

Sistema de Gestão de Documentos Eletrônicos: atas de reunião

Cleber Filipini da Silva¹, Alexandre de O. Zamberlan^{1 2}

¹Curso de Sistemas de Informação – Universidade Franciscana (UFN)
Santa Maria – RS

²Laboratório de Práticas – Universidade Franciscana (UFN)
Santa Maria – RS

{filipini.cleber, alexz}@ufn.edu.br

Abstract. *This research deals with computerized document management, as Brazilian legislation has made it easier for organizations and/or institutions to manipulate documents in their electronic form, instead of their physical form, freeing up spaces and assisting in the various queries of content in documents stored. The objective was to implement a Web document management system of the "ata" type, for coordinators and secretariats in educational institutions. The work was supported by a bibliographical review and a case study, in which was built following the best practices of the Scrum methodology, using the Python language and its Django framework as technologies.*

Resumo. *O presente trabalho trata de gestão informatizada de documentos, uma vez que a legislação brasileira têm facilitado às organizações em manipular documentos na sua forma eletrônica, em detrimento de sua forma física, liberando espaços e auxiliando em consultas diversas a conteúdos presentes nos documentos armazenados. O objetivo deste trabalho foi implementar um sistema Web de gestão de documentos do tipo ata, para coordenações e secretarias de cursos de instituições de ensino. O trabalho foi amparado por revisão bibliográfica e estudo de caso, e o sistema foi construído seguindo as boas práticas da metodologia Scrum, tendo como tecnologias a linguagem Python e seu framework Django.*

1. Introdução

Sistemas de Informação é uma área da Ciência da Computação que auxilia, efetivamente, na gestão de tantas outras áreas do saber, por se tratar de um conjunto organizado de pessoas, processos e recursos, os quais interagem entre si para tratar, processar, armazenar e divulgar tanto informação, quanto conhecimento em uma organização [Laudon and Laudon 2014] e [Baldam 2017]. Sistemas de informação têm um papel fundamental na gestão do conhecimento, pois fornece recursos para apoiar a criação, a transferência e a aplicação do conhecimento, promovendo diminuição de gastos em produtos e gerando investimento em capital intelectual.

Segundo [Baldam 2017], um dos recursos da Gestão do Conhecimento, implementados pela área de Sistemas de Informação é a Gestão Eletrônica de Documentos (GED), que é uma tecnologia que provê um meio de facilmente gerar, controlar, armazenar, compartilhar e recuperar informações existentes em documentos.

Os sistemas GED permitem aos usuários acessarem os documentos de forma ágil e segura, normalmente via navegador Web. A capacidade de gerenciar documentos é uma ferramenta indispensável para a Gestão do conhecimento [Baldam 2017]. De acordo com esse mesmo autor, qualquer tipo de organização deve utilizar GED, uma vez que é fundamental facilitar, de forma transparente, o acesso aos documentos produzidos nas organizações, principalmente textos oficiais como requerimentos, ofícios e atas, por exemplo. Além disso, essas organizações tornam-se transparentes a órgãos de fiscalização e/ou vigilância, pois tais órgãos podem acessar documentos das organizações sem visitas presenciais. Lembrando que qualquer GED deve obedecer a Lei Geral de Proteção de Dados (13.709/2018), que objetiva proteger os direitos fundamentais de liberdade, de privacidade e o livre desenvolvimento da personalidade da pessoa natural, além da criação de um cenário de segurança jurídica [Ministério Público Federal 2021].

Dessa forma, este trabalho tem como motivação a gestão de documentos do tipo ata, gerados em reuniões em cursos de Graduação e/ou Pós-graduação em instituições de ensino superior, já que a legislação brasileira têm facilitado às instituições em manipular documentos na sua forma eletrônica, em detrimento de sua forma física.

O objetivo deste trabalho foi projetar e implementar um sistema Web de gestão de documentos. Como objetivos específicos, assumiram-se: i) revisão bibliográfica sobre gestão de documentos; ii) revisão bibliográfica sobre Sistemas de Informação; iii) pesquisa e aplicação de melhoras práticas no desenvolvimento de software via Scrum e a técnica Kanban; iv) projeto e implementação do sistema Web.

Para facilitar a compreensão do trabalho, o texto foi dividido em seções. A próxima seção, traz conceitos e a discussão sobre Sistemas de Informação, Gestão de Documentos, Metodologia Scrum e boas práticas no desenvolvimento (versionamento de código e uso de *frameworks*). Em seguida, apresentou-se alguns trabalhos relacionados que colaboram com a construção do sistema desenvolvido. A seção 3, Proposta de Trabalho, mostrou-se a modelagem e alguns resultados alcançados na construção do sistema. Por fim, Conclusões e Referências Bibliográficas.

2. Revisão bibliográfica

Nesta seção, são abordados e discutidos conceitos sobre Sistemas de Informação, Sistema de Gestão de Documentos Eletrônicos, Metodologia Scrum e Trabalhos Relacionados.

2.1. Sistemas de Informação

A área de Sistemas de Informação (SI) possui definições, categorias e aplicações. Segundo apresentado em [O'Brien and Marakas 2013] e [Laudon and Laudon 2014], SI e seus recursos têm como foco coletar, processar, armazenar e apresentar informações. Todo esse conjunto deve contemplar os processos de tomada de decisão e de controle da organização, seja pelo setor da Tecnologia da Informação (TI), seja pelo dono ou responsável do negócio (*stakeholders*).

Em relação às categorias, destacam-se: Sistemas de Informação Gerencial (SIG) e Sistemas de Apoio à Decisão (SAD). SIG fornece condições de coletar, tratar, armazenar e, posteriormente, entregar informações (relatórios), enquanto o SAD auxilia na solução de problemas ou situações específicas, apontando tendências e/ou ocorrências percebidas.

das nos relatórios e/ou documentos entregues pelo SIG [O'brien and Marakas 2013] e [Laudon and Laudon 2014].

