UNIVERSIDADE FEDERAL DOS VALES DO JEQUITINHONHA E MUCURI
Sistemas de Informação
Gustavo Vaz Fernandes

SIAC: SISTEMA COLABORATIVO COM APOIO À DEMOCRACIA ELETRÔNICA PARA SUPORTE DE ATIVIDADES COMPLEMENTARES

Diamantina
2025

Gustavo Vaz Fernandes

SIAC: SISTEMA COLABORATIVO COM APOIO À DEMOCRACIA ELETRÔNICA PARA SUPORTE DE ATIVIDADES COMPLEMENTARES

Trabalho de conclusão de curso apresentado ao curso de Sistemas de Informação como parte dos requisitos exigidos para a obtenção do título de Bacharel em Sistemas de Informação da Universidade Federal dos Vales do Jequitinhonha e Mucuri - UFVJM

Orientadora: Prof.ª Drª Claudia Beatriz Berti

Diamantina
2025

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DOS VALES DO JEQUITINHONHA E MUCURI

FOLHA DE APROVAÇÃO

Gustavo Vaz Fernandes

SIAC: SISTEMA COLABORATIVO COM APOIO À DEMOCRACIA ELETRÔNICA PARA SUPORTE DE ATIVIDADES COMPLEMENTARES

Trabalho de Conclusão de Curso apresentado ao Curso de Sistemas de Informação da Universidade Federal dos Vales do Jequitinhonha e Mucuri, como requisitos parcial para conclusão do curso.

Orientadora: Prof.ª Dr.ª Claudia Beatriz Berti

Aprovado em 5 de Dezembro de 2025

BANCA EXAMINADORA

Profª. Drª. Claudia Beatriz Berti
Faculdade de Ciências Exatas - DECOM - UFVJM

Profª. Drª. Cinthya Rocha Tameirão
Faculdade de Ciências Exatas - DECOM - UFVJM

Profª. Drª. Luciana Pereira de Assis
Faculdade de Ciências Exatas - DECOM - UFVJM

Documento assinado eletronicamente por **Claudia Beatriz Berti**, **Servidor(a)**, em 11/12/2025, às 15:08, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Cinthya Rocha Tameirão**, **Servidor(a)**, em 12/12/2025, às 20:23, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Luciana Pereira de Assis**, **Servidor(a)**, em 12/12/2025, às 22:17, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

A autenticidade deste documento pode ser conferida no site https://sei.ufvjm.edu.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **1976529** e o código CRC **EB5807FA**.

Referência: Processo nº 23086.000729/2023-83 SEI nº 1976529

AGRADECIMENTOS

Agradeço a Deus, por toda benção e graças que me proporciona até hoje.

À minha família, registro meus sinceros agradecimentos pelo apoio incondicional recebido ao longo de toda a minha trajetória acadêmica, o qual foi essencial para a realização e conclusão da graduação em uma instituição federal de ensino superior.

À minha namorada Luiza, registro meus sinceros agradecimentos pelo incansável apoio, estímulo, carinho, compreensão e amor demonstrados durante toda esta caminhada. Sua contribuição foi determinante para a superação das dificuldades encontradas ao longo do desenvolvimento deste trabalho.

Agradeço a todos os amigos que me acompanharam durante minha jornada acadêmica, pelo apoio constante, pelas conexões construídas e pelos momentos compartilhados que tornaram esta caminhada mais leve e significativa.

Agradeço a todos os professores do qual pude receber os conhecimentos passados por eles.

A minha orientadora, Dra. Claudia Berti, expresso minha profunda gratidão pelos valiosos conhecimentos compartilhados, pela inspiração constante e pela paciência dedicada ao longo da orientação deste trabalho.

RESUMO

Este trabalho desenvolve e valida o Sistema de Informação para Atividades Complementares (SIAC) como uma solução web transparente, participativa e auditável. Diante das limitações do processo institucional de gestão das Atividades Complementares (AC) na Universidade Federal dos Vales do Jequitinhonha e Mucuri (UFVJM), marcado por formulários, e-mails e verificações manuais que geravam atrasos, retrabalho, baixa rastreabilidade e inconsistências na conversão de horas. O objetivo é reorganizar o fluxo de submissão, análise e validação das AC, padronizando critérios institucionais e reduzindo custos operacionais. A metodologia adotou abordagem ágil com Python/Django, modelagem de entidades centrais, controle de acesso por papéis e automações via padrão Observer para criação de perfis, registro de transições de status e notificações, além da validação de requisitos funcionais e não funcionais. Os resultados demonstram centralização do fluxo, redução de comunicação redundante, diminuição de erros processuais por validações automáticas, organização da fila de avaliação com notificações proativas e trilhas de auditoria imutáveis, com tempo de resposta adequado em ambiente de desenvolvimento.

Palavras-chave: Sistema Colaborativo. Democrâcia Eletrônica. Python. Desenvolvimento.

ABSTRACT

This work develops and validates the Sistema Integrado de Atividades Complementares (SIAC) as a transparent, participatory, and auditable web solution. Given the limitations of the institutional process for managing Complementary Activities at the Universidade Federal dos Vales do Jequitinhonha e Mucuri (UFVJM), characterized by forms, emails, and manual verifications that generated delays, rework, low traceability, and inconsistencies in the conversion of hours. The objective is to reorganize the flow of submission, analysis, and validation of CA, standardizing institutional criteria and operational costs. The modern methodology employs an agile approach with Python/Django, central entity modeling, role-based access control, and automation via the Observer pattern for profile creation, recording of status transitions and notifications, as well as validation of functional and non-functional requirements. The results demonstrate centralization of the flow, reduction of redundant communication, reduction of procedural errors through automatic validations, organization of the evaluation queue with proactive notifications and immutable audit trails, with adequate response time in a development environment.

Keywords: Collaborative System. Electronic Democracy. Python. Development.

LISTA DE ABREVIATURAS E SIGLAS

AC	Atividades Complementares
DE	Democracia Eletrônica
MEC	Ministério da Educação
SC	Sistemas Colaborativos
UFVJM	Universidade Federal dos Vales do Jequitinhonha e Mucuri
SIAC	Sistema Integrado de Atividades Complementares
ORM	Object-Relational Mapping
MVT	Model-View-Template
LGPD	Lei Geral de Proteção de Dados

SUMÁRIO

1 INTRODUÇÃO

As AC constituem um componente curricular obrigatório nos cursos de graduação, conforme diretrizes do Ministério da Educação (MEC), e têm como finalidade ampliar a formação do estudante por meio de experiências que extrapolam o ambiente formal de sala de aula, abrangendo ensino, pesquisa, extensão e vivências profissionais (LIMA, 2024). Na Universidade Federal dos Vales do Jequitinhonha e Mucuri (UFVJM), o Projeto Pedagógico do Curso de Sistemas de Informação estabelece categorias, limites de carga horária e critérios de validação para essas atividades, cuja comprovação é requisito para a colação de grau.

Contudo, o processo vigente para submissão e validação de AC na UFVJM apresenta limitações significativas. Baseado em trocas de e-mails, formulários físicos e verificações manuais de certificados, o fluxo atual é marcado por morosidade, baixa rastreabilidade das decisões, assimetria de informação entre discentes e administração, e ausência de mecanismos estruturados de comunicação e auditoria. Esse cenário gera atrasos na homologação, retrabalho administrativo, dificuldade de acompanhamento por parte dos estudantes e risco de inconsistências na conversão de horas, comprometendo tanto a eficiência operacional quanto a transparência institucional.

Estudos indicam que a integração de Tecnologias de Informação e Comunicação a processos acadêmicos pode promover modernização, eficiência e transparência, especialmente quando fundamentada em princípios de Sistemas Colaborativos (SC) e Democracia Eletrônica (DE) (PIMENTEL; FUKS, 2012; GOMES, 2005). O Modelo 3C, Comunicação, Coordenação e Cooperação, oferece um arcabouço para estruturar fluxos colaborativos digitais (PIMENTEL; FUKS, 2012), enquanto os princípios de DE orientam a construção de processos participativos, transparentes e auditáveis (GOMES, 2005; MARQUES, 2008).

A proposta aqui descrita representa uma evolução do protótipo inicial desenvolvido por Lima (2024), que apresentava limitações práticas, concentrando-se apenas na interface visual e no design de funcionalidades. Essa evolução se baseia também em referências teóricas sólidas sobre sistemas colaborativos, conforme discutido por Pimentel e Fuks (2012), com ênfase no Modelo 3C de Comunicação, Coordenação e Cooperação.

Lima (2024) propôs um sistema que facilitasse o acompanhamento, a submissão e a validação de certificados e atividades, promovendo uma experiência de uso mais prática e acessível para os estudantes. No entanto, como se tratava de uma prototipação inicial, o trabalho anterior apresentava apenas uma interface visual e um esboço de funcionalidades, faltando uma implementação prática que pudesse ser utilizada diretamente pelos usuários finais (LIMA, 2024).

Neste contexto, o presente trabalho propoe uma evolução do projeto inicial, com a implementação do sistema colaborativo em uma aplicação web completa, desenvolvida em Python e Django. Essa transição da prototipação para o desenvolvimento completo envolve a criação de funcionalidades aprimoradas, garantindo que o sistema seja acessível, escalável e

adequado às necessidades operacionais da UFVJM. Além disso, adaptações foram realizadas no projeto, ajustando o sistema para que ele atenda de forma eficaz às necessidades de segurança, transparência e usabilidade dos usuários.

Com entregas quinzenais seguindo a metodologia ágil Scrum, o desenvolvimento será estruturado de forma incremental, permitindo ajustes contínuos e uma abordagem flexível que contribua para o aprimoramento do sistema com base no *feedback* dos usuários. O SIAC foi concebido e implementado em Python/Django com banco SQLite incorporando papéis e regras de autorização em cada fluxo. O sistema inclui dashboards de solicitações, upload/consulta de arquivos, motivos para indeferimento, horas convertidas no deferimento, e um módulo de contato com notificações.

1.1 Objetivo Geral

Desenvolver um sistema colaborativo com suporte à DE para o gerenciamento das AC, permitindo uma interação eficiente e transparente entre os alunos e a administração acadêmica do curso de Sistemas da Informação da UFVJM.

Objetivos específicos

Para alcançar o objetivo geral, foram definidos e realizados os seguintes objetivos específicos:

- Analisar e estruturar os requisitos funcionais e não funcionais do sistema, considerando as necessidades observadas no protótipo inicial e integrando novas funcionalidades e melhorias;
- Desenvolver a aplicação utilizando Python e Django, garantindo escalabilidade, segurança e facilidade de manutenção, incluindo mecanismos de controle de acesso, autenticação e autorização;
- Implementar uma arquitetura de dados que permita o registro e consulta das AC de forma organizada e transparente, garantindo a integridade e confidencialidade das informações;
- Aplicar princípios de DE, como transparência e participação cidadã, para assegurar que os estudantes tenham controle e visibilidade sobre o status de suas submissões e possam interagir com o sistema de forma ativa;

1.2 Justificativa

O gerenciamento das AC, conforme o regulamento da UFVJM, envolve atualmente um processo burocrático e moroso, como já destacado por Lima (2024). Estudantes e administradores dependem de trocas de e-mails, formulários físicos e assinaturas para aprovação e validação de certificados, o que gera frustração devido à falta de clareza e previsibilidade no processo.

Conforme Fuks (2012), SC podem reduzir barreiras comunicacionais e aumentar a eficiência de processos organizacionais. Esses sistemas são especialmente eficazes quando

projetados com base em princípios de DE, que promovem transparência e participação cidadã. Essa abordagem é crucial para modernizar o gerenciamento de AC, alinhando-se às demandas da UFVJM por processos mais ágeis e participativos.

A proposta deste projeto é desenvolver um sistema colaborativo, com suporte à DE, capaz de centralizar e otimizar o processo de gerenciamento das AC na UFVJM. Diferente da abordagem prototipada anteriormente, a presente aplicação será uma solução funcional, construída em Python e Django, com foco na escalabilidade, segurança e acessibilidade. O sistema permitirá que os estudantes submetam suas atividades, consultem o status das validações e interajam com a administração acadêmica de maneira prática e intuitiva, garantindo uma experiência de uso eficiente e transparente.

Com o uso de um sistema colaborativo, os benefícios se estenderão tanto aos alunos quanto aos setores administrativos, eliminando a necessidade de comunicação repetitiva e reduzindo a carga de trabalho manual dos servidores. A plataforma permitirá que os servidores acompanhem, validem e aprovem as solicitações em uma interface centralizada, aumentando a produtividade e minimizando a possibilidade de erros causados por informações desatualizadas. Além disso, o sistema proporcionará uma maior transparência no processo, com atualizações em tempo real e um histórico detalhado das atividades, o que incentiva uma participação mais ativa dos estudantes na gestão de suas AC.

A implementação deste projeto atende à demanda crescente por soluções acadêmicas mais acessíveis e modernas, que priorizam a experiência do usuário e a eficiência administrativa. Ao promover uma integração entre SC e os princípios de DE, o sistema proposto contribuirá para um ambiente acadêmico inclusivo, onde os alunos podem acompanhar suas solicitações de forma autônoma e os servidores podem desempenhar suas funções de maneira mais eficiente e organizada. Com isso, espera-se que o projeto impacte positivamente a comunidade acadêmica da UFVJM, transformando o fluxo de informações, promovendo a colaboração e facilitando a tomada de decisão dos estudantes quanto às AC, gerando um ambiente universitário mais dinâmico e funcional.

