazes e eficientes, mas também altamente adaptáveis às necessidades em evolução dos negócios. Essa abordagem não só atende às necessidades imediatas de desenvolvimento, mas também prepara o caminho para inovação contínua e sucesso a longo prazo.

Com foco no desenvolvimento futuro, a próxima entrega prevê uma refatoração que visa implementar padrões de design como *Builder*, *Flyweight*, e *Command*. Essa abordagem não apenas busca otimizar ainda mais a eficiência da aplicação, mas também aumentar sua flexibilidade e escalabilidade. Ao estruturar a aplicação com esses padrões, a equipe de desenvolvimento estará equipada para responder rapidamente às mudanças nas necessidades do usuário e às evoluções tecnológicas, garantindo que a aplicação permaneça relevante e eficaz a longo prazo.

Além de beneficiar os servidores responsáveis pelo controle patrimonial, o sistema contribui para a transparência pública, facilitando auditorias e prestação de contas, o que fortalece a governança e a responsabilidade fiscal do Estado. A

integração da tecnologia ao setor público se mostra cada vez mais necessária para enfrentar desafios administrativos, garantindo maior eficiência na gestão dos recursos públicos e promovendo um serviço mais acessível e confiável para a população.

Diante desses avanços, o estudo sugere algumas perspectivas para trabalhos futuros, como a expansão do i-Patri para outras esferas governamentais, a integração com API's de sistemas administrativos já utilizados pelo governo e o desenvolvimento de uma versão mobile que permita a gestão remota do patrimônio onde futuramente será registrado com um código de barras junto. A partir dessas melhorias, o sistema poderá ampliar seu impacto e consolidar-se como uma solução inovadora e eficaz na modernização da administração pública.

## 5 CONCLUSÃO

Em conclusão, este trabalho enfatiza a relevância da implementação do sistema iPatri, que busca aprimorar a gestão de bens públicos. Ao longo do desenvolvimento, foi evidente que as deficiências do sistema atual foram um ponto crítico que motivou a necessidade de uma solução mais robusta. A interação entre gestores e servidores, especialmente após o aplicativo, foi crucial para dissipar resistências e construir um ambiente favorável à adoção de novas tecnologias. Os *feedbacks* recebidos demonstraram que a aceitação do sistema está diretamente ligada à clareza e à utilidade das informações oferecidas, facilitando a tomada de decisões em tempo real.

A arquitetura da solução, fundamentada na metodologia *Lean*, visa não apenas a eficiência operacional, mas também a flexibilidade necessária para se adaptar às mudanças nas demandas do setor público. A utilização de padrões de design como *Factory* e MVT, otimizam ainda mais a aplicação, garantindo que ela não apenas atenda às necessidades atuais, mas também seja capaz de evoluir conforme as tecnologias emergentes. Tal foco no design e na estruturação dos dados é essencial para assegurar a relevância do sistema em um cenário em constante mudança.

Os resultados obtidos até o momento demonstram que a integração do *Django* com a metodologia ágil *Lean* proporciona uma base sólida para o desenvolvimento de aplicações *web* que sejam não apenas eficazes, mas também adaptáveis. A ênfase na organização de dados e na capacidade de integração futura é um dos principais pilares que sustentam a proposta do iPatri. Essa abordagem não apenas atende às exigências imediatas de gestão dos bens patrimoniais, mas também se alinha com as expectativas de um futuro digital, onde a interoperabilidade e a transparência são fundamentais.

Além disso, este trabalho destaca a importância da documentação e do treinamento contínuo para os usuários do sistema. A preparação adequada da equipe, aliada a um suporte constante, é vital para garantir que a aplicação seja utilizada de forma eficiente. A comunicação aberta entre os diversos *stakeholders*, incluindo gestores e usuários finais, é essencial para fomentar um ambiente colaborativo que valorize a inovação e a melhoria contínua.

Por fim, a conclusão deste estudo não se limita a apresentar os resultados alcançados, mas também propõe direções para trabalhos futuros. A continuidade das pesquisas sobre a eficácia do sistema e a exploração de novas tecnologias, como inteligência artificial e análise de dados, pode ampliar ainda mais as capacidades do iPatri. A expectativa é que, com essas melhorias, o sistema se torne uma referência na gestão de patrimônio público, contribuindo para uma administração mais eficiente e transparente.