O uso da Internet para execução de sistemas (*online*) distribuídos geograficamente é um fato nas diferentes áreas. Usar sistemas em nuvem é a escolha para sistemas que executam em navegadores ou em aplicativos móveis. Portanto, o conceito de *Cloud Computing* (Computação em Nuvem) refere-se ao processamento computacional e armazenamento de dados que são executados fora da infraestrutura local [dos Santos et al. 2021]. Ou seja, através de uma conexão com a Internet é possível utilizar serviços computacionais (processamento, banco de dados, etc.) de modo seguro. Com isso, todo o processamento e armazenamento deixa de existir no lado do cliente, mas sim no lado do servidor, permitindo manutenção e garantias de segurança do sistema em um ambiente centralizado, seguro e controlado [Sangiovo et al. 2018].

2.2. Sistema de Gestão de Documentos Eletrônicos

Em instituições e/ou organizações diversas há geração e manipulação de documentos, que muitas vezes são formais e com informações importantes (ou sigilosas), como por exemplo ofícios, requerimentos, atas de reuniões, entre outros. Dessa forma, um documento é a representação material da manifestação de pensamento, como uma voz fixada permanentemente [ESMEC 2020].

Um documento eletrônico é todo registro que tem como meio físico um suporte eletrônico. E conforme [ESMEC 2020], para a plena eficácia comprobatória do documento, é preciso que ele possua a capacidade de armazenar informações de forma que impeça ou permita detectar eliminação ou adulteração de conteúdo.

Segundo [Bax et al. 2011], a gestão de documentos eletrônicos demanda novas formas de gerenciamento para garantir agilidade e atender requisitos jurídicos e arquivísticos. Muitas organizações e/ou instituições, na gestão de seus documentos, possuem sistemas informatizados para manipular e manter documentos de forma eletrônica, sem a necessidade da cópia física. Dessa forma, espaços como contêineres de documentos, armários, gavetas, etc. são liberados, além do próprio consumo de papel.

Um sistema informatizado de documentos eletrônicos, além de liberar espaço físico, permite que usuários possam consultar sobre assuntos, pessoas envolvidas, datas, horas, etc. presentes nos documentos armazenados. Além disso, os gestores têm condições de gerar relatórios estatísticos diversos, como quantas reuniões foram realizadas, quem participou, quais foram as pautas das reuniões, quais as decisões tomadas, por exemplo em documentos do tipo atas.

2.3. Metodologia Scrum

Conforme algumas obras [Wykowski and Wykowska 2019], [Sutherland 2016] e [Pagotto et al. 2016], é consenso que a metodologia tem por prioridade o desenvolvimento de um sistema (produto ou serviço) de modo menos complexo, mais eficiente e focado em resultados. Essas mesmas obras, registram que Scrum melhora a comunicação e motivação do grupo de programadores com as pessoas conhecedoras das regras do negócio ('dono' ou responsáveis), possui um custo baixo, diminui o tempo e o risco do projeto e aumenta a produtividade da equipe. Os encontros (*sprints*) são semanais ou quinzenais e sempre há apresentações das funcionalidades do sistema (*product backlog*)

em forma de protótipo. Isto é, o interessado ou o responsável pelo sistema acompanha e testa toda funcionalidade implementada até aquele instante.

Dentro do contexto deste trabalho, em que a gestão informatizada de documentos eletrônicos, principalmente, atas de reuniões de membros de um curso de graduação, há os papéis do coordenador, secretária, alunos e professores de um curso. Dessa forma, os autores deste trabalho (aluno e orientador) foram os responsáveis por todo o processo de criação, desenvolvimento e testes do sistema. O delineamento e as definições de aspectos funcionais (planejamento, execução, revisão e retrospectiva do *sprint*) do sistema, os *layouts* e a dinâmica de funcionamento geral, além de fornecer todo o conteúdo relacionado ao procedimento de geração de atas, ficaram sob responsabilidade da coordenação e da secretaria. O orientador do trabalho comportou-se como *scrum master*, que gerenciou o desenvolvimento como um todo (prazos, metas, responsáveis, controle de versões do sistema, reuniões, etc). E a secretaria e a coordenação foram os donos do produto (*product owner*).

Portanto, na metodologia Scrum é possível dividi-la:

- papéis
 - *Scrum master* (gestor);
 - *Product owner* ('dono' ou responsável);
 - *Dev team* (equipe de desenvolvimento).
- eventos
 - Planejamento de *sprint*;
 - Execução do *sprint*;
 - *Daily Scrum* (encontro diário da equipe de desenvolvimento);
 - Revisão *sprint*;
 - Retrospectiva *sprint*.
- artefatos
 - *Product backlog*;
 - *Sprint backlog*;
 - Incremento ou entrega.

Destaca-se que o desenvolvimento do sistema foi baseado na programação em par, em que orientador e o aluno programavam/implementavam juntos nos encontros de orientação. Como o trabalho obedeceu as boas práticas de Scrum, baseada em lista de funcionalidades por *sprint*, foi possível finalizar o sistema em menos de 15 semanas.

2.4. Frameworks e Versionamento de código

Dentro das boas práticas de projeto e desenvolvimento de sistemas, há o de versionamento de código associado a *frameworks*. Um *framework* para desenvolvimento de software é uma abstração que fornece funcionalidades genéricas para reutilização de código para partes do sistema em desenvolvimento. Isso ocorre por meio de pacotes, classes, interfaces, métodos, entre outros. Ao contrário das bibliotecas, é o *framework* que assume o fluxo de controle da aplicação [Freeman and Freeman 2009]. Destaca-se que padrões de projeto de software não podem ser confundidos com *frameworks*, pois padrões possuem um nível maior de abstração. *Framework* inclui código, diferentemente de um padrão de projeto, e pode ser modelado com vários padrões de projeto. Além disso, sempre possuem um domínio de uma aplicação particular, algo que não ocorre nos padrões de projeto

de software [Freeman and Freeman 2009]. As principais vantagens de usar *frameworks* são: maior facilidade para a detecção de erros; concentração na abstração de soluções do problema; eficiência na resolução dos problemas e otimização de recursos.