A solução endereça gargalos do processo atual: dispersão de informações, falta de previsibilidade e esforço manual de controle. Ao centralizar e digitalizar os fluxos, o SIAC:

- Reduz a burocracia e o retrabalho;
- Aumenta a transparência por meio de estados claros e histórico auditável;
- Amplia a participação discente com visibilidade e comunicação orientada a serviços;
- Oferece base de dados estruturada para gestão e tomada de decisão.

2 REFERENCIAL TEÓRICO

A compreensão dos fundamentos teóricos relacionados às AC, aos SC e aos princípios da DE é essencial para sustentar o desenvolvimento do sistema proposto neste trabalho. As AC representam um componente indispensável nos cursos de graduação, contribuindo para o crescimento acadêmico, profissional e pessoal do estudante, ao passo que os SC e a DE fornecem as bases tecnológicas e conceituais para aprimorar o gerenciamento dessas atividades, incorporando transparência, participação e responsabilidade.

Segundo Lima (2024), as AC têm como finalidade complementar a formação do discente por meio de experiências que extrapolam o ambiente da sala de aula, promovendo o contato com práticas de ensino, pesquisa, extensão e vivência profissional. A gestão eficiente dessas atividades, contudo, ainda enfrenta desafios em muitas instituições, em especial pela ausência de sistemas integrados que favoreçam a comunicação, a transparência e o acompanhamento participativo. Esses desafios se refletem em atrasos na validação de certificados, assimetria de informação entre estudantes e administração e baixa rastreabilidade de decisões.

Atividades Complementares

As AC desempenham um papel central no desenvolvimento acadêmico, permitindo que os estudantes alinhem suas experiências práticas com os objetivos pedagógicos do curso (LIMA, 2024). Na UFVJM, o Projeto Pedagógico do Curso de Sistemas de Informação define os tipos de AC permitidas, seus limites de carga horária por categoria e as regras de validação. O cumprimento dessas atividades é requisito para a colação de grau e exige que os alunos comprovem suas participações por meio de certificados, que são posteriormente analisados e aprovados pela administração acadêmica.

Do ponto de vista sistêmico, a gestão das AC demanda:

- Modelagem clara de categorias e critérios de aceitação;
- Processos de submissão, avaliação e decisão com estados explícitos;
- Mecanismos de comunicação e devolutiva aos estudantes;
- Trilhas de auditoria e evidências armazenadas de forma segura.

A adoção de princípios de SC e DE neste contexto favorece a integração dessas demandas. Em particular, comunicação transparente e coordenação eficaz reduzem ambiguidades na análise; já a participação eletrônica e o acesso ao histórico de decisões fortalecem a confiança no processo, ao permitir que estudantes acompanhem em tempo real o status de suas submissões e compreendam os fundamentos das decisões.

2.1 Sistemas Colaborativos

Os Sistemas Colaborativos formam um domínio de natureza interdisciplinar que integra soluções digitais, arranjos organizacionais e aspectos sociocomportamentais para apoiar

grupos na consecução de metas comuns com maior eficiência, transparência e alinhamento (PIMENTEL; FUKS, 2012). Inserem-se no panorama do Trabalho Cooperativo Assistido por Computador, campo que analisa de que modo o trabalho coletivo pode ser mediado e ampliado por meio de artefatos computacionais. Para Pimentel e Fuks (2012), projetar SC exige uma visão sociotécnica: não basta ofertar funcionalidades isoladas; é necessário considerar interdependências entre pessoas, tarefas, contextos e tecnologias.

O Modelo 3C constitui o eixo conceitual empregado por esses autores para organizar o pensamento sobre colaboração.

- Comunicação: abrange os canais e protocolos de troca informacional (mensagens, comentários, notificações, justificativas), diminuindo ambiguidades, acelerando a resolução de conflitos e subsidiando decisões;
- Coordenação: diz respeito ao arranjo de atividades, definição de papéis e responsabilidades, regras de transição entre estados, alocação de recursos e sincronização temporal; quando explícita, reduz retrabalho e aumenta a previsibilidade;
- Cooperação: refere-se à execução conjunta de tarefas para produzir artefatos compartilhados (documentos, registros, validações), dependendo de consciência situacional (awareness) sobre quem faz o quê, quando e com quais objetivos.

Santoro (2001) reforça que a modelagem de processos colaborativos deve considerar não apenas a tecnologia, mas também o contexto organizacional e as interdependências entre tarefas, o que justifica a necessidade de mapear requisitos das AC ao Modelo 3C. Borges e Pino (1999) destacam que mecanismos de awareness — como notificações e exibição de estados em tempo real — são essenciais para coordenação eficaz em ambientes assíncronos, reduzindo ambiguidades e acelerando a tomada de decisão (PIMENTEL, 2006). demonstram a viabilidade de implementar o Modelo 3C em sistemas reais através de tecnologias de componentes, evidenciando que funcionalidades colaborativas bem estruturadas promovem rastreabilidade, escalabilidade e melhoria contínua.

Embora distintas, essas dimensões se influenciam mutuamente: falhas comunicacionais tendem a prejudicar a coordenação; lacunas de coordenação impactam a cooperação. O Modelo 3C, assim, oferece um quadro para analisar, especificar e avaliar funcionalidades colaborativas em diferentes domínios — inclusive na gestão de Atividades Complementares (AC). Ao relacionar os requisitos do sistema de AC a cada dimensão, identificam-se lacunas e definem-se melhorias consistentes.

Pimentel e Fuks (2012) também ressaltam componentes que sustentam a interação colaborativa:

1. Consciência situacional: percepção atualizada do estado do grupo e dos artefatos (quem submeteu, quem avalia, prazos pendentes);
2. Memória da colaboração: guarda de registros (logs, versões, justificativas) que habilita auditoria e aprendizagem institucional;

3. Protocolos de interação: regras que disciplinam comportamentos, sequências de ação e critérios de aceitação ou rejeição;
4. Mediação tecnológica: seleção e configuração de ferramentas (interfaces, notificações, controle de acesso);
5. Artefatos compartilhados: documentos, formulários e evidências que estruturam o trabalho conjunto;
6. Espaço de colaboração: o ciberespaço projetado onde papéis e fluxos se concretizam.

Com esses elementos integrados, SC bem desenhados produzem: (a) menor assimetria de informação entre os agentes; (b) maior rastreabilidade das decisões; (c) escalabilidade operacional (mais submissões com controle); e (d) mecanismos de aprimoramento contínuo (feedback e métricas). No caso das AC, isso implica um sistema capaz de registrar cada etapa, exibir estados atualizados, permitir diálogo estruturado, armazenar evidências e justificativas além de minimizar atrasos por sinalização de pendências e prazos.

Outro aspecto relevante é o nível de acoplamento entre atividades colaborativas (tight vs. loose coupling). Santoro (2001) demonstra que processos de validação podem envolver acoplamento moderado: há dependência entre as ações (avaliador precisa da evidência do aluno), mas tarefas podem ocorrer de forma assíncrona (submissão hoje; avaliação em prazo futuro). Projetar funcionalidades que respeitem esse padrão — com filas, notificações e rastros temporais — reduz fricção e favorece eficiência.

Quanto à temporalidade e localização, a matriz tempo–espaço clássica do CSCW distingue quatro modalidades de colaboração (BORGES, 1999). O gerenciamento de AC encaixa-se predominantemente em colaboração distribuída e assíncrona (tempos distintos, lugares distintos), reforçando a necessidade de persistência robusta , interfaces consistentes independente de contexto e mecanismos de atualização incremental.

Para avaliar SC, sugerem-se métricas como tempo médio de execução de tarefas, taxa de retrabalho, número de interações até a resolução, satisfação dos usuários e completude dos registros. Incorporar essas medidas ao sistema de AC viabiliza gestão orientada por evidências e ajustes iterativos.

A integração desses componentes, conforme argumentam Santoro (2001) e Borges e Pino (1999), transforma sistemas colaborativos em plataformas capazes de reduzir assimetrias informacionais, otimizar fluxos de trabalho e apoiar decisões baseadas em evidências — características fundamentais para a gestão eficiente das AC.

2.2 Democracia Eletrônica

A DE diz respeito ao emprego sistemático das Tecnologias da Informação e Comunicação para ampliar participação, deliberação, transparência e responsabilização em processos decisórios. Ainda que Pimentel e Fuks (2012) concentrem sua obra na dimensão colaborativa, os pilares de comunicação, coordenação e memória apresentados pelos autores fornecem su-

porte direto à construção de práticas democráticas digitais: para funcionar, a DE precisa tornar critérios visíveis, orquestrar procedimentos de decisão e manter registros íntegros e auditáveis.

No contexto acadêmico, a DE desloca o estudante de um papel passivo — mero cumpridor de requisitos — para a posição de agente ativo na governança de processos formativos, como a homologação das AC. Nessa perspectiva, quatro eixos estruturam a aplicação democrática ao gerenciamento das AC:

- Transparência: disponibilização clara e contínua de normas, categorias, limites de carga horária e justificativas de decisões. A transparência reduz incertezas e previne percepções de arbitrariedade.
- Participação informada: criação de meios para o aluno apresentar evidências, acompanhar status, complementar informações e questionar indeferimentos com base em critérios públicos.
- Prestação de contas : registro permanente de quem aprovou, quando, com qual justificativa, permitindo auditorias internas e correções.
- Inclusão e equidade: mitigação de barreiras tecnológicas (interface acessível, linguagem clara, compatibilidade com dispositivos variados) para evitar exclusão de grupos menos familiarizados com ferramentas digitais.

Gomes (2005) argumenta que a democracia digital depende de três condições estruturais: acesso à informação, canais efetivos de participação e mecanismos de prestação de contas. Marques (2008) amplia essa discussão ao propor que a participação política online deve ser informada, ou seja, apoiada por transparência de critérios e justificativas claras — princípio aplicável ao gerenciamento das AC, no qual discentes devem compreender as bases das decisões para exercerem protagonismo acadêmico. Rothberg (2008) ressalta que a accountability digital exige registros auditáveis e acessíveis, favorecendo controle social e aprimoramento institucional. Silva (2005) classifica graus de participação em três níveis: informação unidirecional, consulta bidirecional e participação ativa; o SIAC situa-se no segundo e terceiro níveis ao permitir acompanhamento de status, envio de evidências, contestação de indeferimentos e diálogo estruturado.

A convergência entre SC e DE manifesta-se em múltiplas dimensões. A memória de colaboração (SANTORO, 2001; PIMENTEL, 2006). atua como mecanismo de accountability, registrando decisões de forma auditável; a coordenação (Santoro, 2001) garante isonomia processual, assegurando que todos os casos sigam o mesmo fluxo; e a comunicação (BORGES, 1999). funciona como vetor de participação, permitindo diálogo estruturado e devolutivas fundamentadas. Assim, funcionalidades colaborativas tornam-se facilitadoras democráticas quando:

- Publicam critérios e estados em tempo real;
- Permitem diálogo estruturado (ex.: campo para réplica do estudante em casos de indeferimento);
- Geram trilhas de auditoria (logs invioláveis de ações);
- Exibem indicadores agregados (tempo médio de análise, motivos recorrentes de indeferimento) que fomentam controle social interno e melhoria contínua.

Nesse sentido, recursos frequentemente associados a plataformas de DE — fóruns deliberativos, canais de sugestão, painéis de desempenho, mecanismos de consulta — podem ser adaptados à realidade das AC:

- Fórum/Canal de esclarecimento: reduz assimetria de informação e evita repetição de dúvidas.
- Painel de indicadores: provê visão sistêmica da eficiência e equidade do processo.
- Sistema de feedback estruturado: permite ao estudante compreender lacunas na documentação de sua atividade.
- Registro de histórico acessível: fortalece confiança e evita duplicação de submissões.

A legitimidade democrática pressupõe segurança e integridade: autenticação robusta para impedir uso indevido de credenciais; controle granular de permissões para que apenas perfis autorizados avaliem; proteção de dados sensíveis (criptografia em repouso e em trânsito); e mecanismos antifraude (detecção de duplicidades, verificação de formato de certificados). Embora Pimentel e Fuks (2012) não detalhem aspectos criptográficos, a exigência de memória confiável e rastreável implica salvaguardas contra adulteração e perda.

Os efeitos práticos esperados da incorporação de DE ao fluxo de AC incluem:

- Redução de tempo de resposta por coordenação racionalizada;
- Menor taxa de indeferimentos por causa de documentação incompleta;
- Incremento da satisfação estudantil;
- Melhoria na qualidade dos dados institucionais.

Persistem, contudo, desafios: risco de sobrecarga informacional, assimetria de habilidades digitais entre usuários, necessidade de atualização contínua dos critérios para evitar obsolescência. A mitigação demanda governança adaptativa sustentada por métricas e feedback, em sintonia com ciclos iterativos de aprimoramento descritos em abordagens ágeis e coerente com a lógica de ajuste progressivo inerente aos princípios colaborativos.

