

Módulo SISGEP-COMIC

Gestão de Avaliação de Projetos

Patrick Tolio Müller¹, Alexandre de Oliveira Zamberlan¹

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

patricktmuller@gmail.com, alexz@ufn.edu.br

Abstract. *This work presents the modeling and development of a Health Committee Project Management System module. The module was designed and implemented through the Scrum Solo methodology, Kanban activity management technique (Trello tool), Python programming language, Django and Bootstrap frameworks, and Bizagi and Astah modeling tools. The module will assist in project evaluation processes with Health Committee.*

Resumo. *Este trabalho apresenta a modelagem e o desenvolvimento de um módulo do Sistema de Gestão de Projetos da Comissão Científica (COMIC) dos Hospitais Casa de Saúde e São Francisco de Assis (SISGEP-COMIC). O módulo foi construído por meio da metodologia Scrum Solo, técnica de gestão de atividades Kanban (ferramenta Trello), linguagem de programação Python, frameworks Django e Bootstrap e ferramentas de modelagem Bizagi e Astah. O módulo disponibilizado auxilia nos processos de avaliação de projetos junto ao COMIC.*

1. Introdução

Sistemas automatizados que gerenciam e fornecem suporte a projetos¹ em instituições são ferramentas no contexto de Sistemas de Informação, pois agilizam os fluxos de submissão, avaliação e acompanhamento das atividades pretendidas e dos objetivos propostos. A gestão desses projetos em ferramentas que funcionem na Internet agiliza significativamente as submissões e as avaliações, principalmente. Um vez com dados e/ou informações armazenadas na base do sistema, é possível realizar questões que podem auxiliar nas tomadas de decisões, como por exemplo: qual o hospital com o maior número de projetos; qual o pesquisador que mais participa de editais; qual as áreas de interesse em um determinado hospital. Portanto, este trabalho final de graduação (TFG) está inserido em projeto do Laboratório de Práticas (Ciência da Computação e Sistemas de Informação) junto à Comissão Científica dos Hospitais Casa de Saúde e São Francisco de Assis.

O sistema em desenvolvimento é um software de gestão de projetos que trata dos processos de submissão, avaliação e acompanhamento de projetos junto à Comissão Científica (COMIC²), nominado como SISGEP-COMIC. Neste estudo,

¹Descrição escrita e detalhada de um empreendimento a ser realizado; plano, delineamento, esquema [Da Cunha 2019].

²<http://comic.lapinf.com.br>

o módulo de avaliação de projetos foi modelado, implementado e inserido no sistema. Destaca-se que todo o sistema é para computação em nuvem (*Cloud Computing*). O sistema SISGEP-COMIC é considerado um Sistema de Informação Gerencial [O'brien and Marakas 2013].

O objetivo geral do trabalho é modelar e implementar um módulo do sistema de SISGEP-COMIC para avaliar, via critérios e por avaliadores cadastrados, um projeto, respondendo questões dos administradores, por meio de relatórios.

Como objetivos específicos:

- estudar a modelagem do SISGEP-COMIC (aspectos estruturais e funcionais) para as adequações com a inserção do módulo;
- adaptar o modelo Entidade-Relacionamento (ER) e o banco de dados do SISGEP-COMIC para suportar o módulo proposto;
- adaptar o modelo de processo do SISGEP-COMIC;
- adaptar os diagramas estruturais e funcionais do SISGEP-COMIC;
- pesquisar e testar bibliotecas de geração de relatórios.

Este módulo é relevante para a gestão dos administradores COMIC, pois por meio dele, é possível identificar projetos fora dos critérios essenciais para se ocupar dos hospitais escola da UFN. Além disso, a possibilidade do administrador/gestor identificar pesquisadores que tiveram problema ou sucesso em seus projetos. Acredita-se que esses questionamentos são respondidos pela gestão de relatórios e filtros diversos, criados pelo administrador do SISGEP-COMIC.

2. Revisão bibliográfica

Nesta seção, são discutidos os principais assuntos, conceitos, metodologias e técnicas utilizadas na construção deste Trabalho Final de Graduação.

2.1. Sistema de Informação

Sistema de informação surgiu como uma subdisciplina de Ciência da Computação, e está relacionado a um conjunto de elementos manuais ou automatizados. Todas as pessoas que utilizam aplicativos móveis, ingresso em uma rede social, leitura de uma notícia são denominados usuários e estão enviando e recebendo informação por meio de um sistema de informação. Os processos que representam um sistema de informação, são responsáveis pela coleta, processo, transmissão e armazenamento dos dados úteis para desenvolvimento de produtos ou serviços [Laudon and Laudon 2014].