Um sistema de controle de versões de código (ou versionamento) é um software que tem como objetivo gerenciar as diferentes versões de arquivos (programas ou não) em um sistema. O versionamento deve manter histórico de alteração dos arquivos fontes e também da documentação produzidos e/ou alterados, controlando conflitos, produtividade, status das versões. As vantagens do uso de versionamento, por exemplo, são qualidade no trabalho em equipe (várias pessoas trabalhando sobre um mesmo sistema sem problemas de conflitos); resgate de versões; ramificação do projeto (divisão do projeto em várias linhas/ramos de desenvolvimento, que podem ser trabalhadas paralelamente, sem que uma interfira na outra); segurança (controle de invasão e criptografia); backup e confiança (armazenamento em nuvem de arquivos, por exemplo) [Molinari 2007].

2.4.1. Git, Bootstrap e Django

O sistema de controle de versão utilizado neste trabalho é o Git, associado ao serviço Web GitHub¹. Essa combinação é baseada em software livre e é distribuída em versões Linux, Windows e MacOS. Podem ser utilizadas diretamente na Web, via terminal (console), ou como *plugin* de ambientes de desenvolvimento como Eclipse, Netbeans, Visual Studio Code, Visual Studio, por exemplo. A Figura 1 mostra o GitHub do projeto, em que é possível visualizar as datas das atualizações, quem atualizou e em que parte do sistema.

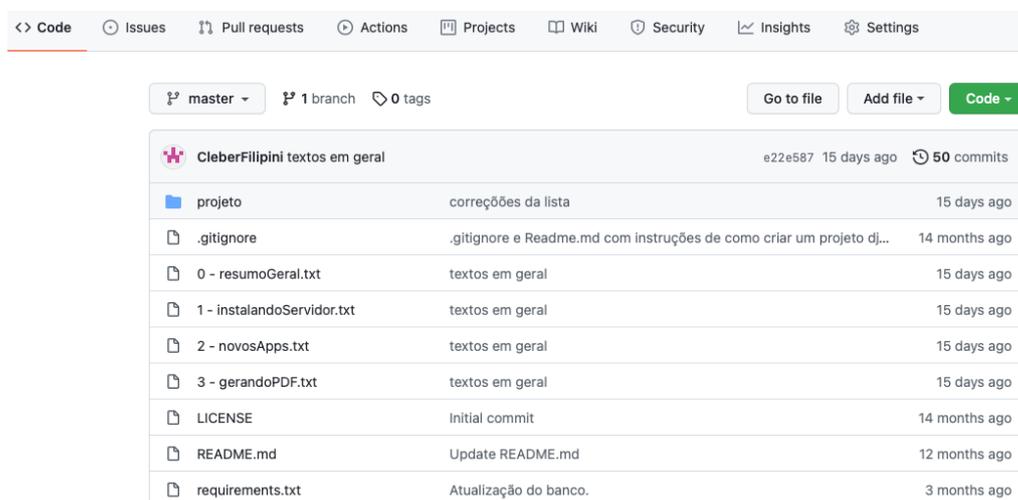


Figura 1. Site do GitHub com dados do projeto do sistema para compartilhamento.

Um dos *frameworks* mais utilizados para construção de sistemas Web na parte de exibição e visualização é o Bootstrap². Bootstrap, conforme o site da W3School, é o mais popular *framework* que trabalha com HTML (linguagem de marcação de hipertexto), CSS (folha de estilo em cascatas, utilizada com a linguagem de marcação) e JavaScript

¹<http://github.com>.

²<https://www.w3schools.com/bootstrap/>.

no desenvolvimento de sistemas Web-Mobile responsivos. Além disso, é gratuito e fácil de incorporar ao sistema em construção.

O *framework* Django³ está associado à linguagem de Programação Python e garante prototipação e entregas rápidas (modelo para construção de sistemas Web). Ele fornece inúmeros modelos de design, *drivers* de conexão com bancos de dados, de processos de validação (como *login* e senhas), de aplicativos com CRUD (*Create, Retrieve, Update, Delete*) completo. O padrão arquitetural utilizado no *framework* Django é o *Model View Template* (MVT). O *Model* é a interface com o banco de dados, cada classe compara-se a uma tabela do banco de dados, e as instâncias das classes representam os registros das tabelas. Para o mapeamento dos dados, não é necessário utilizar códigos na linguagem SQL para garantir a persistência dos dados no banco, pois há o recurso *Object-Relational Mapping* (ORM ou Mapeamento Objeto-Relacional). A camada *View* é responsável pela implementação das regras de apresentação e negócio do sistema. É por meio dela que se faz a interligação entre *Model* e *Template*. Por fim, *Template* é a camada que retorna a visão para o usuário do programa. Essa camada é composta por HTML, CSS, Javascript. O Django oferece uma interface com o banco de dados que permite ao desenvolvedor não se preocupar com a conexão entre suas classes de domínio e banco [dos Santos et al. 2021]. Na Figura 2, pode-se perceber como as pastas do projeto do sistema ficaram organizadas conforme a orientação do modelo MVT.

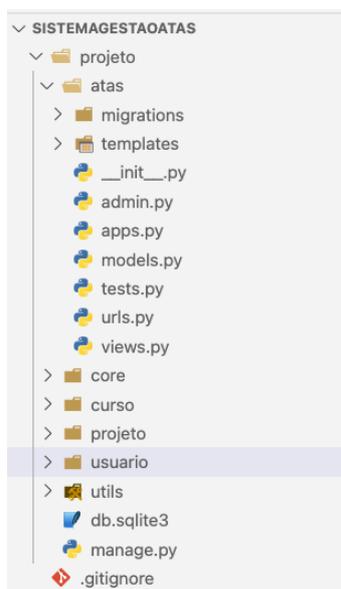


Figura 2. Organização das pastas do projeto de acordo com o modelo MVT.

Enfatiza-se que *frameworks* consolidados como Bootstrap e Django têm em seus sites uma documentação completa e fóruns de discussão que auxiliam em seus usos.

2.5. Trabalhos relacionados

Nesta subseção, buscou-se apresentar e discutir trabalhos com tecnologias ou metodologias que pudessem dar suporte para esta pesquisa.