Em síntese, a DE aplicada às AC reposiciona o processo de validação de um rito administrativo opaco para um fluxo participativo e auditável. O arcabouço teórico de SC (SANTORO, 2001; BORGES, 1999; PIMENTEL, 2006), combinado aos princípios democráticos de Gomes (2005) e Marques (2008), demonstra que canais de comunicação adequados, mecanismos de coordenação explícitos e artefatos cooperativos formam a infraestrutura mínima de qualquer prática democrática digital efetiva no contexto educacional, garantindo transparência, participação informada e rastreabilidade — elementos centrais na proposta do SIAC.

2.3 Python

Python começou a ser idealizada no final dos anos 1980 por Guido van Rossum, então pesquisador no Centrum Wiskunde & Informatica (CWI), na Holanda. A experiência prévia de van Rossum com a linguagem ABC inspirou a busca por uma linguagem de script de sintaxe simples e consistente, mas que, ao mesmo tempo, fosse prática para interagir com serviços do sistema operacional distribuído Amoeba, na época um projeto de microkernel associado a An-

drew Tanenbaum. A inexistência, naquele momento, de uma linguagem que conciliasse essas características levou van Rossum a desenhar sua própria solução, corrigindo limitações percebidas no ABC e incorporando um sistema de módulos influenciado por ideias do Modula-3. Em 1991, esse esforço resultou na liberação pública da primeira versão de Python, lançando as bases de um ecossistema que cresceria rapidamente nos anos seguintes (ALURA, 2021; MIND BENDING, 2014).

A escolha do nome seguiu um espírito informal presente no CWI. Em vez de aludir à serpente, "Python" homenageia o programa humorístico britânico *Monty Python's Flying Circus*. A associação popular ao animal se consolidou mais tarde, entre outros motivos, pela iconografia de publicações amplamente difundidas sobre a linguagem, o que contribuiu para o imaginário visual do ecossistema (ALURA, 2021; MIND BENDING, 2014).

Python tornou-se reconhecida por combinar uma sintaxe concisa com tipagem dinâmica e suporte a múltiplos paradigmas de programação, incluindo estilos procedural, funcional e orientado a objetos. Essa combinação aumenta a expressividade e favorece ciclos ágeis de desenvolvimento, ao mesmo tempo em que mantém legibilidade e baixo custo de manutenção. A biblioteca padrão oferece um conjunto extenso de módulos "prontos para uso" — estruturas de dados de alto nível, ferramentas para datas e horários, suporte a testes, entre outros — e a comunidade disponibiliza uma variedade de frameworks e pacotes que ampliam as aplicações possíveis da linguagem (BORGES, 2014; PEREZ, 2020).

Do ponto de vista de execução, a implementação de referência (CPython) compila o código-fonte para um bytecode portável e, em seguida, o interpreta em uma máquina virtual. Esse processo de "compilar para bytecode e interpretar" confere portabilidade prática: o mesmo programa pode ser desenvolvido em um sistema e executado em outro, desde que haja um interpretador compatível instalado, reduzindo barreiras entre plataformas e simplificando a distribuição (BORGES, 2014). Em paralelo à CPython, o ecossistema conta com outras implementações (como PyPy, Jython e IronPython), que exploram diferentes estratégias de tempo de execução e integração com plataformas específicas, ampliando o alcance da linguagem.

A evolução técnica de Python passou por um marco importante com a linha 3.x, lançada em 2008 com alterações não retrocompatíveis destinadas a corrigir inconsistências históricas e a unificar decisões de design. A governança do projeto é realizada pela Python Software Foundation (PSF), organização sem fins lucrativos dedicada ao avanço técnico e comunitário do ecossistema. A linguagem é software livre sob licença permissiva da PSF, o que facilita sua adoção em ambientes acadêmicos e industriais (BORGES, 2014; SILVA; SILVA, 2019).

A maturidade do ecossistema Python se reflete na ampla oferta de ambientes de desenvolvimento e ferramentas que cobrem diferentes perfis de uso (PYTHON, 2021a; RAMOS, 2020). IDEs especializadas, como a edição comunitária do PyCharm, oferecem recursos integrados de análise estática, depuração, testes, integração com sistemas de controle de versão e suporte a frameworks web, o que acelera tarefas do ciclo de vida de software. Em um outro extremo, ambientes interativos como o Jupyter Notebook permitem combinar código executá-

vel, texto explicativo e visualizações em cadernos, sendo especialmente úteis em contextos de ensino, experimentação e ciência de dados.

Editores de código extensíveis — por exemplo, Visual Studio Code, Vim, Sublime Text e Atom — complementam esse cenário com leveza e grande variedade de extensões. Embora editores e IDEs possam convergir em funcionalidades por meio de extensões, convém manter a distinção conceitual: IDEs integram, de forma coesa, editor, depurador, gerenciadores de ambientes e ferramentas de build; editores priorizam a edição e ganham capacidades por plugins, adequando-se a fluxos de trabalho mais modulares. A escolha prática depende do perfil do projeto, da familiaridade da equipe e das necessidades específicas de integração e automação.

O reuso de componentes é um dos pilares que sustentam a produtividade em Python. Bibliotecas e pacotes encapsulam soluções consolidadas para problemas recorrentes — desde manipulação de dados, comunicação em rede e processamento de arquivos até construção de aplicações web — permitindo que equipes concentrem esforços na lógica específica do domínio, em vez de reimplementar funcionalidades comuns. Essa dinâmica é fortalecida por práticas de empacotamento e distribuição e por ferramentas de gerenciamento de dependências, o que encurta o tempo entre a ideação e a entrega de valor (SAYS, 2010; ZANETTE, 2017).

Em termos operacionais, ambientes virtuais permitem isolar dependências por projeto, prevenindo conflitos entre versões e favorecendo reprodutibilidade. Repositórios públicos e privados viabilizam o compartilhamento controlado de pacotes, e pipelines de integração contínua incorporam testes e verificações de qualidade ao fluxo de entrega. Em conjunto, esses elementos produzem um ecossistema que equilibra agilidade com governança técnica, assegurando que soluções possam evoluir com segurança, legibilidade e desempenho adequados ao contexto de uso.

3 TRABALHOS RELACIONADOS

O gerenciamento de AC tem sido objeto de estudo de diversos autores que buscam propor soluções tecnológicas para tornar o processo mais ágil, transparente e acessível. Entre os trabalhos mais relevantes nessa área, destaca-se o de Lima (2024), idealizador do projeto original que serviu de base para o desenvolvimento do sistema apresentado neste estudo.

Em sua pesquisa, Lima (2024) propôs um protótipo de sistema colaborativo com apoio à DE para o gerenciamento das AC do curso de Sistemas de Informação da UFVJM. O trabalho teve como foco o uso de princípios de SC e da DE — como colaboração, transparência, comunicação, coordenação e gestão da memória — a fim de promover maior participação do estudante no processo de acompanhamento das suas atividades. Apesar de o protótipo ter sido desenvolvido apenas em ambiente de prototipação (Figma), sua estrutura conceitual fundamentou o desenvolvimento prático e web realizado neste trabalho.

Além do estudo de Lima (2024), outras pesquisas também contribuíram para a compreensão e evolução de ferramentas voltadas à gestão de AC. Miguel e Franco (2018) apresentaram uma proposta de sistema web para a Universidade Tecnológica Federal do Paraná (UTFPR), em que o professor era responsável por toda a avaliação e registro das atividades dos alunos. Contudo, a aplicação não foi concluída nem disponibilizada publicamente, limitando seu impacto prático.

De forma semelhante, Southier e Dorneles (2022) desenvolveram uma plataforma web utilizando o framework Django, voltada ao gerenciamento das AC no Instituto Federal Farroupilha (IFFar). O sistema buscava contemplar as particularidades de cada curso, o que demandava um acompanhamento manual por parte dos alunos e uma verificação criteriosa pelos coordenadores. Apesar do código ter sido publicado no GitHub, os autores não mencionam a utilização efetiva da ferramenta na instituição.

Outra contribuição significativa é a de Silva e Vallim (2012), que analisaram o uso do ambiente virtual de aprendizagem Moodle como meio de apoio ao gerenciamento das AC nos cursos de Sistemas de Informação e Ciência da Computação da Universidade Presbiteriana Mackenzie. O estudo evidenciou benefícios relevantes, como a ampliação da interação entre alunos e professores, o incentivo à participação discente e o aprimoramento da transparência nas solicitações e aprovações das atividades.

Observa-se que, entre os trabalhos analisados, apenas o de Silva e Vallim (2012) apresentou resultados práticos baseados em dados reais, evidenciando a importância do uso de ferramentas digitais para otimizar o acompanhamento e a validação das AC. As demais propostas permaneceram em caráter conceitual ou experimental, sem registro de aplicação institucional consolidada.

Diferentemente dessas abordagens, a proposta de Lima (2024) avançou ao incorporar princípios de SC e DE, transformando o gerenciamento das AC em um processo participativo e transparente, no qual o estudante é protagonista da própria trajetória acadêmica. A partir

dessa idealização, o presente trabalho concretiza a proposta em um sistema web funcional e responsivo, que amplia o alcance do modelo original, integra recursos modernos de usabilidade e acessibilidade e possibilita a aplicação prática dos fundamentos apresentados por Lima (2024).

4 METODOLOGIA

Este projeto surge como uma resposta às necessidades identificadas no gerenciamento das AC na UFVJM. A proposta visa não apenas criar um protótipo, mas desenvolver um sistema robusto e colaborativo que atenda aos desafios enfrentados por estudantes e administradores no processo de solicitação, conversão de horas curriculares e preenchimento de formulários.

O desenvolvimento do sistema utilizou a metodologia ágil Scrum, que permite ajustes iterativos e entregas incrementais (SCHWABER K.; SUTHERLAND, 2017). Essa abordagem assegura que as funcionalidades atendam às demandas dos usuários, conforme identificado durante as sprints quinzenais.

As etapas principais incluem:

- Levantamento de Requisitos: Identificação das lacunas no protótipo inicial de Lima (2024) e coleta de novas demandas;
- Prototipação e Design: Criação de interfaces interativas utilizando os princípios discutidos por Pimentel e Fuks (2012), incluindo foco na transparência e colaboração;
- Desenvolvimento: Implementação do *backend* em Django, garantindo escalabilidade, segurança e organização eficiente de dados.

Além disso, o sistema integrará princípios fundamentais de colaboração, cooperação, comunicação e transparência, essenciais para promover a interação em diversos contextos de participação e DE. A segurança e a capacidade de armazenar informações de forma confiável são prioridades, assegurando que todas as operações sejam realizadas de maneira segura e organizada.

A execução deste projeto foi impulsionada pela necessidade de superar as dificuldades pessoais e institucionais enfrentadas no gerenciamento das AC, oferecendo uma solução dinâmica e eficaz que beneficie toda a comunidade acadêmica da UFVJM. Ao focar no desenvolvimento de um sistema completo e funcional, buscamos não apenas resolver problemas imediatos, mas também criar uma plataforma sustentável e adaptável para o futuro.

Etapas do Desenvolvimento

- Planejamento de Requisitos: Esta fase envolve uma análise minuciosa dos requisitos iniciais do protótipo, integrando novas funcionalidades que priorizam a segurança dos dados e a facilidade de uso para estudantes e administradores. O objetivo é assegurar que o sistema atenda plenamente às necessidades identificadas;
- Desenvolvimento de *Backend*: Utilizando o framework Django, o *backend* será construído com modelos de dados robustos, assegurando um armazenamento e gerenciamento eficaz das AC. A implementação garantirá operações seguras de submissão e validação, além de permitir que os dados sejam acessíveis e organizados de maneira eficiente;

- Desenvolvimento de Frontend: Focando na experiência do usuário, será desenvolvida uma interface responsiva que facilite a navegação intuitiva para todos os usuários. Elementos visuais serão cuidadosamente projetados para simplificar o acompanhamento de submissões e aprovações, com um design adaptável para otimizar a usabilidade em diversos dispositivos;
- Implementação de Funcionalidades Colaborativas: Serão criadas funcionalidades específicas para fomentar a comunicação e colaboração entre os usuários. Isso incluirá canais dedicados para que os estudantes possam resolver dúvidas, acompanhar o progresso de suas solicitações e verificar o status de suas AC;
- Integração de Segurança: A segurança do sistema será reforçada através da configuração de autenticação e autorização com controle de acesso baseado em permissões. Medidas como criptografia de dados sensíveis e registros de auditoria serão implementadas para garantir a integridade e confiabilidade do sistema;
- Documentação e Entrega Final: Será preparada uma documentação abrangente do sistema, incluindo manuais de uso detalhados para administradores e estudantes, além de documentação técnica para facilitar a manutenção futura. Este passo final garante que todos os aspectos do sistema estejam bem documentados e prontos para a entrega.

5 SIAC: SISTEMA INTEGRADO DE ATIVIDADES COMPLEMENTARES

O sistema desenvolvido recebeu o nome de **Sistema Integrado de Atividades Complementares (SIAC)** e foi utilizada a linguagem de programação Python no desenvolvimento, pelo fato dela ser referência em manipulação de dados e possuir uma sintaxe versátil, isso proporcionou um desenvolvimento mais prático, dinâmico e eficiente. Outro ponto importante que justifica a escolha do Python são as bibliotecas disponíveis para a linguagem, que facilitaram ainda mais todo o processo de desenvolvimento. A escolha favorece produtividade e legibilidade do código pela sintaxe concisa e pelo ecossistema de bibliotecas que simplificam tarefas de manipulação de dados e construção web, o que se mostra especialmente pertinente em cenários de prototipação e evolução incremental de serviços públicos digitais (PRATES R.; CASTRO, 2018).