No contexto empresarial, os sistemas de informação podem ser classificados por diversas formas. Um Sistema de Informação Gerencial (SIG), na maioria das vezes, está relacionado a algum software visando um sistema de apoio à decisão ou um sistema especialista, que disponibilize de forma segura, precisa e imediata informações [Laudon and Laudon 2014]. Ele garante o armazenamento de dados e informações, bem como suas relações em relatórios eventuais.

Sistemas de Apoio a Decisão (SAD), também conhecido como *Decision Support Systems* (DSS), surgiu na década de 70 em virtude da elevada competição entre as organizações, necessidade de agilidade nas tomadas de decisões entre outros fatores. SAD possuem especificamente funções que não estão vinculadas diretamente aos

sistemas de informação gerencial. Os SAD permitem realizar buscas de informação em bases de dados existentes tendo por objetivo retirar a contribuição para o processo de decisão. O processo de tomada de decisão está associado com a interação do usuário com um ambiente de negócio da empresa, de forma que ajude nas melhores escolhas e/ou ações [Laudon and Laudon 2014]. Em geral, um SAD responde perguntas de administradores ou gestores, com o intuito de se descobrir algumas anomalias, ou padrões, ou incoerências.

Sistemas de Informação Gerencial auxiliam na gestão das informações manuseadas, via processos de CRUD (*Create, Retrieve, Update, Delete*) com armazenamento em bases de dados. Já os Sistemas de Apoio à Decisão auxiliam nas escolhas das melhores decisões. Por exemplo, um SIG armazena dados de editais, de pesquisadores, de projetos, de áreas de pesquisa, os históricos de submissão de projetos por um professor-pesquisador. Nele, é possível acompanhar as submissões de projetos por edital. Enquanto que o SAD descobre qual hospital recebe a maior submissão de projetos e em que sub-área da Saúde. Então, o SAD vai ajudar na decisão de ofertar mais editais para este hospital.

2.2. Metodologias de projeto e de desenvolvimento de sistemas

Em geral, as metodologias de projeto e desenvolvimento de sistemas possuem etapas (e sub-etapas), boas práticas (recomendações) e resultados pontuais, como artefatos do sistema (diagramas ou protótipos). As metodologias organizam as atividades e planos com seus prazos, responsáveis e equipes. Elas podem ser guiadas pelos aspectos estruturais como base de dados (metodologias mais tradicionais) ou por aspectos funcionais como a lista de funcionalidades do sistema como um todo (metodologias mais atuais) [McLaughlin et al. 2007] e [Pressman 2010].

Scrum é uma metodologia ágil em forma de *framework*³ para gerenciar atividades, pessoas e entregas de soluções em diversas áreas, principalmente na área de desenvolvimento de software [Wykowski and Wykowska 2019]. É uma metodologia projetada para equipes pequenas (até 10 pessoas), que devem dividir suas ações ou tarefas em metas que possam ser completadas iterativamente em 2 semanas (*sprints*), para que o avanço ou progresso possa ser replanejado em reuniões de 15 minutos (*scrums* diários).

Um princípio fundamental do Scrum é o reconhecimento de que os clientes podem mudar de ideia sobre o que desejam ou precisam (volatilidade de requisitos) e que haverá desafios imprevisíveis [Sutherland 2016]. Dessa forma, o Scrum adota uma abordagem empírica baseada em evidências (aceitando que o problema não pode ser totalmente compreendido com antecedência), e foca em maximizar a capacidade da equipe de entregar rapidamente, responder a requisitos emergentes e adaptar-se à evolução do ambiente a ser informatizado [Sutherland 2016]

O Scrum Solo é uma adaptação do Scrum, e a finalidade é o desenvolvimento de processo com apenas um programador e não equipe. Auxilia o programador com o planejamento das atividades para posterior execução sem precisar altos investimentos [Wykowski and Wykowska 2019]. Essa adaptação é uma versão do Scrum, que utiliza alguns de seus princípios como as reuniões para *sprint backlog*, onde os requisitos selecionados para um *log* são trabalhados semanalmente, com a participação do aluno e

³Conjunto de regras, ideias, recomendações, ou modelos que ajudam a decidir o que e como fazer operações/métodos em projetos ou atividades [Da Cunha 2019]

do professor orientador. O *product backlog* representa os requisitos necessários para o software, os quais podem ser alterados ao longo do projeto (acordados entre todos nas reuniões). As retrospectivas de *sprints* são realizadas durante as reuniões semanais entre o aluno (programador) com o orientador (scrum master). Em termos práticos, ferramenta kanban e sistema para versionamento de código são elementos fundamentais para operacionalização do Scrum Solo.

2.3. Versionamento de código e *frameworks*

Dentro das boas práticas de projeto e desenvolvimento de sistemas, há o de versionamento de código associado a *frameworks*.