³<https://www.djangoproject.com>.

Na pesquisa realizada por [Bax et al. 2011], foi construído e analisado um estudo de caso para gerenciar documentos eletrônicos via PLONE. Essa gestão buscou garantir agilidade e atender requisitos jurídicos e arquivísticos para a uma empresa de jornalismo, que em seu processo de trabalho apresenta necessidade de automatização do contínuo fluxo de interação autor-editor. O sistema construído funciona na Web com apoio do PLONE, que é um *framework* voltado à construção de portais corporativos em software livre, de forma integrada com o aplicativos da suíte MS Office para Sistemas Operacionais MS Windows.

Já no trabalho realizado em [Palharini et al. 2020], foi projetado, implementado e disponibilizado um sistema Web de gestão informatizada de Trabalhos Finais de Graduação, para agilizar os processos de submissão, avaliação e emissão de pareceres, além de promover mecanismos de busca de assuntos, tecnologias, metodologias trabalhadas, ao longo dos anos, em pesquisas conduzidas por alunos e seus orientadores. O sistema funciona como um serviço Web e foi implementado usando as tecnologias *frameworks* Bootstrap e Django que estão associados à linguagem Python.

No trabalho realizado por [Zamberlan et al. 2021], foi mostrado uma experiência extensionista em disciplinas dos cursos de Ciência da Computação e de Sistemas de Informação. O relato foi centrado nos processos de projeto e implementação de um sistema Web, realizado por alunos das disciplinas, que atendeu uma demanda da comunidade interna. O sistema foi um protótipo em forma de uma ferramenta para gestão colaborativa de documentos. A experiência ocorreu no segundo semestre de 2020, em duas disciplinas, com total de 35 alunos, coordenadores de curso e contou com a consultoria de profissional de uma empresa de tecnologia. Nesse trabalho, os requisitos funcionais, as principais funcionalidades e o protótipo inicial foram construídos usando a tecnologia Python-Django.

Destaca-se que os três trabalhos foram importantes. O primeiro por tratar especificamente de Gestão Eletrônica de Documentos, dando amparo conceitual e legal a essa proposta. O segundo trabalho relacionado forneceu toda a metodologia e as tecnologias, uma vez que o sistema projetado e construído neste trabalho também obedeceu a mesma arquitetura e dinâmica de funcionamento. E o último trabalho, trouxe todas as regras do negócio necessárias ao bom funcionamento de um sistema para gestão de documentos eletrônicos.

3. Proposta de trabalho

Este trabalho busca auxiliar na gestão informatizada de documentos, uma vez que a legislação brasileira têm facilitado às organizações e/ou instituições em manipular documentos na sua forma eletrônica, em detrimento de sua forma física, liberando espaços e auxiliando em consultas diversas a conteúdos presentes nos documentos armazenados. O sistema projetado e construído gerencia atas de reuniões de cursos de graduação e pós-graduação de instituições de ensino. O sistema funciona na Web como um serviço, em que usuários (coordenadores, secretárias, professores e alunos) acessam para gerar reuniões, ou visualizar, ou imprimir e/ou validar atas.

O sistema de gestão de atas começou como uma atividade em sala de aula, nos cursos de Ciência da Computação e Sistemas de Informação, em disciplinas optativas de programação e de caráter extensionista. Nas disciplinas, foram trabalhados conceitos de programação Python-Django, boas práticas no versionamento de código, gestão

de tarefas, entre outras. Durante as disciplinas, os estudantes modelaram e projetaram algumas funcionalidades do sistema, que serviram de alicerce para este trabalho [Zamberlan et al. 2021]. Dessa forma, o sistema proposto, implementado e entregue pôde incrementar funcionalidades e fornecer outras tantas que garantissem as funcionalidades desejadas em sistema de gestão de atas.

3.1. Materiais e métodos

Este trabalho foi baseado em pesquisa exploratória com revisão bibliográfica e amparado com estudo de caso. Já no projeto e desenvolvimento da solução, foram utilizados a metodologia *Scrum* e a técnica *Kanban*.

As ferramentas utilizadas foram:

- Trello: ferramenta online para operacionalizar a técnica Kanban, que gerencia o cronograma de atividades deste trabalho;
- Astah: ferramenta gratuita para a diagramação de aspectos funcionais e estruturais da solução proposta;
- Linguagem Python 3.8, *framework* Django 3.1.5 e o ambiente de desenvolvimento Visual Studio Code;
- Ambiente virtual *virtualenv*;
- Pacotes Python-Django (*requirements.txt*): *django-bootstrap3*, *django-easy-pdf*, *unicorn* (Python WSGI HTTP Server), *nginx* (servidor HTTP, POP, IMAP e de *proxy* reverso), entre outros;
- Bancos de dados SQLite (máquina local) e MySQL (máquina servidor).

3.2. Modelagem e projeto do sistema

Na Figura 3, é possível visualizar os principais atores, as funcionalidades do sistema, bem como as suas relações com esses atores. A Figura 4 mostra a dinâmica de geração de uma reunião (convocação/convite) até a geração/validação da ata da reunião.

Em relação aos aspectos estruturais do projeto, a Figura 5 ilustra, ao mesmo tempo, as classes do sistema e as tabelas do banco de dados, uma vez que Django possui a ferramenta de mapeamento objeto-relacional. Dessa forma, toda a modelagem orientada a objetos presente nos arquivos *models.py* é convertida automaticamente em tabelas do banco, para persistência dos dados (*makemigrations* e *migrate*).

Ressalta-se que o diagrama de classe foi gerado a partir dos modelos construídos no sistema. Para isso, foram necessárias algumas instalações e configurações no sistema⁴:

- instalação e configuração dos pacotes *django-extensions* e *graphviz*;
- em *settings.py* do projeto, inserção do pacote *django-extensions* como uma aplicação do sistema;
- execução pelo gerenciador *python-django* (*manage.py*) o comando *graph_models*.