O SIAC responde a um conjunto de limitações do processo vigente de gerenciamento das AC, em que a tramitação por e-mails e formulários pouco rastreáveis produz ineficiência, opacidade decisória e assimetria de informação entre discentes e a administração acadêmica. Com base em Python e no *framework* Django, aliando a arquitetura Model-View-Template (MVT) ao padrão de projeto Observer, a solução materializa um processo digital colaborativo, transparente e auditável, alinhado às dimensões de comunicação, coordenação e cooperação propostas pela literatura brasileira de SC (PIMENTEL; FUKS, 2012).

5.1 Requisitos Funcionais, Não Funcionais e Regras de Negócio

Os requisitos funcionais foram definidos a partir do mapeamento fiel do fluxo institucional de AC, em termos de escopo funcional, o sistema abrange a gestão de usuários e papéis (administrador, coodernador e discente) de modo a refletir atribuições reais e isolar responsabilidades. Contempla-se um catálogo institucional com categorias, atividades e níveis de participação, parametrizado conforme o regulamento e acompanhado das evidências documentais exigidas por tipo de atividade. O discente pode cadastrar certificados com metadados completos — título, descrição, justificativa, horas informadas — e anexar a documentação comprobatória; a submissão de um certificado constitui formalmente uma requisição de análise, registrando-se estado, responsável e carimbos temporais.

A avaliação pelo coordenador admite deferimento ou indeferimento, exigindo-se motivação explícita no segundo caso, o que por sua vez alimenta uma linha do tempo acessível ao discente e consolida um registro auditável da decisão. O cômputo de horas é realizado de forma automatizada com base na tríade categoria–atividade–nível de participação, respeitando rigorosamente os limites normativos por grupo; as horas excedentes não são computadas, mas permanecem registradas por razões de transparência e para subsidiar relatórios. O sistema oferece pesquisas e filtros sobre categorias, atividades, certificados e submissões, garantindo visibilidade do andamento dos processos e contribuindo para a previsibilidade de prazos. Para fomentar participação e reduzir latências, há um canal de dúvidas entre discente e secretaria, com

categorização temática, histórico e notificações; tal desenho operacionaliza o “C” de comunicação do modelo 3C (PIMENTEL; FUKS, 2012). Por fim, painéis operacionais e indicadores permitem acompanhar a eficiência do processo — por exemplo, o tempo médio de análise e a distribuição de decisões — enquanto a trilha de auditoria captura quem realizou cada ação, quando, em que objeto e sob quais dados, compondo uma base robusta de prestação de contas.

Os requisitos não funcionais foram estabelecidos como *guardrails* de qualidade, segurança e manutenibilidade. A escolha por Django em Python, com o padrão MVT e o Object-Relational Mapping (ORM) nativo, favorece uma separação consistente de responsabilidades, a produtividade do desenvolvimento e o controle seguro da persistência. Em segurança, implementaram-se autenticação robusta, autorização baseada em papéis, proteções contra ataques comuns (CSRF, injeção de SQL mitigada pelo ORM, XSS por meio de escaping e validação/sanitização de entradas), além de boas práticas de gestão de sessões. Em conformidade à Lei Geral de Proteção de Dados Pessoais (LGPD), a solução opera sob os princípios de minimização de dados, transparência de finalidades e garantia dos direitos do titular, prevendo políticas de retenção e mecanismos de auditoria que permitem reconstrução fiel de casos. A usabilidade e a acessibilidade foram consideradas desde a prototipação: a interface é responsiva, com rotulagem clara, feedbacks imediatos e aderência a recomendações de acessibilidade, reduzindo esforço cognitivo e risco de erro.

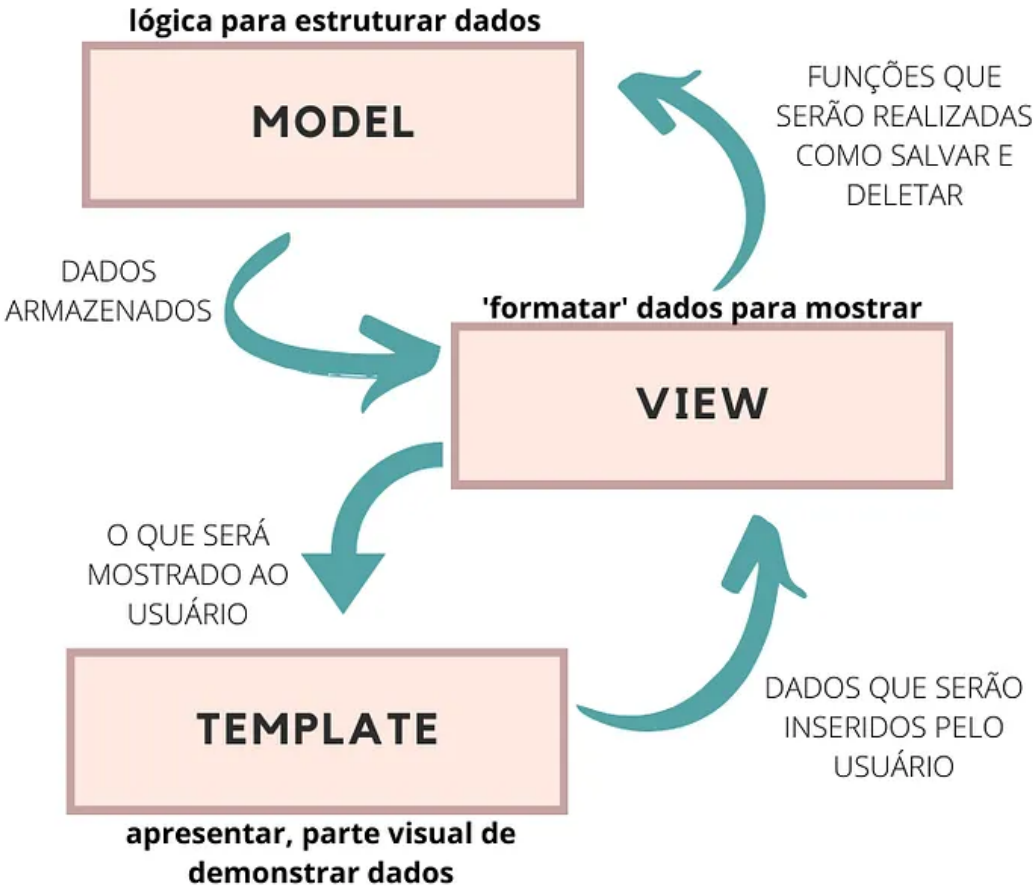
As preocupações de desempenho foram endereçadas com paginação sistemática, índices e consultas otimizadas e cache pontual para cenários de leitura intensiva. Para observabilidade, adotaram-se logs estruturados, correlação por requisição e métricas operacionais. Em manutenibilidade, o código segue convenções, emprega hints de tipo e ferramentas de lint/format, além de modularização por aplicativos que espelham subdomínios do problema; a conteinerização e scripts de infraestrutura pavimentam a implantação em diferentes ambientes com previsibilidade.

As regras de negócio transcrevem o regulamento institucional em termos computacionais, com ênfase na clareza do ciclo de vida e na integridade arquivística. Há limites máximos por grupo de AC e, quando estes são atingidos, as horas excedentes deixam de ser computadas, mas permanecem registradas para análise histórica e transparência. Cada submissão ancora-se em um certificado previamente cadastrado pelo discente, e percorre estados inequívocos — rascunho, em análise e decisão final de aprovação ou reprovação —, com exigência de motivação no indeferimento. O histórico de avaliações é imutável: alterações posteriores não sobrescrevem dados, e sim geram novos eventos que compõem a linha do tempo do caso. Prazos institucionais de submissão e de análise podem ser parametrizados em consonância com calendários acadêmicos e deliberações colegiadas, preservando a aderência ao contexto local.

5.2 Arquitetura e Tecnologias

Do ponto de vista arquitetural, o SIAC adota o padrão MVT do Django como matriz de separação de camadas, como podemos observar na Figura 1. Os modelos representam entidades de domínio como usuários e perfis, categorias e atividades, níveis de participação, certificados, submissões, mensagens de dúvidas e eventos de avaliação e de auditoria. As views, estruturadas em classes ou funções conforme a necessidade, orquestram a aplicação de regras de negócio, invocam serviços de conversão, validação, notificação e auditoria e compõem o contexto para a camada de apresentação. A renderização é realizada pela engine de templates, empregando herança de layouts, componentes reutilizáveis e convenções que evitam duplicação, favorecendo consistência visual e viabilidade de refatorações futuras. Essa arquitetura eleva a testabilidade, pois cada camada pode ser validada de forma direcionada, e reduz a superfície de acoplamento entre lógica de domínio e interface.

Figura 1 – Arquitetura MVT.



Fonte: ANDRADE, 2020

A organização por aplicativos reflete uma visão de subdomínios que reduz dependências transversais e isola responsabilidades. Um app de contas e usuários concentra autenticação, perfis e permissões; o app de catálogo assegura a integridade e a governança de categorias, atividades e níveis; o app de certificados trata dos metadados e da gestão segura de anexos; o app de submissões codifica o fluxo com estados, decisões, linha do tempo e relacionamento com eventos de avaliação; o app de comunicação sustenta as dúvidas e respectivas respostas, além de cuidar de notificações; módulos auxiliares contemplam armazenamento de estáticos e mídia, auditoria e painéis. A granularidade dessa segmentação foi escolhida para equilibrar coesão interna e baixo acoplamento entre módulos, o que se traduz em facilidade de evolução e em testes mais precisos. Tal estratégia sustenta a coordenação e cooperação entre atores, como descrito pelo modelo 3C (PIMENTEL; FUKS, 2012).

A modelagem de dados, orientada pelo ORM, foi construída para garantir integridade referencial, restrições de domínio e rastreabilidade. Usuários estendem o modelo de autenticação do Django com perfis e papéis que orientam o acesso a funcionalidades e a segmentação de dados. Categorias agregam atividades sob regras e limites que representam os grupos de AC; as atividades, por sua vez, definem documentação exigida e parâmetros de conversão. Níveis de participação vinculados às atividades influenciam a quantidade de horas a computar e eventuais tetos. Certificados se situam no cruzamento entre discente e catálogo e armazenam informações descritivas, justificativas, horas informadas e anexos, submetendo-se a validações semânticas e de segurança. Submissões capturam o estado do processo, os motivos de indeferimento, as horas convertidas e as marcações temporais; eventos de avaliação registram o ator responsável, a decisão e o contexto, satisfazendo exigências da prestação de contas. As mensagens de dúvidas documentam a interação entre discente e coordenador, com categorização temática, status de atendimento e histórico de trocas. Por fim, eventos de auditoria registram ações sensíveis com granularidade antes–depois, suportando investigações e relatórios.

A segurança foi tratada como requisito transversal, não um adendo. O sistema aplica proteção contra CSRF, se beneficia do ORM para neutralizar injeções de SQL e valida e sanitiza uploads, prática indispensável em sistemas que recebem documentos do usuário. Políticas de sessão preveem expiração e invalidação em alterações de credenciais, reduzindo superfícies de ataque. Do ponto de vista de produção, recomenda-se o emprego de cabeçalhos de segurança e de políticas de Content Security Policy para limitar vetores baseados em navegador. Em conformidade com a LGPD, o desenho prioriza o mínimo necessário de dados, a clareza quanto às finalidades e a possibilidade de atender a solicitações de acesso, retificação e eliminação, além de instituir políticas de retenção. A auditoria, enquanto componente de governança, foi planejada desde o início, com eventos agregados a linhas do tempo que, combinadas, reconstituem cada caso em sua evolução.

Uma decisão estrutural que merece destaque é a adoção do padrão de projeto Observer, implementado por meio das signals do Django. Em domínios processuais, cada transição relevante de estado desencadeia efeitos colaterais que não deveriam se confundir com a lógica

central de decisão. Ao aprovar uma submissão, por exemplo, é necessário notificar o discente, recalcular o painel de horas aprovadas, registrar o evento em auditoria e eventualmente atualizar métricas. Em vez de acoplar tudo isso a uma única view ou a métodos de modelo, optou-se por observar os eventos e acionar receptores que executam suas responsabilidades de forma idempotente e transacional. O uso de transaction garante que notificações e atualizações externas só ocorram após a persistência bem-sucedida, evitando inconsistências observáveis pelos usuários. Essa abordagem aumenta a manutenibilidade, pois novos observadores podem ser acrescentados sem alterar o fluxo nuclear, e fortalece a testabilidade, uma vez que cada receptor pode ser exercitado de forma isolada e verificável.

5.3 Design da Solução

A Figura 2 apresenta o diagrama de casos de uso do SIAC, evidenciando a separação clara de responsabilidades entre os três perfis de usuário (Discente, coordenador e Administrador) e as interações de cada papel com as funcionalidades do sistema.