## REFERÊNCIAS

- ALCHIN, Marty. **Pro Django**. Apress, v. 3, f. 156, 2009. 312 p.
- ALURA. **O que é Python**. 2024. Disponível em: <https://www.alura.com.br/apostila-python-orientacao-a-objetos/o-que-e-python#python>. Acesso em: 4 ago. 2024.
- BARROS, Rosaura Haddad ; MOTA, Francisco Glauber Lima; AMORIM, Mírian Patrícia. **Contabilidade aplicada ao setor público**. Universidade Aberta do Brasil, v. 3, 2021.
- BOOCH, Grady; RUMBAUGH, James; JACOBSON, Ivar. **UML: Guia do Usuário**. 2012.
- CARVALHO, Vinícius. **MySQL: Comece com o principal banco de dados open source do mercado**. Editora Casa do Código, 2022.
- CRUZ, Felipe. **Python: Escreva seus primeiros programas**. Editora Casa do Código, 2022.
- DENNIS, Pascal. **Produção Lean Simplificada**. Bookman Editora, 2008.
- DOWNEY, Allen. **Pense em Python**. 6 ed. Novatec Editora, 2010.
- ELMASRI, Ramez; NAVATHE, Shamkant B.. **Sistemas de banco de dados**, f. 362. 2006. 724 p.
- FOWLER, Martin. **Patterns of Enterprise Application Architecture**. Addison-Wesley, 2002.
- FREITAS, Marcio. **UML Fundamentos**. DevMedia. 2008. Disponível em: <https://www.devmedia.com.br/uml-fundamentos/8640>. Acesso em: 2 ago. 2024.
- GAMMA, Erich *et al.* **Padrões de Projetos: Soluções Reutilizáveis de Software Orientado a Objeto**. Porto Alegre: Bookman, 2008.
- GIL, Antônio Carlos . **Métodos e técnicas de pesquisa social**. 6 ed. São Paulo: Ed Atlas, 2008.
- GOMES, André Farias. **Lean Software Development: Java Magazine 81**. DevMedia. 2010. Disponível em: <https://www.devmedia.com.br/lean-software-development-java-magazine-81/17442>. Acesso em: 4 mar. 2024.
- GUEDES, Gilleanes T. A.. **UML 2: Guia Prático** . 3 ed. São Paulo: Novatec Editora, 2018.
- KULSHRESTHA, Saurabh. **What Is Agile Methodology: Know the What and How?**. Medium. Edureka, 2019. Disponível em: <https://medium.com/edureka/what-is-agile-methodology-fe8ad9f0da2f>. Acesso em: 4 mai. 2024.
- LABAKI, Josué; WOISKI, Emanuel. **Introdução A Python: Python Orientado a Objetos**. Ilha Solteira: Grupo Python. 2016.

LUCIDCHART. **O que é diagrama de atividades UML?**. Disponível em: <https://www.lucidchart.com/pages/pt/o-que-e-diagrama-de-atividades-uml>. Acesso em: 4 ago. 2024.

LUCIDCHART. **O que é um diagrama de classe UML?**. Lucidchart. Disponível em: <https://www.lucidchart.com/pages/pt/o-que-e-diagrama-de-classe-uml>. Acesso em: 4 ago. 2024.

MACHADO, Luis Fernando Pires. **Patrimônio Público**. Brasília: Grupo NT, 2013.

MYSQL. **MySQL 8.4 Reference Manual: Including MySQL NDB Cluster 8.4**. 2025. Disponível em: <https://dev.mysql.com/doc/refman/8.4/en/>. Acesso em: 10 mar. 2025.

OLIVEIRA, Celso Henrique Poderoso de. **SQL: Curso Prático**. 6 ed. São Paulo: Novatec, 2010. 272 p.

PENDER, Tom. **UML Bible**. John Wiley & Sons, f. 494, 2003. 988 p.

RIBEIRO, Leandro. **O que é UML e Diagramas de Caso de Uso: Introdução Prática à UML**. DevMedia. 2021. Disponível em: <https://www.devmedia.com.br/o-que-e-uml-e-diagramas-de-caso-de-uso-introducao-pratica-a-uml/23408>. Acesso em: 10 set. 2024.

SAMPIERI, Roberto Hernández; COLLADO, Carlos Fernandez; LUCIO, María del Pilar Baptista. **Metodologia de Pesquisa**. 5 ed. Porto Alegre: Penso, 2013.

SCOTT, Maria Eliane. **Análise dos aspectos patrimoniais dos bens móveis na prefeitura municipal de Gravataí**. Repositório Institucional da UFSC, 2021 Trabalho de Conclusão de Curso (Graduação em Administração Pública) - Universidade Federal de Santa Catarina, Praia Grande - SC, 2021.

SILVA, Ana Carolina Andrei; GRACIANO, Fernando. Desenvolvimento Agil de Software. **Recima 21**, 12 2022. Revista Científica Multidisciplinar.

TEIXEIRA, William Sebastião. **Desenvolvimento de um sistema para o controle de estoque de uma Instituição de Segurança Pública do Estado de Santa Catarina**. 2023 Trabalho de Conclusão de Curso (Graduação em Tecnologias da Informação e Comunicação) - Universidade Federal de Santa Catarina, Araranguá, 2023.

VICO, Renato de. **Um sistema de controle de estoque de uma ONG baseado no CRM da Salesforce**, f. 57 Trabalho de Conclusão de Curso (Graduação em Ciência da Computação) - Universidade Federal Fluminense, Niterói, 2021.

YAHYA, Mochammad Agus. **Implementing Design Patterns in Django: Best Practices for Efficient Web Development**. Medium. 2023. Disponível em: <https://medium.com/@mochammadagusyahya/implementing-design-patterns-in-django-best-practices-for-efficient-web-development-694b23bc6b7a>. Acesso em: 4 set. 2024.