A principal ideia foi gerar um diagrama de classe fiel à estrutura implementada no sistema, facilitando o entendimento do sistema e do processo de mapeamento objeto-relacional.

⁴Há um passo-a-passo em <https://ohmycode.com.br/gerando-o-diagrama-de-classes-do-seu-projeto-django>.

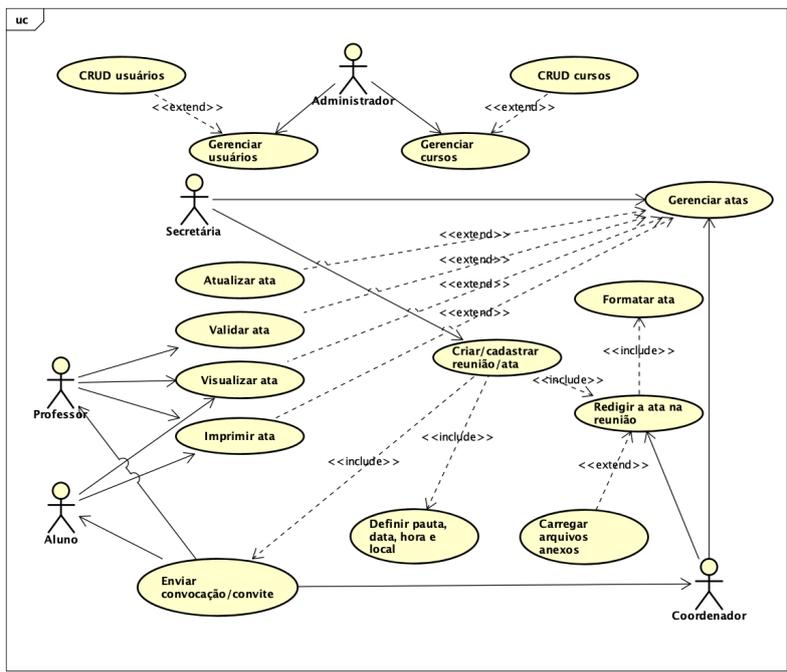


Figura 3. Diagrama de Casos de Uso Sistema Online de Gestão de Atas para coordenação e secretaria de cursos de graduação e pós-graduação.

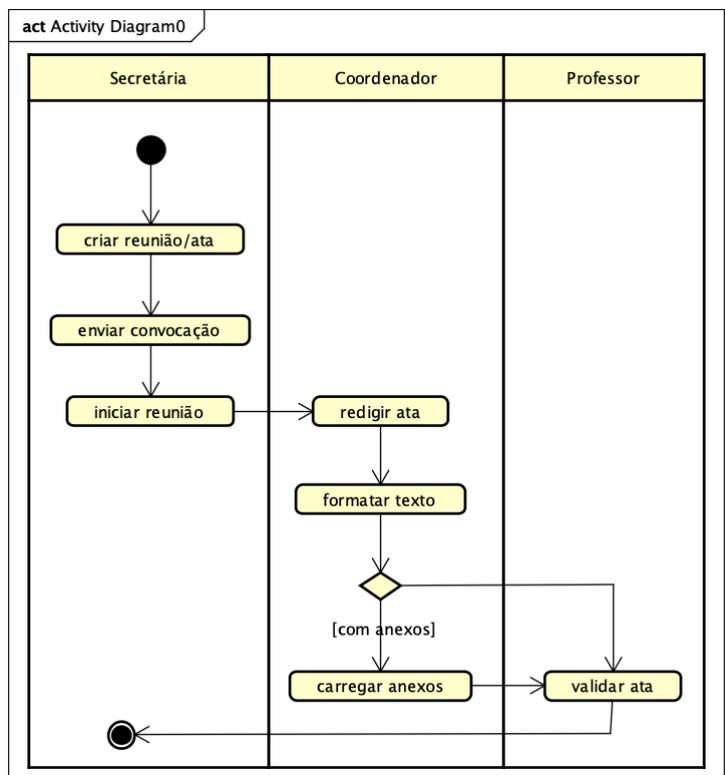


Figura 4. Diagrama de Atividades ilustrando o processo de criação de reunião, logo ata, até sua validação por professores.