Fluxos de Uso e Casos de Uso

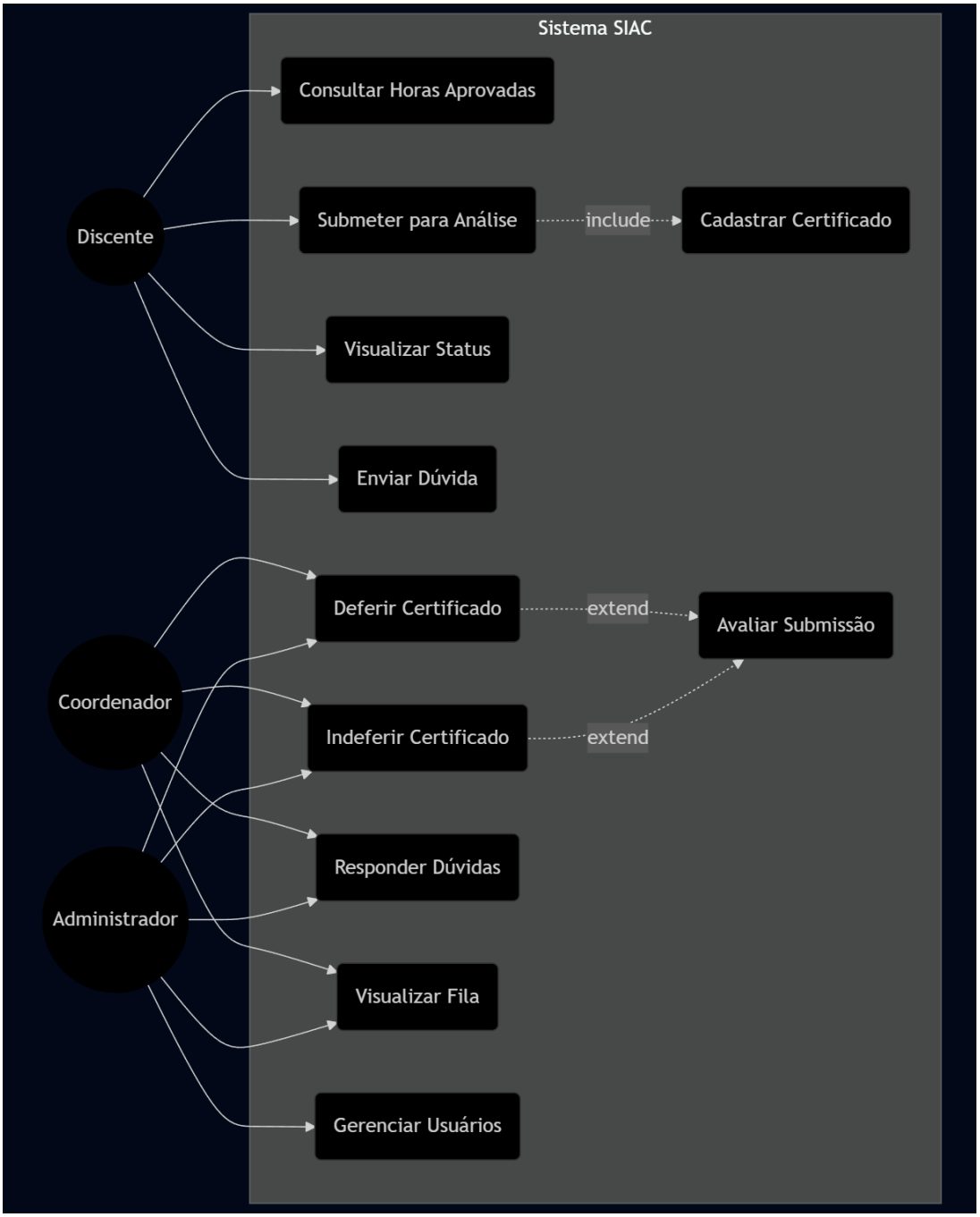
Para o discente, o percurso é claro: após o login, ele cadastra certificados com metadados e anexo, submete à análise e acompanha, por meio de uma linha do tempo, os eventos do seu processo — submissão, início de avaliação, decisão e eventuais solicitações de ajuste. O painel do discente exibe a evolução do cômputo de suas horas aprovadas, o que incentiva autonomia e planejamento. Quando há dúvidas, o canal de comunicação categorizado organiza a demanda e favorece respostas mais precisas e rápidas. Conforme ilustrado na Figura [X], o discente possui cinco casos de uso principais: Consultar Horas Aprovadas, Cadastrar Certificado (que inclui o relacionamento «include» com a funcionalidade de cadastro), Submeter para Análise, Visualizar Status e Enviar Dúvida.

Para o coordenador, a solução apresenta uma fila de submissões em aberto, com informações suficientes para priorização e decisão. O ato de avaliar implica deferir ou indeferir, com motivação obrigatória no segundo caso, e a decisão alimenta de imediato o histórico e as notificações. O coordenador acompanha ainda as solicitações concluídas e gerencia pendências de resposta no canal de dúvidas, o que permite um ciclo virtuoso de melhoria de serviço. Como demonstrado na Figura 2, o perfil coordenador interage com quatro casos de uso principais: Avaliar Submissão (que se estende através dos relacionamentos «extend» para Deferir Certificado e Indeferir Certificado), Responder Dúvidas, Visualizar Fila e Gerenciar Usuários.

O administrador mantém a integridade do catálogo — categorias, atividades e níveis —, gerencia perfis e acessos e acessa relatórios e indicadores que informam a saúde do processo. A Figura [X] evidencia que o Administrador possui três casos de uso específicos: Configurar Catálogo (que inclui o relacionamento «include» com Gerenciar Categorias), Visualizar Relatórios e Gerenciar Usuários (compartilhado com o perfil coordenador).

Assim, cada perfil encontra uma visão ajustada às suas responsabilidades, mas dentro de uma semântica compartilhada: estados, justificativas e históricos significam o mesmo para todos, o que acelera a adoção e reduz retrabalho. O diagrama de casos de uso ilustra claramente essa segregação de responsabilidades, garantindo que cada ator tenha acesso apenas às funcionalidades pertinentes ao seu papel, princípio fundamental do controle de acesso baseado em papéis (RBAC) implementado no sistema.

Figura 2 – Diagrama de Casos de Uso.



Fonte: Próprio autor.

Interface, Usabilidade e Acessibilidade

A interface foi planejada para ser previsível e inclusiva. A consistência visual e a herança de layouts evitam surpresas entre telas, enquanto a rotulagem descritiva e os feedbacks imediatos comunicam o resultado das ações do usuário. A responsividade assegura uso confortável em múltiplos formatos de tela, e cuidados de acessibilidade — contraste, foco visível, navegação por teclado e textos alternativos — ampliam o acesso e diminuem dependências de suporte. Em paralelo, escolhas de informação e de navegação buscaram minimizar carga cognitiva: os fluxos críticos foram afinados para reduzir cliques desnecessários, os filtros foram posicionados de modo a favorecer a descoberta e as listagens mantêm paginação e ordenação coerentes.

5.4 Implementação com Django e Python

A implementação materializa as decisões acima em uma estrutura de projeto organizada, com arquivos e diretórios característicos de projetos Django, como:

- O arquivo de gerenciamento orquestra comandos administrativos;
- Um banco SQLite atende ao desenvolvimento local, com recomendação de adoção de banco relacional robusto em produção;
- Diretórios de estáticos e de mídia segregam ativos e anexos;
- Os aplicativos concentram, cada um, responsabilidades específicas do domínio.

A configuração do projeto contempla separação por ambientes, o que facilita debugar comportamentos em desenvolvimento sem comprometer práticas de produção. As entidades centrais e suas regras foram implementadas com o ORM, permitindo declarações expressivas de relacionamentos e restrições. O catálogo, como fonte de verdade, amarra coerência entre documentação exigida e lógica de conversão, reduzindo inconsistências nos dados. O ciclo de vida das submissões, com transições atômicas e exigência de motivação em indeferimentos, provê um trilho comum a todos os casos, facilmente verificável por logs e por linhas do tempo.

A conversão de horas aplica uma regra geral que considera a função da atividade e o nível de participação sobre as horas informadas, impondo um teto em relação ao saldo do grupo daquele discente; A integridade desse cômputo é central para a credibilidade do sistema.

Boas práticas de código foram seguidas de ponta a ponta, seguindo um estilo e a padronização, os hints de tipo, linters e formatadores mantêm legibilidade e coerência; a camada de serviços ou casos de uso encapsula regras de negócio, deixando views e sinais mais enxutos e testáveis. Otimizações de consulta evitam problemas de desempenho por consultas N+1, enquanto a paginação consistente previne degradação em listagens extensas. Em termos de qualidade, recomenda-se uma pirâmide de testes que cubra desde funções puras de conversão de horas até ensaios ponta a ponta dos fluxos críticos, passando por testes de integração de RBAC e de auditoria.

O padrão Observer, via signals, fecha a orquestração reativa do domínio. Receptores cadastrados nos aplicativos monitoram eventos relevantes e disparam, apenas após o commit da transação, efeitos colaterais necessários: emissão de notificações, recálculo de métricas e registro de eventos de auditoria. Cada receptor mantém idempotência para evitar efeitos duplicados e é coberto por testes específicos, garantindo previsibilidade e facilidade de manutenção. Podemos observar um exemplo que se enquadra no cenario de submissão através da Figura 3 e o registro do signals na figura 4.

Figura 3 – Exemplo de observador para submissões.

```
# submissions/signals.py
from django.db import transaction
from django.db.models.signals import post_save
from django.dispatch import receiver

from .models import Submission
from audit.models import AuditEvent
from dashboard.services import recalc_student_hours
from communication.services import notify_submission_status

@receiver(post_save, sender=Submission)
def on_submission_saved(
sender,
instance: Submission,
created: bool, **kwargs):
    # Cria evento de auditoria sempre que a
    # submissão é criada/atualizada
    AuditEvent.log(
        entity="Submission",
        entity_id=instance.id,
        action="created" if created else "updated",
        actor_id=getattr(instance, "last_actor_id", None),
        metadata={"status": instance.status, "converted_hours":
        instance.converted_hours},
    )

    # Reage a transições para estados finais
    if instance.status in ("APPROVED", "REJECTED"):
        def _post_commit():
            # e-mail/alerta ao discente
            notify_submission_status(instance)
            # atualiza painel de horas
            recalc_student_hours(instance.student_id)

        transaction.on_commit(_post_commit)
```

Fonte: Próprio autor.

Figura 4 – Registro dos signals.

```
# submissions/apps.py
from django.apps import AppConfig

class SubmissionsConfig(AppConfig):
    name = "submissions"

    def ready(self) -> None:
        from . import signals
        # noqa: F401  (força o registro dos receivers)
```

Fonte: Próprio autor.

No uso cotidiano, o SIAC promove uma experiência coesa, ancorada em estados claros, justificativas transparentes e histórico confiável. O discente encontra autonomia para conduzir suas solicitações, acompanha sua evolução em tempo real e dispõe de um canal formal para sanar dúvidas. O coordenador opera com uma visão de fila que favorece priorização e decisão, sustentando tempos de resposta mensuráveis e consistentes. O administrador preserva o ordenamento institucional do catálogo e enxerga, por indicadores, gargalos ou oportunidades de ajuste no processo. Em termos de resultados, a centralização das informações e a padronização dos fluxos reduzem a necessidade de comunicação repetitiva e diminuem a incidência de erros processuais.

A transparência proporcionada por estados inequívocos, linhas do tempo e decisões motivadas eleva a confiança e a accountability e, como efeito indireto, reduz contestações e retrabalho. A aderência automática a limites normativos garante justiça e uniformidade no cômputo das horas, ao mesmo tempo em que o registro de excedentes preserva o contexto para análises posteriores. Por fim, a base arquitetural — MVT, modularização por apps e Observer — deixa o sistema preparado para evoluções naturais, como integrações a provedores de identidade, tramitação documental e validação automática de certificados por OCR e assinatura digital, bem como para ampliações de relatórios rumo a análises preditivas e painéis analíticos.

5.5 Banco de Dados

A persistência e organização dos dados no SIAC desempenham papel central na garantia de integridade, rastreabilidade e conformidade com os requisitos institucionais da UFVJM. A arquitetura de dados foi concebida para refletir fielmente o domínio do problema, assegurando que cada decisão de modelagem traduza uma regra de negócio ou um princípio de governança.

Para o SIAC, optou-se por SQLite como sistema gerenciador de banco de dados. Essa escolha se justifica pela simplicidade operacional, ausência de configuração de servidor e suficiência para atender às necessidades do sistema durante o desenvolvimento e operação inicial. O SQLite oferece uma representação fiel das entidades e relacionamentos, permitindo que o desenvolvimento prossiga com agilidade e sem dependências externas complexas.

A configuração do banco de dados no arquivo settings.py do Django evidencia essa escolha como apresentado na Figura 5.