Um sistema de controle de versões de código (ou versionamento) é um software que tem como objetivo gerenciar as diferentes versões no desenvolvimento de um programa ou sistema qualquer. O versionamento deve manter histórico de desenvolvimento dos códigos-fontes e também da documentação produzidos, controlando conflitos, produtividade, status das versões, entre muitos outros [Molinari 2007].

As vantagens do uso de versionamento podem ser inúmeras, como 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].

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.

2.3.1. Git, Django e Bootstrap

O sistema de controle de versão utilizado é o Git associado ao serviço Web GitHub⁴. 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.

⁴<http://github.com>.

Um dos *frameworks* mais utilizados para construção de sistemas Web na parte de exibição e visualização é o Bootstrap⁵. 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 no desenvolvimento de sistemas Web-Mobile responsivos. Além disso, é gratuito e fácil de incorporar ao sistema em construção.

O *framework* Django⁶ que está associado à linguagem de Programação Python e que garante prototipação e entregas rápidas. 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. É um modelo para construção de sistemas Web com uso da linguagem Python.

O padrão arquitetural utilizado no *framework* Django para Python 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. A camada *View* é responsável pela implementação das regras de apresentação e negócio do nosso 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. 2015].

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

2.4. SISGEP-COMIC

Este é um sistema de informação gerencial para gerir projetos científicos submetidos junto Comissão Científica dos Hospitais Casa de Saúde e São Francisco de Assis. Ele está disponível na Web (<http://comic.lapinf.com.br>) e vem sendo desenvolvido no Laboratório de Práticas dos cursos de Computação. O principal foco do SISGEP-COMIC é gerenciar submissões, avaliações e acompanhamento dos projetos científicos vinculados à área da Saúde da Universidade Franciscana (UFN).

A necessidade de projetar e implementar um sistema em nuvem, que auxiliasse na gestão e acompanhamento de projetos científicos desenvolvidos nos hospitais de ensino da Universidade Franciscana, surgiu para que fosse possível dar maior celeridade à avaliação de projetos bem como para que se padronizasse sua respectiva avaliação. O sistema aqui apresentado tem como objetivo central a rastreabilidade das ações tanto por parte dos proponentes quanto dos avaliadores, a fim de que se tenham estatísticas de produção, relatórios de atividades fidedignos. Além disso, que seja possível avaliar qualitativamente os itens de acordo com a Resolução 466/2012 [Brasil 2013] (Projetos de Pesquisa com Seres Humanos). Cabe lembrar que a Comissão Científica dos Hospitais Casa de Saúde e São Francisco de Assis (COMIC) avalia a exequibilidade dos projetos e seu potencial retorno para as Instituições. Desse modo, é importante salientar que após

⁵<https://www.w3schools.com/bootstrap/>.

⁶<https://www.djangoproject.com>.

a aprovação dos projetos, esses ainda necessitam tramitar junto ao respectivo Comitê de Ética da Universidade Franciscana, a fim de que possam vir a ser implementados.

A Figura 1 ilustra a interface de administrador. É nela que o responsável pelo CO-MIC cadastra usuários, gerencia instituições parceiras, gerencia editais, faz os vínculos de avaliações de projetos a avaliadores e, eventualmente, gerencia submissões de projetos com problemas.



Figura 1. Interface principal da Visão do Administrador.

Na Figura 2, é apresentada a interface contendo todas as submissões realizadas por um professor específico (neste caso, usuário Professor Pierre), com os dados: responsável, colaboradores, local execução, instituições parceiras e arquivo Projeto. Nessa interface, o professor pode alterar ou apagar suas submissões enquanto não estiverem em status de avaliação.

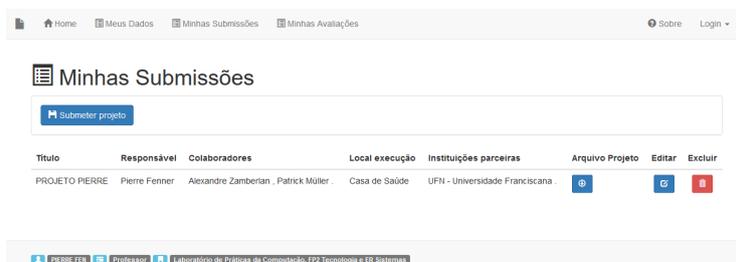


Figura 2. Visão do Professor - listagem de submissão de projetos.

A Figura 3 ilustra parte do processo de cadastro de uma submissão por um professor. O processo solicita que o professor associe um edital, cadastre responsável e opcionalmente colaboradores, informe dados da instituição em que o projeto será executado, área de pesquisa, curso de graduação, pós-graduação, grupo de pesquisa, título, resumo, palavras-chave, instituições parceiras e o arquivo *pdf* do projeto.