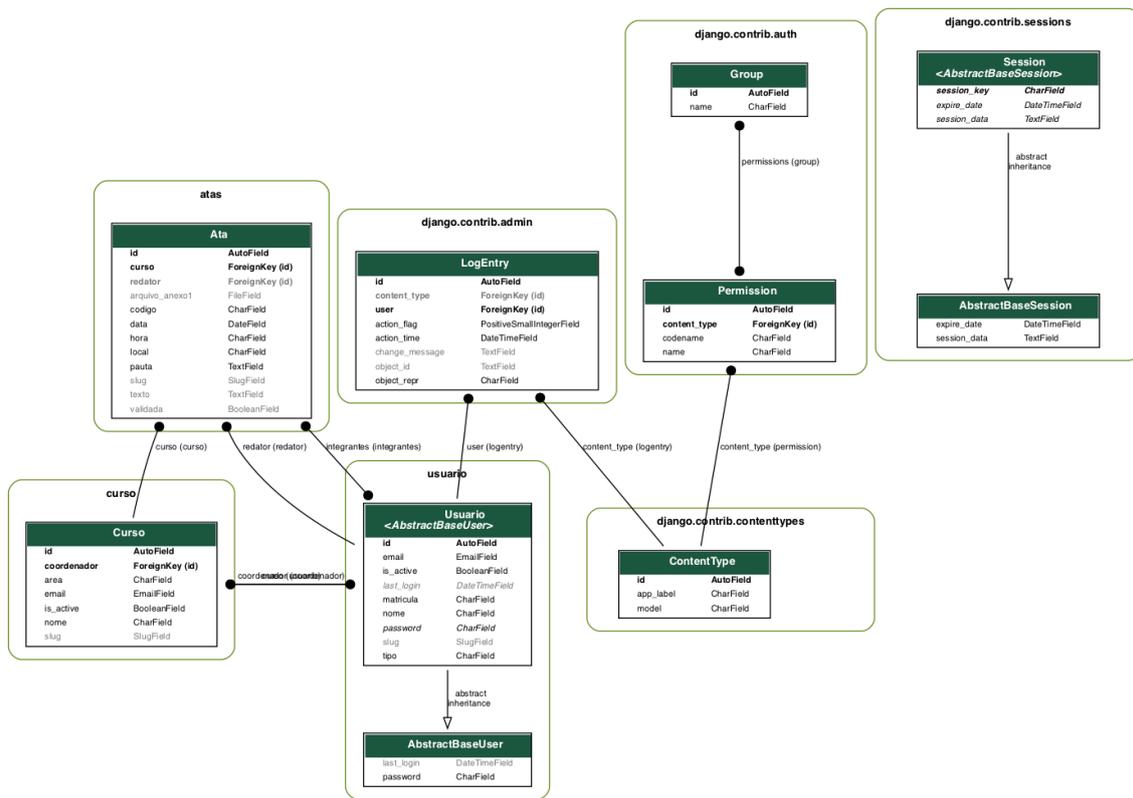


Figura 5. Diagrama de Classes.

3.3. Interfaces gráficas do sistema

Nas Figuras (6, 7, 8, 9 e 10) desta seção, há exemplos das principais funcionalidades do sistema, na visão do Administrador. No sistema, há diferentes tipos de usuários (secretária, professor e aluno) com diferentes processos permitidos.

A Figura 6 ilustra o gerenciamento clássico (CRUD) de cursos: cadastro, recuperação, atualização e deleção. Um curso tem nome, área, coordenador responsável e email.

Nome	Área	Coordenador	Email	Ativo	Editar	Excluir
ARQUITETURA E URBANISMO	CIÊNCIAS TECNOLÓGICAS	12353 - FRANCISCO LIMA - francisco@ufn.edu.br	arquitetura@ufn.edu.br	✓		
CIÊNCIA DA COMPUTAÇÃO	CIÊNCIAS TECNOLÓGICAS	234234 - REINER PEROZZO - reiner@ufn.edu.br	cc@ufn.edu.br	✓		
SISTEMAS DE INFORMAÇÃO	CIÊNCIAS TECNOLÓGICAS	234 - GUILHERME CHAGAS KURTZ - guilherme@ufn.edu.br	cc@ufn.edu.br	✓		

Figura 6. Relação de cursos cadastrados - gestão de cursos.

Na Figura 7, é possível visualizar a relação de usuários do sistema e realizar toda a gestão dos diferentes tipos de usuários e suas permissões. Nessa interface, o administrador pode acompanhar os acessos de cada usuário, desativá-lo, entre outras operações.

Nome	Tipo	E-mail	Matricula	Curso(s)	Ativo	Último acesso	Editar	Excluir
ALEXANDRE ZAMBERLAN	ADMINISTRADOR	alexz@ufn.edu.br	123456	CIÊNCIA DA COMPUTAÇÃO , SISTEMAS DE INFORMAÇÃO .	✓	11/11/2021 19:53		
FRANCISCO LIMA	PROFESSOR	francisco@ufn.edu.br	12353	ARQUITETURA E URBANISMO .	✓	26/08/2021 21:51		
GUILHERME CHAGAS KURTZ	PROFESSOR	guilherme@ufn.edu.br	234	CIÊNCIA DA COMPUTAÇÃO , SISTEMAS DE INFORMAÇÃO .	✓	28/10/2021 20:35		
LETÍCIA CARVALHO	SECRETÁRIA	leticia@ufn.edu.br	23423	CIÊNCIA DA COMPUTAÇÃO , SISTEMAS DE INFORMAÇÃO .	✓	28/10/2021 20:28		
REINER PEROZZO	ADMINISTRADOR	reiner@ufn.edu.br	234234	CIÊNCIA DA COMPUTAÇÃO , SISTEMAS DE INFORMAÇÃO .	✓			
CLEBER FILIPINI	ADMINISTRADOR	cleber@ufn.edu.br	34535345	Nenhum curso associado	✓	23/09/2021 17:17		

Figura 7. Relação de usuários cadastrados - gestão de usuários.

Na Figura 8, destaca-se a funcionalidade de pesquisar atas por data, curso e por expressões em textos redigidos em reuniões. Esse recurso permite que gestores, professores e/ou avaliadores possam verificar se determinado assunto foi discutido, em que dia, com quais participantes, por exemplo. Nessa interface, é possível cadastrar, atualização, visualizar e imprimir atas.

Curso	Código	Data	Hora	Local	Pauta	Ata validada?	Arquivo Anexo	Visualizar	Editar	Excluir
ARQUITETURA E URBANISMO	1/2021	24/09/2021	10:00	TEAMS	Discussão sobre o novo PPC; Avaliação das ACCs pelo Colegiado.					
ARQUITETURA E URBANISMO	2021/03	28/10/2021	20:30	TEAMS	Finalização do projeto do Cléber					
CIÊNCIA DA COMPUTAÇÃO	2021/09	29/10/2021	10:00	Sala 102 Prédio 2	Aprovação dos gastos da coordenação					

Figura 8. Relação de atas geradas e armazenadas por curso.

A Figura 9 mostra como seria a visualização de uma ata. Essa operação pode ser acessada por qualquer usuário do sistema, seja ele aluno, professor, secretária ou coordenador.

E por fim, na Figura 10, há a geração de arquivo *.pdf* com a impressão da ata. Note que o cabeçalho da impressão não segue os padrões clássicos de uma ata, porque foi decidido compilar no cabeçalho alguns dados que facilitasse o entendimento dos assuntos da ata (pauta), curso, redator, participantes, data e hora.