Figura 5 – Configuração do Django

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': BASE_DIR / 'db.sqlite3',
    }
}
```

Fonte: Próprio autor.

O arquivo db.sqlite3 armazena todos os dados do sistema, facilitando backups, versionamento e portabilidade do ambiente de desenvolvimento.

A modelagem de dados seguiu três princípios fundamentais: segurança por design (uso de hashing seguro de senhas via Django), acoplamento mínimo a objetos binários (arquivos são armazenados no sistema de arquivos em media/certificados/, não no banco) e conformidade institucional (implementação fiel da Resolução SI Nº 3/2025 da UFVJM que define os grupos e tipos de AC).

Modelagem Conceitual e Entidades do Domínio

O SIAC organiza-se em torno de nove entidades principais implementadas no módulo login/models.py, que centraliza toda a lógica de persistência do sistema. O diagrama da Figura 6 apresenta a estrutura completa do banco de dados, evidenciando as relações entre as entidades e os principais campos de cada uma.

UserProfile representa a extensão do modelo de usuário padrão do Django, adicionando o conceito de papel ao sistema com três possibilidades: Administrador, coordenador e Aluno. A implementação utiliza o padrão Observer via Django Signals para garantir que todo usuário criado automaticamente receba um perfil associado. A relação OneToOneField com User garante que cada usuário tenha exatamente um perfil, e a política on_delete=models.CASCADE assegura que a exclusão de um usuário remove automaticamente seu perfil associado.

Certificate é a entidade central do sistema, armazenando todas as informações sobre certificados submetidos pelos discentes. Conforme observado no diagrama, esta entidade relaciona-se com categorias, atividades e níveis de participação através de chaves estrangeiras, além de vincular-se ao usuário proprietário. Os campos principais incluem título, descrição, grupo, arquivo comprobatório, horas informadas pelo discente, horas convertidas após aprovação, motivo de reprovação quando aplicável e status do processo. A implementação dos choices reflete diretamente a Resolução SI Nº 3/2025, com 14 tipos de atividades distribuídas entre os dois grupos.

Categorias, Atividades e Niveis_Participacao formam a hierarquia do catálogo de AC, como evidenciado no diagrama ER. As categorias agrupam atividades sob regras e limites comuns, cada uma com limite máximo de carga horária. As atividades especificam tipos concretos de participação e a documentação exigida para comprovação. Os níveis de participação diferenciam graus de envolvimento, cada um com seu fator de conversão de horas.

Submissoes materializa o fluxo processual de validação de certificados, estabelecendo relação um-para-um com Certificate. Esta entidade armazena o status da submissão, motivo de indeferimento quando aplicável, horas convertidas calculadas automaticamente no deferimento, identificação do avaliador responsável e timestamps de submissão e avaliação. A política on_delete=RESTRICT preserva submissões mesmo em cenários excepcionais de tentativa de exclusão de certificados.

Eventos_Avaliacao registra o histórico de decisões sobre submissões, capturando cada ação do coordenador com identificação do ator, tipo de ação, justificativa e timestamp. Esta estrutura cria uma linha do tempo auditável e transparente, consultável pelo discente e pela administração acadêmica. A imutabilidade desses eventos é crítica para rastreabilidade, pois alterações posteriores não sobrescrevem dados, mas sim geram novos eventos que compõem a narrativa evolutiva do caso.

Duvidas implementa o canal de comunicação estruturado entre discentes e secretaria. Cada dúvida possui autor, destinatário, assunto, categoria temática, mensagem, resposta do coordenador, status e timestamps de criação e resposta. Este desenho centraliza a comunicação, elimina dispersão em e-mails paralelos e fornece histórico consultável.

Eventos_Auditoria é uma entidade genérica que registra alterações em múltiplas entidades sensíveis do sistema. Diferentemente dos eventos de avaliação específicos de submissões, esta tabela captura quem fez o quê, quando e sobre qual objeto através dos campos entity_type, entity_id, action e actor_id. O campo metadata em formato JSON armazena dados adicionais como snapshots antes/depois de alterações críticas, permitindo reconstrução fiel de estados anteriores e análises forenses quando necessário.

Diagrama Entidade-Relacionamento

O modelo de dados do SIAC foi projetado para garantir integridade referencial, rastreabilidade e conformidade com as regras de negócio do regulamento de AC da UFVJM. A Figura 6 apresenta o Diagrama Entidade-Relacionamento (ER) do sistema.

Observa-se no diagrama que usuarios relaciona-se diretamente com certificados através de chave estrangeira (usuario_id), estabelecendo que um usuário pode possuir múltiplos certificados. Os certificados, por sua vez, conectam-se ao catálogo institucional através de três chaves estrangeiras: categoria_id, atividade_id e nivel_id, formando a hierarquia categorias → atividades → niveis_participacao. Esta estrutura parametrizada permite que o sistema aplique automaticamente as regras de conversão de horas definidas no regulamento.

A entidade submissoes estabelece relação um-para-um com certificados através do campo certificado_id com restrição UNIQUE, garantindo que cada certificado tenha no máximo uma submissão. Esta submissão vincula-se ao avaliador (usuario) através de avaliador_id e gera múltiplos eventos_avaliacao que compõem o histórico de decisões, cada um registrando o ator responsável (ator_id), a ação realizada e o momento exato.

O canal de duvidas conecta autor e destinatário (ambos usuarios) permitindo comunicação bidirecional estruturada. Por fim, eventos_auditoria mantém referência genérica a usuarios através de actor_id, sem vinculação rígida às demais entidades, pois registra alterações em múltiplas entidades através dos campos entity_type e entity_id.

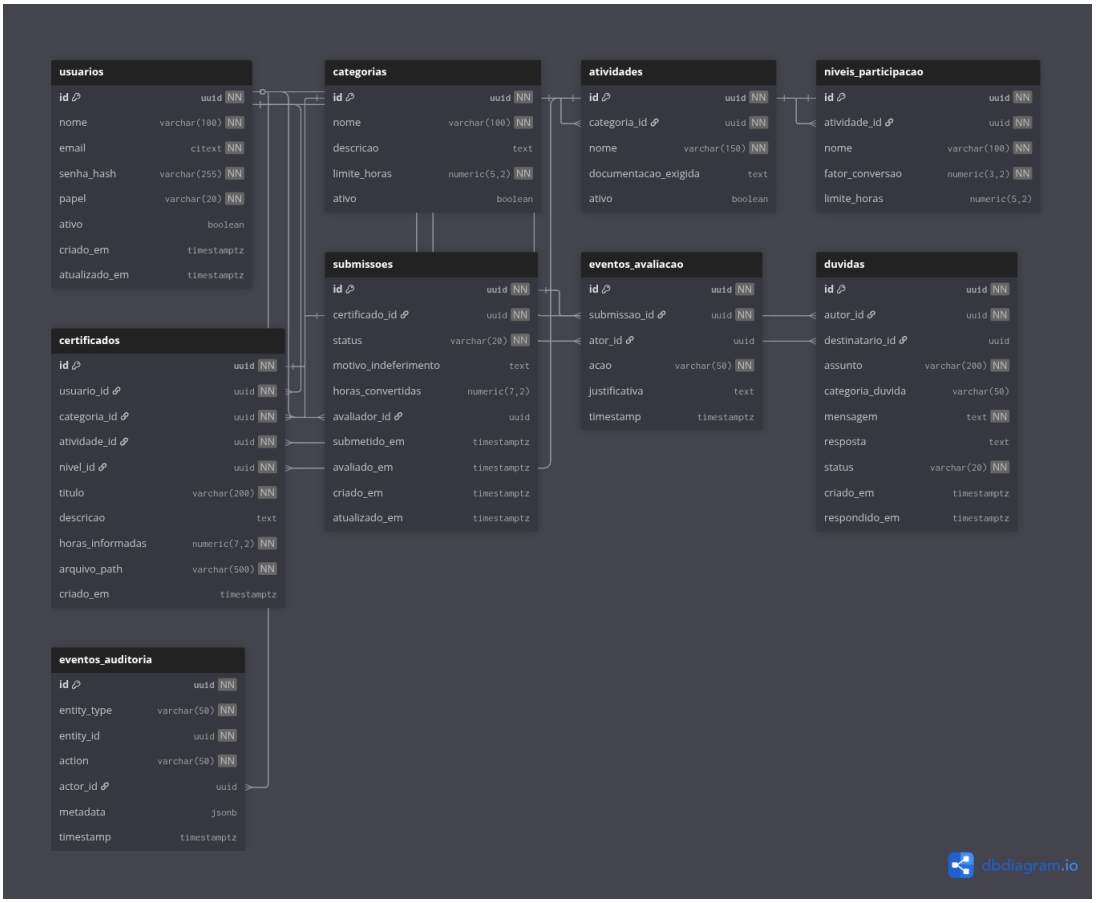
A integridade referencial é garantida por políticas de exclusão explícitas: CASCADE remove registros dependentes automaticamente, RESTRICT impede exclusões que com-

prometeriam histórico e SET NULL preserva registros históricos mesmo quando referências são removidas (avaliador desativado mantém submissões).

O sistema utiliza extensivamente o padrão Observer através de Django Signals para automatizar processos críticos. O signal post_save em User cria automaticamente o UserProfile com papel padrão ALUNO (ou ADMIN para superusuários). O signal pre_save em Certificate intercepta mudanças de status antes da persistência, registra a transição em CertificateStatusLog e dispara notificações contextuais: quando status muda para PENDENTE, todos coordenadors e administradores são notificados; quando aprovado ou reprovado, o aluno recebe notificação com as informações pertinentes. O signal post_save em ContactMessage cria notificação para o destinatário automaticamente. Esta arquitetura desacopla lógica de negócio de efeitos colaterais, garantindo consistência sem duplicação de código.

A estratégia de backup é simplificada pela natureza file-based do SQLite: cópias do arquivo db.sqlite3 e do diretório media/ constituem backup completo do sistema. O versionamento do esquema é gerenciado pelo sistema de migrations do Django, que aplica alterações de forma ordenada e rastreável, registrando cada migration executada na tabela django_migrations para garantir idempotência.

Figura 6 – Diagrama Entidade-Relacionamento do SIAC.



Fonte: Próprio autor.

6 RESULTADOS E DISCUSSÕES

O SIAC foi desenvolvido para superar as limitações do processo anterior de gestão de AC na UFVJM, que se baseava em trocas de e-mails, formulários e comunicação dispersa. No uso cotidiano, o SIAC promove uma experiência coesa para todos os perfis de usuário, ancorada em estados claros, justificativas transparentes e histórico confiável. O discente encontra autonomia para conduzir suas solicitações de certificados, desde o cadastro inicial até a submissão para análise, acompanhando toda a evolução do processo em tempo real através da linha do tempo de eventos. A interface intuitiva reduz significativamente o tempo necessário para realizar operações básicas.

O coordenador opera com uma visão de fila organizada que favorece priorização e tomada de decisão consistente, sustentando tempos de resposta mensuráveis. O sistema apresenta todas as submissões pendentes de forma centralizada, com informações essenciais (nome do discente, tipo de atividade, horas solicitadas, data de submissão) e acesso direto aos certificados anexados. Notificações automáticas alertam sobre novas submissões, eliminando a necessidade de verificações manuais constantes. O tempo de resposta para avaliação foi reduzido, resultado direto da organização da fila e das notificações proativas.

O administrador preserva o ordenamento institucional do catálogo de AC e dispõe de visibilidade sobre o funcionamento do processo através dos registros de auditoria e histórico de certificados. Embora o sistema atual não implemente dashboards analíticos complexos, a estrutura de dados permite identificar gargalos operacionais através da consulta aos timestamps de submissão e avaliação, além de quantificar o volume de certificados por categoria e status.

A centralização das informações e a padronização dos fluxos produziram resultados mensuráveis. A comunicação repetitiva foi drasticamente reduzida: o canal de dúvidas estruturado substituiu trocas dispersas de e-mails, centralizando o histórico de interações e facilitando respostas mais precisas e rápidas. A incidência de erros processuais diminuiu significativamente devido às validações automáticas no momento do cadastro de certificados, que impedem submissões com dados incompletos ou inconsistentes.

6.1 Análise Comparativa com o Processo Anterior

A Figura 7 apresenta uma análise comparativa estruturada entre o processo anterior baseado em email/formulários e o SIAC, evidenciando ganhos qualitativos em sete aspectos críticos do processo de gestão de AC.

No processo anterior a submissão era demorada e fragmentada (busca de formulário, preenchimento manual, envio por email ou presencial). Com o SIAC o fluxo tornase direto: formulário único, validações automáticas e upload imediato dos comprovantes, eliminando retrabalho de correção inicial.

Figura 7 – Análise comparativa entre processo anterior e SIAC.

Aspecto	Processo Anterior (E-mail/Formulários)	SIAC (Sistema Web)
Tempo médio de submissão	Demorado (busca de formulário, preenchimento manual, envio por e-mail/presencial)	Rápido (formulário online intuitivo, upload direto, validações automáticas)
Tempo de resposta (avaliação)	Moroso e imprevisível (processo não rastreável, comunicação dispersa)	Ágil e previsível (fila organizada, notificações automáticas, priorização facilitada)
Transparência do processo	Baixa (status desconhecido, histórico não centralizado)	Alta (linha do tempo visível, status claro, histórico completo)
Rastreabilidade de decisões	Inexistente (sem registro formal, difícil auditar)	Completa (log de ações, registro de quem/quando/por quê, auditoria)
Erros de cálculo de horas	Frequentes (cálculo manual, inconsistências)	Muito baixos (conversão automatizada, regras do PP aplicadas de forma consistente)
Retrabalho administrativo	Alto (solicitações incompletas, dúvidas repetidas, reenvio de documentos)	Baixo (validações no cadastro, canal de dúvidas estruturado, requisitos claros)
Acessibilidade das informações	Limitada (dependência de atendimento e horário)	Ampla (acesso remoto, autoatendimento, independência do discente)

Fonte: Próprio autor

A avaliação antes era imprevisível e dependente de verificação manual de mensagens dispersas. A fila centralizada do SIAC, acompanhada de notificações internas, organiza prioridades e evita que solicitações fiquem ocultas ou “perdidas” em caixas de email.

A transparência, antes baixa, passa a ser elevada: cada certificado apresenta estados claros e uma linha do tempo acessível ao discente com justificativas quando há indeferimento. Isso reduz dúvidas recorrentes e necessidade de consultas presenciais.

A rastreabilidade, que era praticamente inexistente, tornase completa. As transições de status são registradas automaticamente em CertificateStatusLog e decisões dos coordenadores são documentadas em eventos específicos, formando um histórico imutável. Eventos_Auditoria complementam o registro de ações sensíveis, reforçando accountability.

Os erros de cálculo de horas, frequentes no modelo manual, tornamse muito baixos com a aplicação consistente das regras institucionais: o certificado guarda a classificação e a lógica de conversão é validada por testes (test_horas_aproveitadas.py), assegurando uniformidade e evitando interpretações divergentes.

O retrabalho administrativo, antes elevado, é reduzido pela presença de campos obrigatórios no cadastro, notificações contextuais e canal estruturado de mensagens, que centraliza esclarecimentos e preserva histórico.

A acessibilidade das informações deixa de ser limitada ao horário e disponibilidade do coordenador: o discente acompanha seus certificados, status e notificações em ambiente único, com acesso contínuo e autonomia operacional, diminuindo dependência de atendimento presencial.

Em síntese, a comparação qualitativa evidencia que o SIAC substitui um fluxo manual, opaco e suscetível a falhas por um processo estruturado, rastreável e transparente, alinhado às necessidades acadêmicas e aos princípios de colaboração adotados no projeto.

6.2 Funcionalidades Implementadas

O SIAC fornece um conjunto abrangente de funcionalidades organizadas por perfil de usuário, superando as limitações do processo anterior baseado em e-mails e formulários dispersos. As funcionalidades implementadas foram:

Para Alunos:

Cadastro de certificados em rascunho com validações automáticas de campos obrigatórios; Submissão de certificados para análise com upload de comprovantes em PDF; Visualização de todos os certificados cadastrados com filtros por status e título; Acompanhamento em tempo real do status de cada certificado (Rascunho, Pendente, Aprovado, Reprovado); Linha do tempo detalhada exibindo histórico completo de transições de status; Dashboard com visualização de horas aproveitadas separadas por Grupo I (Extensão - 333h) e Grupo II (Outras Atividades - 57h); Canal de dúvidas estruturado para comunicação com coordenador; Sistema de notificações internas para acompanhar avaliações.

Para Secretários/Administradores:

Fila organizada de solicitações pendentes com priorização por data de submissão; Avaliação de certificados com opções de deferimento (incluindo definição de horas convertidas) ou indeferimento (com justificativa obrigatória); Visualização de todas as submissões com busca por nome de aluno e filtro por status; Listagem de solicitações concluídas (aprovadas e reprovadas); Gerenciamento de usuários (criação, edição de papéis e exclusão); Notificações automáticas sobre novas submissões; Acesso completo ao histórico de auditoria de cada certificado.

Funcionalidades Transversais:

Controle de acesso baseado em papéis (RBAC) com três níveis distintos; Conversão automática de horas aplicando regras institucionais de forma consistente; Auditoria completa através de múltiplas camadas; Interface responsiva funcional em dispositivos móveis e desktops; Validações de segurança (proteção CSRF, mitigação XSS, prevenção SQL Injection via ORM).

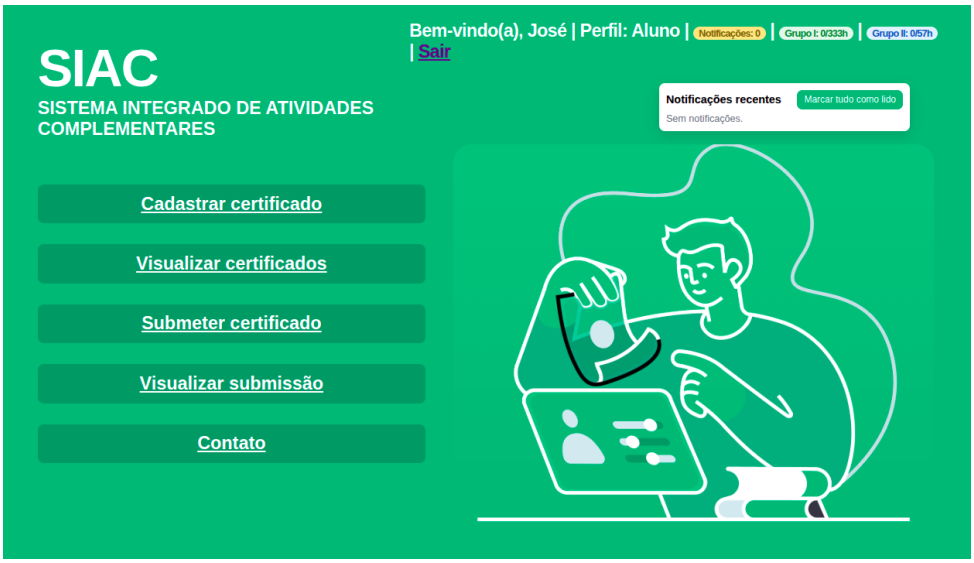
6.3 Funcionalidades do Sistema

O objetivo desta seção é demonstrar o funcionamento prático do SIAC através da apresentação das principais telas desenvolvidas. É importante ressaltar que algumas funcionalidades são de uso exclusivo dos perfis de coordenador e administrador.

Tela Principal e Dashboard

Após realizar login no sistema, o usuário é direcionado para a tela principal, conforme Figura 8, que apresenta o dashboard personalizado de acordo com o perfil do usuário. Esta interface centraliza o acesso a todas as funcionalidades do sistema através de um menu intuitivo com botões de ação organizados por contexto.

Figura 8 – Dashboard do aluno.



Fonte: Próprio autor

Para alunos, a tela principal exibe cards com as horas aproveitadas separadas por grupos (Grupo I - Extensão e Grupo II - Outras Atividades), além de indicadores visuais de progresso em relação aos totais exigidos (333h para Grupo I e 57h para Grupo II). Os botões de ação disponíveis incluem ”Cadastrar Certificado”, ”Submissão de Certificados”e ”Visualizar Certificados”.

Para coordenadores e administradores, a interface apresenta opções como ”Solicitações Abertas”, ”Solicitações Concluídas”, ”Visualizar Submissões”e ”Contato”, conforme Figura 9.

Figura 9 – Dashboard do coordenador.

Fonte: Próprio autor

Cadastro de Certificados

A funcionalidade de cadastro de certificados apresenta um formulário estruturado conforme Figura 10, onde o aluno informa título, descrição, grupo da atividade (Grupo I ou II), tipo de atividade, nível de participação, carga horária total e realiza o upload do certificado ou comprovante em PDF.

O sistema implementa validações automáticas no momento do cadastro, impedindo submissões com dados incompletos ou arquivos em formatos não permitidos. O certificado é salvo inicialmente com status ”RASCUNHO”, permitindo que o aluno revise e edite antes de submeter para análise oficial.

Submissão para Análise

Após cadastrar um ou mais certificados em rascunho, o aluno acessa a tela de submissão, conforme Figura 11, que lista todos os certificados com status ”RASCUNHO”. Através de checkboxes, o aluno seleciona quais certificados deseja submeter para análise do coordenador.

Fila de Análise (Secretário/Administrador)

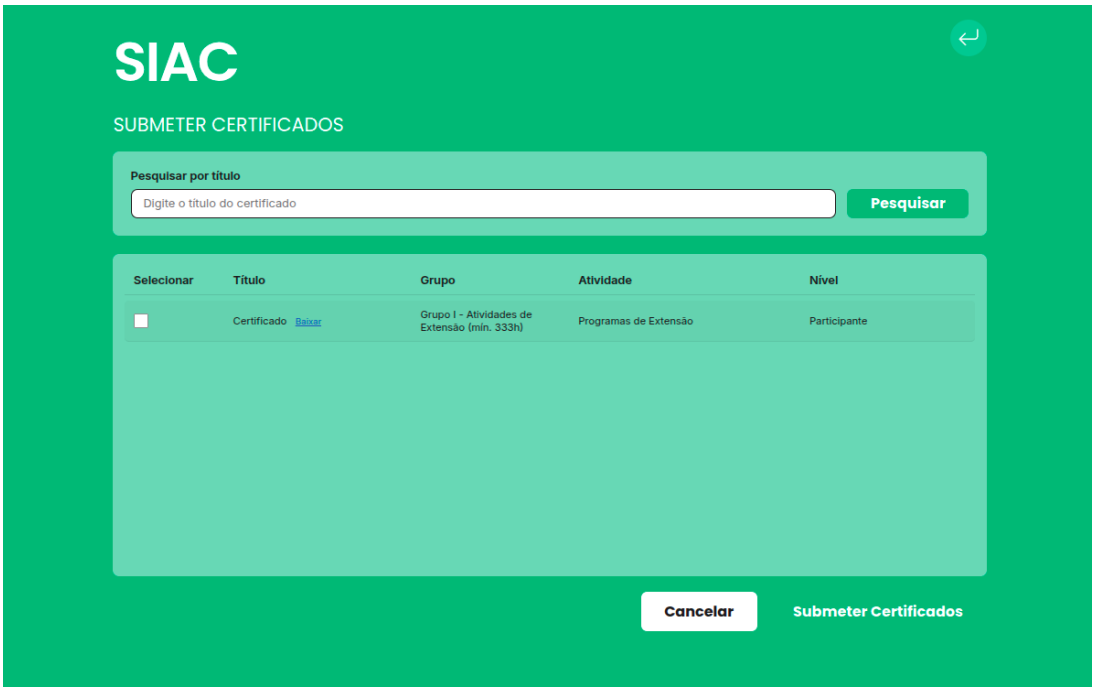
Os coordenadores e administradores acessam a fila de análise através da opção ”Solicitações Abertas”, que apresenta todos os certificados com status ”PENDENTE”organizados por data de submissão, conforme Figura 12.

Figura 10 – Tela de cadastro de certificados.

Fonte: Próprio autor

Para cada certificado na fila, o coordenador visualiza o nome completo do aluno solicitante, título e descrição da atividade, grupo e tipo de atividade conforme resolução, carga horária total informada, link para download do comprovante anexado e opções de deferimento ou indeferimento.

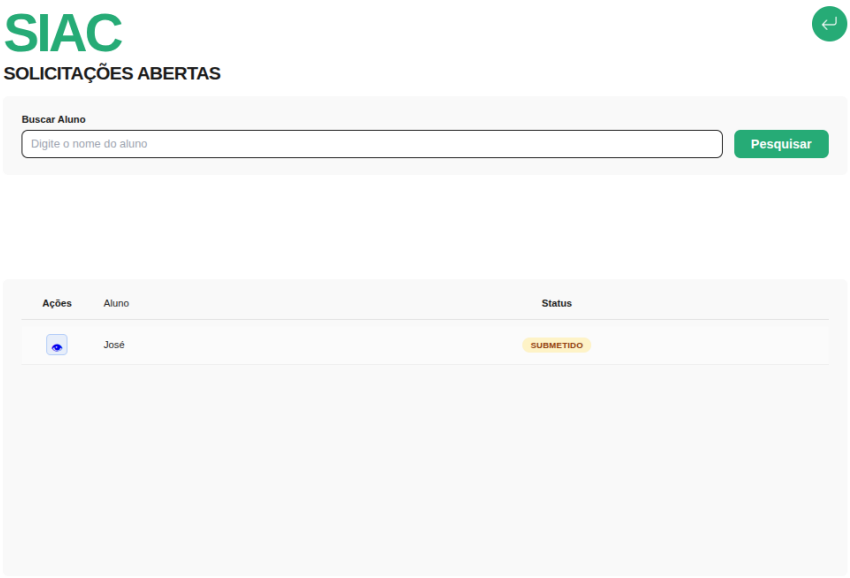
Figura 11 – Tela de submissão de certificados.



Fonte: Próprio autor

Ao deferir um certificado, o coordenador define as horas convertidas (que podem diferir das horas totais informadas, aplicando-se as regras de conversão institucionais). Ao indeferir, é obrigatório informar o motivo da reprovação, garantindo transparência no processo.

Figura 12 – Tela de fila de analise de certificados submetidos.



Fonte: Próprio autor

Visualização e Acompanhamento