Curso : SISTEMAS DE INFORMAÇÃO Ata número: 03/2021	
Data e hora:	08/12/2021 às 17:00
Local:	Conjunto 3
Pauta:	Encerramento do semestre
Redator:	111111 - CLEBER FILIPINI - filipini.cleber@ufn.edu.br
Integrantes:	CLEBER FILIPINI , TESTE TFG , ALEXANDRE ZAMBERLAN , RICARDO FROHLICH DA SILVA .
Validada?	Sim
Texto e anexo	
<p>O sistema foi uma continuidade em uma atividade iniciada em sala de aula com caráter extensionista. E, justamente, por se tratar de uma ação extensionista dentro da instituição, o sistema merecia cuidados especiais, no que se refere a funcionalidades importantes e desejadas a um sistema de gestão de documentos que deve ser utilizado em coordenações de cursos de Graduação e Pós-graduação. Ademais, o sistema já está disponível em https://atas.lapinf.ufn.edu.br. Todo o processo de construção obedeceu boas práticas da metodologia Scrum e de versionamento de código, gerando um sistema modular e de fácil manutenção (ou incremento). Em relação a avaliação do sistema, isso ocorreu durante a disciplina extensionista, em que nela, protótipos foram implementados e enviados à avaliação (secretaria e coordenação). E devido a essas avaliações, surgiram os apontamentos que deveriam ser melhorados na ferramenta. Dessa forma, implementados neste trabalho.</p>	
Arquivo em Anexo	
	

Figura 9. Visualização de uma ata em específico.

Gestão de Atas Online UFN

Data e hora: 12/03/2021 às 14:00	Local: TEAMS	Ata APROVADA pelos participantes
Pauta: Colegiado de curso para aprovação de ACCs.		
Redator: ALEXANDRE ZAMBERLAN		
Participantes da reunião: ALEXANDRE ZAMBERLAN , REINER FRANTHESCO PEROZZO , GUILHERME KURTZ , RICARDO FROHLICH DA SILVA .		

Ata 01/2021 de reunião do curso CIÊNCIA DA COMPUTAÇÃO

1. O sistema foi uma continuidade em uma atividade iniciada em sala de aula com caráter
2. extensionista. E, justamente, por se tratar de uma ação extensionista dentro da instituição, o
3. sistema merecia cuidados especiais, no que se refere a funcionalidades importantes e desejadas a
4. um sistema de gestão de documentos que deve ser utilizado em coordenações de cursos de Graduação
5. e Pós-graduação. Ademais, o sistema já está disponível em <https://atas.lapinf.ufn.edu.br>. Todo o
6. processo de construção obedeceu boas práticas da metodologia Scrum e de versionamento de código,
7. gerando um sistema modular e de fácil manutenção (ou incremento). Em relação a avaliação do
8. sistema, isso ocorreu durante a disciplina extensionista, em que nela, protótipos foram
9. implementados e enviados à avaliação (secretaria e coordenação). E devido a essas avaliações,
10. surgiram os apontamentos que deveriam ser melhorados na ferramenta. Dessa forma, implementados
11. neste trabalho.

Figura 10. Impressão de uma ata em específico.

3.4. Instalação e configuração de serviços

Esta seção busca apresentar algumas informações necessárias de instalação e configuração de serviços e/ou recursos para facilitar e reprodução deste trabalho.

Em relação ao processo de geração do arquivo *.pdf* (Figura 10) de uma ata, há alguns destaques:

- instalação na *venv* (ambiente virtual do sistema) dos pacotes: *django-easy-pdf* e *django-utils-six*;
- no arquivo *settings.py*, adição de *easy-pdf* na variável de ambiente *INSTALLED_APPS*;

- no *app* desejado para ter geração de arquivos *.pdf*, alteração dos códigos fontes de *models.py*, *urls.py* e *views.py*.

Dessa forma, no arquivo *models.py*:

```
@property
def get_relatorio_url(self):
    return reverse('ata_report', args=[str(self.id)])
```

No arquivo *urls.py*:

```
url(r'(?P<pk>\d+)/relatorio/\$', AtaRelatorioView.as_view(), name="ata_report"),
```

E no arquivo *views.py*, importar do pacote *easy_pdf* a classe *PDFTemplateResponseMixin*. Em seguida, criar a classe *AtaRelatorioView*:

```
class AtaRelatorioView(LoginRequiredMixin,
PDFTemplateResponseMixin, DetailView):
    model = Ata
    template_name = 'atas/relatorios/ata.html'
```

Finalmente, uma vez construído o sistema, ele deve ser disponibilizado em um servidor, que neste caso foi uma máquina virtual com 4 Gbytes de memória RAM e 1 Tbytes de memória secundária, sistema operacional Ubuntu 18.04, com os serviços nginx, gunicorn e mysqlclient. O sistema foi colocado em produção, ou seja, colocado em funcionamento via acesso remoto terminal com *ssh*, com o projeto clonado do GitHub, com a instalação e configuração do ambiente virtual *venv*.

No serviço do gunicorn, foi necessário criar o arquivo:

```
#/etc/systemd/system/nomeProjeto.service
[Unit]
Description=nomeProjeto daemon
After=network.target
[Service]
User=usuarioServidor
Group=www-data
WorkingDirectory=pastaServidor/nomeProjeto/projeto
ExecStart=pastaServidor/nomeProjeto/venv/bin/gunicorn --access-logfile -
--workers 3 --bind
unix:pastaServidor/nomeProjeto/projeto/nomeProjeto.sock projeto.wsgi:application
```

Para iniciar o *parser* do gunicorn foram necessários dois passos:

- * Passo 1 - iniciar o *parser* gunicorn:

```
sudo systemctl start nomeProjeto
```

- * Passo 2 - adicionar o sistema como serviço reconhecido no sistema operacional:

```
sudo systemctl enable nomeProjeto
```

Caso aconteça problemas na geração do *.sock* do serviço e o *.service* precise ser atualizado, há necessidade de rodar o código que segue e executar novamente passos 1 e 2.:

```
sudo systemctl daemon-reload
```