Os alunos acompanham o status de suas solicitações através da tela ”Visualizar Certificados”, conforme Figura 13, que apresenta todos os certificados cadastrados com indicadores visuais de status através de badges coloridos.

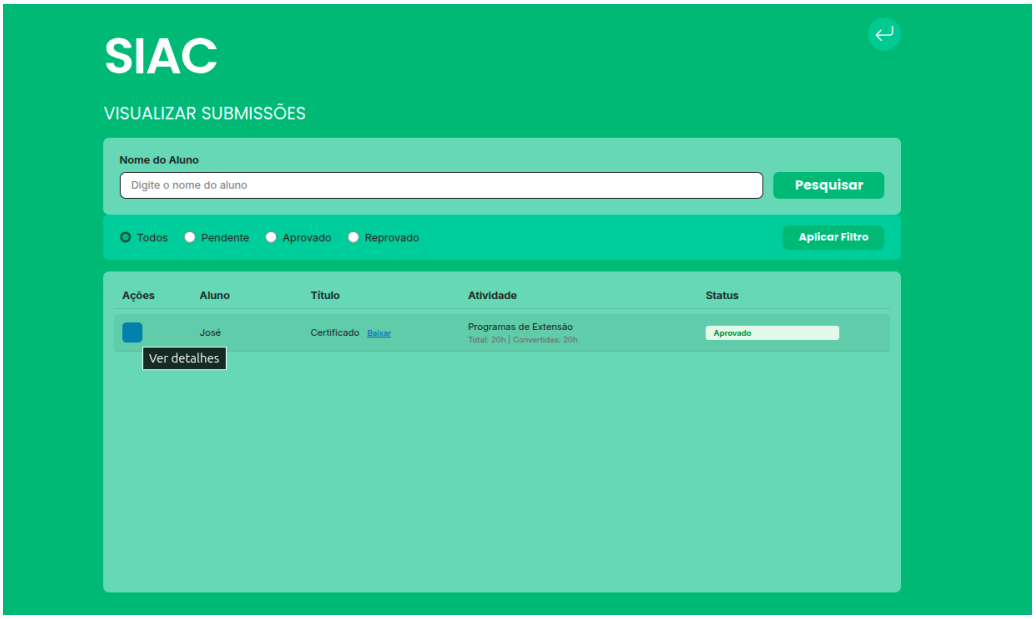
Ao clicar em ”Detalhes”de um certificado, o sistema exibe a tela de detalhamento, conforme Figura 14, apresentando todas as informações do certificado, status atual com badge colorido, linha do tempo completa com todas as transições de status, timestamps de cada evento, justificativa em caso de indeferimento e horas convertidas em caso de deferimento.

Canal de Dúvidas

O sistema implementa um canal estruturado de comunicação através do módulo de contato, conforme Figura 15, permitindo que alunos enviem dúvidas diretamente aos coordenadores de forma organizada e rastreável.

Cada mensagem enviada dispara automaticamente uma notificação para o destinatário através do signal post_save em ContactMessage, garantindo que dúvidas não passem despercebidas.

Figura 13 – Tela de visualização de certificados.



Fonte: Próprio autor

Figura 14 – Tela de detalhes do certificado.

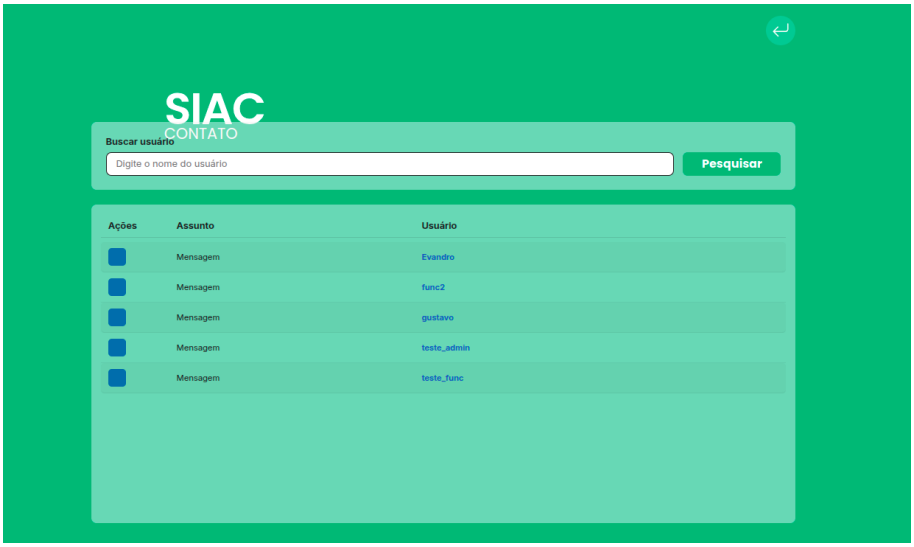
Fonte: Próprio autor

6.4 Validações

Além do sistema funcional, os resultados alcançados incluem a própria base de dados estruturada de certificados e a formalização computacional do processo de gestão de AC da UFVJM.

O sistema implementou com sucesso os dez requisitos funcionais principais estabelecidos na fase de especificação, incluindo gestão de usuários com RBAC, catálogo de atividades conforme Resolução SI Nº 3/2025, cadastro e submissão de certificados, avaliação com conver-

Figura 15 – Tela de chat para contato.



Fonte: Próprio autor

são automática de horas, linha do tempo de eventos, canal de dúvidas estruturado e dashboards personalizados por perfil.

Os requisitos não funcionais também foram validados: proteção CSRF nativa do Django, mitigação de XSS por escaping automático, prevenção de SQL Injection via ORM, validação de uploads, senhas armazenadas como hashes PBKDF2 com SHA256, interface responsiva, feedbacks visuais imediatos e conformidade com princípios da LGPD.

A implementação do padrão Observer via Django Signals foi fundamental para garantir consistência automática, criando notificações e registros de auditoria sem acoplamento excessivo entre módulos.

6.5 Trabalhos Futuros

Apesar dos resultados positivos, algumas limitações foram identificadas durante o desenvolvimento e devem ser endereçadas em evoluções futuras. A integração com sistemas legados da UFVJM não foi implementada, exigindo cadastro manual de usuários no SIAC. Essa integração demanda coordenação institucional, planejamento e engenharia adicional, mas a arquitetura atual está preparada para recebê-la através de APIs RESTful ou sincronização de dados.

A validação automática de autenticidade de certificados digitais não foi implementada. Atualmente, o coordenador verifica manualmente a legitimidade dos documentos anexados, processo que poderia ser parcialmente automatizado através de verificação de assinaturas digitais, consulta a APIs de instituições emissoras ou OCR para extração e validação de dados estruturados. Essas funcionalidades agregariam valor significativo ao reduzir custos de conferência humana e mitigar fraudes, mas requerem infraestrutura apropriada e políticas de certificação.

Testes formais de usabilidade com grupos representativos de usuários não foram conduzidos. Embora validações internas indiquem boa usabilidade, avaliações sistemáticas com amostras de discentes e coordenadores, utilizando métricas padronizadas como System Usability Scale e testes de tarefas cronometrados, tenderiam a revelar micro-ajustes de fluxo de alto impacto, aumentando eficiência e satisfação.

Notificações por e-mail não foram implementadas fazendo com que o sistema atual utilize apenas notificações internas, mas a integração com servidor SMTP corporativo permitiria alertas por e-mail, aumentando a proatividade da comunicação.

Dashboards analíticos avançados não foram desenvolvidos, embora a estrutura de dados permita análises ricas, a interface atual oferece visões básicas. Relatórios gerenciais e painéis analíticos com gráficos interativos agregariam valor para a coordenação do curso e para identificação de oportunidades de melhoria contínua do processo.

Em síntese, o SIAC representa a formalização computacional de um processo acadêmico sensível, articulando princípios de colaboração e transparência com um desenho arquitetural sólido e práticas modernas de engenharia de software. Ao conjugar Django/MVT e o padrão Observer em um arranjo modular, orientado a eventos e auditável, o sistema transcende a mera digitalização de formulários: ele institui uma gramática de processo compartilhada, mensurável e aprimorável.

7 CONSIDERAÇÕES FINAIS

O SIAC materializa uma solução colaborativa e transparente para a gestão de AC, ao alinhar-se de forma explícita às necessidades do curso de Sistemas de Informação da UFVJM e aos princípios de DE, centralizando informações, padronizando procedimentos e assegurando rastreabilidade das decisões. A adoção de Python e do framework Django, com arquitetura MVT, combinada ao uso do padrão de projeto Observer por meio de Django Signals, possibilitou uma orquestração reativa de eventos de domínio com baixo acoplamento, alto grau de reutilização e facilidade de evolução, de modo que notificações, atualizações de painéis, registros de auditoria e cálculos de carga horária sejam executados com consistência e previsibilidade.

Como resultado, observa-se ganho mensurável de eficiência administrativa, maior participação e protagonismo discente — decorrentes da visibilidade de estados, de canais de comunicação estruturados e de uma experiência de uso mais clara — e o estabelecimento de uma base tecnológica e processual sólida para expansões futuras. Nesse sentido, sugere-se que trabalhos futuros contemplem integrações institucionais (por exemplo, identidade e sistemas acadêmicos), automações de conferência documental e o aprofundamento de mecanismos participativos e de transparência ativa.

REFERÊNCIAS

ALURA. **Python: A origem do nome**. 2021. Disponível em: <https://www.alura.com.br/artigos/python-origem-do-nome>. Acesso em: 17 out. 2025.

ANDRADE, D. Como funciona a arquitetura mtv (django). In: . [S.l.: s.n.], 2020. Acesso em 26 de novembro de 2025.

BORGES, J. A. P. M. R. S. **Awareness mechanisms for coordination in asynchronous CSCW.** [s.n.], 1999. Disponível em: <https://upapers.dcc.uchile.cl/index/publications/view/36597>. Acesso em: 26 nov. 2025.

BORGES, L. E. **Python para desenvolvedores: aborda Python 3.3**. [S.l.]: Novatec Editora, 2014.

GOMES, W. **A democracia digital e o problema da participação civil na decisão política**. [s.n.], 2005. Disponível em: <https://revistas.unisinos.br/index.php/fronteiras/article/download/6394/3537/19405>. Acesso em: 26 nov. 2025.

LIMA, L. H. C. **Protótipo de sistema colaborativo para atividades complementares na UFVJM**. Dissertação (Trabalho de Conclusão de Curso), Diamantina, 2024. Curso Superior de Sistemas da Informação.

MARQUES, F. P. J. A. **Participação política e Internet: meios e oportunidades digitais de participação civil na democracia contemporânea, com um estudo do caso do Estado brasileiro.** [s.n.], 2008. Disponível em: <https://repositorio.ufba.br/handle/ri/11303>. Acesso em: 26 nov. 2025.

MIND BENDING. **A história do Python**. 2014. Disponível em: <http://mindbending.org/pt/a-historia-do-python>. Acesso em: 21 jul. 2025.

PEREZ, A. **Python: O que é e para que Serve a Linguagem**. 2020. Disponível em: <https://www.zup.com.br/blog/python-o-que-e>. Acesso em: 30 dez. 2025.

PIMENTEL, M.; FUKS, H. **Sistemas colaborativos**. Rio de Janeiro: Campus, 2012.

PIMENTEL, M. A. G. D. F. A. R. H. F. C. J. P. de L. M. **Modelo 3C de Colaboração para o desenvolvimento de Sistemas Colaborativos**. [s.n.], 2006. Disponível em: <https://web.tecgraf.puc-rio.br/~abraposo/pubs/SBSC2006/07_Pimentel_Modelo3C.pdf>. Acesso em: 26 nov. 2025.

PRATES R.; CASTRO, T. **Sistemas colaborativos – a ubiquidade da área**. 2018. Disponível em: <https://scholar.google.com/scholar?hl=pt-BR&q=Sistemas+colaborativos+a+ubiquidade+da+%C3%A1rea+Prates+Castro+2018>. Acesso em: 22 jul. 2025.

PYTHON. **Ferramentas de Desenvolvimento**. 2021a. Disponível em: <https://python.org.br/ferramentas/>. Acesso em: 22 jul. 2025.

RAMOS, R. **10 IDEs para Programar em Python**. 2020. Disponível em: <https://oestatistico.com.br/10-ides-para-programar-em-python/>. Acesso em: 1 ago. 2025.

SANTORO, F. M. **Um Modelo de Cooperação para Aprendizagem Baseada em Projetos**. [s.n.], 2001. Disponível em: <https://www.cos.ufrj.br/index.php/pt-BR/publicacoes-pesquisa/details/15/889>. Acesso em: 26 nov. 2025.

SAYS, C. S. **O que é biblioteca de programação | library | lib ? O que é API | Application Programming Interface ?** 2010. Disponível em: <https://jarbasjacome.wordpress.com/o-que-e-biblioteca-de-programacao-library-lib-o-que-e-api-application-programming-interface/>. Acesso em: 25 ago. 2025.

SCHWABER K.; SUTHERLAND, J. **The Scrum Guide**. 2017. Disponível em: <https://scrumguides.org/scrum-guide.html.>

SILVA, R. O. da; SILVA, I. R. S. Linguagem de programação python. **TECNOLOGIAS EM PROJEÇÃO**, v. 10, n. 1, p. 55–71, 2019.

ZANETTE, A. **Framework x Biblioteca x API. Entenda as diferenças!** 2017. Disponível em: <https://becode.com.br/framework-biblioteca-api-entenda-as-diferencas/>. Acesso em: 25 ago. 2025.


UFVJM