No serviço nginx, também foi necessário criar o arquivo:

```

#/etc/nginx/sites-available/site-do-sistema.com.br
server {
    server_name site-do-sistema.com.br www.site-do-sistema.com.br;
    client_body_in_file_only clean;
    client_body_buffer_size 32K;
    client_max_body_size 2512M;
    #aceitar arquivos grande via get
    client_header_buffer_size 3232k;
    large_client_header_buffers 4 3232k;
    sendfile on;
    send_timeout 900s;
    location = /favicon.ico { access_log off; log_not_found off; }
    location /static {
        expires 30d;
        access_log off;
        break;
        alias /var/www/site-do-sistema.com.br/nomeProjeto/projeto/projeto/static;
    }
    location / {
        client_max_body_size 2048M;
        include proxy_params;
        proxy_pass http://unix:/var/www/site-do-sistema.com.br/nomeProjeto/nomeProjeto.sock;
    }
}

```

A seguir, foi necessário ativar o arquivo através da criação de um link simbólico dele para o diretório */etc/nginx/sites-enabled*, que o serviço nginx lê durante a inicialização:

```
sudo ln -s /etc/nginx/sites-available/exemplo.com /etc/nginx/sites-enabled/
```

Depois, foi preciso certificar-se de que não existiam erros de sintaxe em quaisquer de seus arquivos do nginx, via o comando:

```
sudo nginx -t
```

Se não houve problemas, reinicie o nginx para ativar suas alterações, novamente por comando:

```
sudo systemctl restart nginx
```

4. Conclusões

O trabalho está no contexto de Sistemas de Informação Gerencial, focado na gestão eletrônica de documentos via um sistema Web de uso colaborativo.

Ao longo do texto, conceitos sobre Sistemas de Informação, Gestão de Documentos, Boas Práticas de Desenvolvimento de Sistemas foram apresentados. Também, três pesquisas relacionadas foram analisadas e que auxiliaram na construção do sistema proposto.

O sistema foi uma continuidade em uma atividade iniciada em sala de aula com caráter extensionista. E, justamente, por se tratar de uma ação extensionista dentro da instituição, o sistema merecia cuidados especiais, no que se refere a funcionalidades importantes e desejadas a um sistema de gestão de documentos que deve ser utilizado em coordenações de cursos de Graduação e Pós-graduação. Ademais, o sistema já está disponível em <https://atas.lapinf.ufn.edu.br>.

Todo o processo de construção obedeceu boas práticas da metodologia Scrum e de versionamento de código, gerando um sistema modular e de fácil manutenção (ou incremento). Em relação a avaliação do sistema, isso ocorreu durante a disciplina extensionista, em que nela, protótipos foram implementados e enviados à avaliação (secretaria e coordenação). E devido a essas avaliações, surgiram os apontamentos que deveriam ser melhorados na ferramenta. Dessa forma, implementados neste trabalho: pesquisa por conteúdo nas atas, acesso com permissões distintas aos usuários, impressão da ata com numeração das linhas, validações de campos como CPF e email, entre outros.

Finalmente, há trabalhos futuros no sistema:

- quando um usuário estiver logado como secretária, ao cadastrar uma ata, o sistema deve mostrar somente os cursos da secretária;
- se um professor for selecionado como redator, ele deve poder editar a ata;
- no cadastrar uma ata, mostrar na lista de redatores, somente professores do curso;
- quando uma ata for validada, também validar a participação dos membros da reunião por assinatura digital.

Referências

- Baldam, R. (2017). *Gerenciamento de Conteúdo Empresarial: ECM-Enterprise Content Management*. Elsevier Brasil.
- Bax, M. P., de Oliveira, J. L. R., and Barbosa, D. M. (2011). Gerenciamento de documentos eletrônicos: Estudo de caso com o sistema de gestão de conteúdo plone. *Perspectivas em Gestão & Conhecimento*, 1(2):166–190.
- dos Santos, R. E., de Oliveira Zamberlan, A., Vieira, S. A. G., Kurtz, G. C., and Frohlich, R. (2021). Proposta de uma plataforma de cloud computing para disponibilização de um sistema online para consultórios e clínicas por meio do modelo saas. *Tópicos em Ciências da Saúde Volume 24*, page 7.
- ESMEC (2020). Documentos eletrônicos: Escola Superior da Magistratura do estado do Ceará. <https://esmec.tjce.jus.br/wp-content/uploads/2009/05/3-documento-eletronico-esmec.pdf>. Acessado em Outubro de 2021.
- Freeman, E. and Freeman, E. (2009). *Use a Cabeça - Padrões de Projetos*. Alta Books Editora, 2nd edition.
- Laudon, J. P. and Laudon, K. C. (2014). *Sistemas de Informação Gerenciais*. Pearson, 11 edition.
- Ministério Público Federal (2021). O que é LGPD. <http://www.mpf.mp.br/servicos/lgpd/o-que-e-a-lgpd>. Acessado em Outubro de 2021.
- Molinari, L. (2007). Gerência de configuração-técnicas e práticas no desenvolvimento do software. *Florianópolis: Visual Book*.
- O'brien, J. A. and Marakas, G. M. (2013). *Administração de sistemas de informação*. AMGH, São Paulo.
- Pagotto, T., Fabri, J. A., Lerario, A., and Gonçalves, J. A. (2016). Scrum solo: software process for individual development. In *2016 11th Iberian Conference on Information Systems and Technologies (CISTI)*, pages 1–6. IEEE.

- Palharini, E. P., dos Santos, R. E., Canal, A. P., da Silva, R. F., Vieira, S. G., and Zamberlan, A. (2020). Sistema web para submissão e avaliação de trabalhos finais de graduação. *Simpósio de Ensino, Pesquisa e Extensão da UFN*, 24(1):1–10.
- Sangiovo, E., Martins, M., da Silva, R. F., and Zamberlan, A. (2018). Um estudo sobre baas: aplicativo para solicitações de orçamentos. Trabalho final de graduação, Universidade Franciscana.
- Sutherland, J. (2016). *Scrum: a arte de fazer o dobro do trabalho na metade do tempo*. Leya.
- Wykowski, T. and Wykowska, J. (2019). Lessons learned: Using scrum in non-technical teams.
- Zamberlan, A., dos Santos, R. E., Kurtz, G., and Perozzo, R. F. (2021). Sistema Web para gestão de documentos atas construído por muitas mãos: experiência extensionista na Computação. *Disciplinarum Scientia— Ciências Humanas*, 23(1):1–12.