A Figura 4 mostra a relação de avaliações que um professor recebe para avaliar. Nela há título do projeto, professores responsável e colaboradores, edital de referência, qual a designação de avaliador (responsável ou suplente), data em que emitiu seu parecer.

Minha Submissão de projeto

Edital
2019/1 - Fluxo contínuo 2019 - Encerramento: 21/12/2019
Campo obrigatório como todos os que tiverem *

Responsavel
Pierre Fenner
Campo obrigatório como todos os que tiverem *

Colaborador
Alexandra Zamberlan
Fernando Sartori Frass
Huanider Andreolla
Patrick Müller
Para selecionar ou deselegionar um colaborador pressione CTRL + Botão Esquerdo do mouse ou Command + Botão Esquerdo do mouse

Nome da instituição de execução do projeto
Casa de Saúde

Área de ensino, pesquisa e extensão de execução do projeto *
Ciências da Saúde

Curso de graduação vinculado *
Ciência da Computação
Nome do curso que está lotado

Curso de pós-graduação em que orienta
Curso de pós-graduação em que orienta
Caso esteja vinculado

Grupo de pesquisa vinculado
Grupo de pesquisa vinculado
Caso esteja vinculado

Título do projeto (200 caracteres) *
PROJETO PIERRE

PIERRE FENNER | Professor | Laboratório de Práticas da Computação, FP2 Tecnologia e ER Sistemas

Figura 3. Visão do Professor - uma submissão.

Home | Meus Dados | Minhas Submissões | Minhas Avaliações | Sobre | Login

Minhas Avaliações

| Projeto a avaliar | Edital | Designação | Dt. Parecer | Parecer |
|---|---|-----------------------|--------------------------------|---------|
| PROJETO DE CIÊNCIA DA COMPUTAÇÃO NA CASA DE SAÚDE - Alexandre Zamberlan - | 2019/1 - Fluxo contínuo 2019 - Encerramento: 21/12/2019 | Avaliador suplente | 11/09/2019 | |
| PROJETO PIERRE - Pierre Fenner - Alexandre Zamberlan, Patrick Müller. | 2019/1 - Fluxo contínuo 2019 - Encerramento: 21/12/2019 | Avaliador responsável | Você ainda não deu seu parecer | |

FERNANDO S | Professor | Laboratório de Práticas da Computação, FP2 Tecnologia e ER Sistemas

Figura 4. Visão do Professor - lista de avaliações delegadas.

2.5. Trabalhos Relacionados

Esta parte do trabalho, elencou-se três trabalhos que tem alguma relação com esta pesquisa. A ideia foi identificar trabalhos que projetaram e implementaram sistemas de informação gerencial para Internet com funcionamento na Web (nuvem), apontando suas metodologias de trabalho e tecnologias utilizadas.

No trabalho ”Proposta de uma plataforma de Cloud Computing para disponibilização de um sistema online para consultórios e clínicas por meio do modelo SaaS”[dos Santos et al. 2015], foi discutido e implementado a virtualização do sistema para computação em nuvem. Como estudo de caso, usou um software baseado na Web projetado para gestão ágil de consultórios e clínicas. Dentre as diversas tecnologias empregadas no desenvolvimento do sistema, destacam-se: Linguagem de programação Python com o framework Django; Node.js como Interpretador de código Javascript no lado do servidor; Sistema gerenciador de banco de dados PostgreSQL; Framework de front-end Bootstrap.

O trabalho realizado por [Sangiovo et al. 2018], foi proposta uma arquitetura Cloud Computing por meio de computação móvel que utiliza a tecnologia como aliada

para a prestação de serviço. O objetivo foi projetar e implementar um aplicativo móvel que gerencie solicitações de orçamentos a prestadores de serviços, por meio do modelo BaaS (*Backend as a Service*). As tecnologias utilizadas foram: ambiente de desenvolvimento Android Studio, linguagem de programação Java e plataforma Firebase.

Já o trabalho realizado por [Ramos 2013], foi desenvolvido um sistema para gerenciamento dos trabalhos finais de graduação, com o intuito de informatizar, agilizar, e principalmente gerar relatórios precisos para auxiliar os envolvidos nas corretas tomadas de decisões. O trabalho analisado fez uso da Linguagem de Programação Groovy, *Framework* Grails, *Spring framework*, *Hibernate framework*.

Todos os trabalhos discutidos, de alguma forma, utilizaram *frameworks* de apoio e computação em nuvem, igualmente proposto neste trabalho, que utiliza Django e o Bootstrap em servidor Web. Os trabalhos de [dos Santos et al. 2015] e [Sangiovo et al. 2018] também utilizaram controle de versionamento de código via Git no servidor GitHub. O trabalho de [Ramos 2013] possui como objetivo específico, geração de relatórios para coordenadores e responsáveis visando futuras tomadas de decisões, igualmente proposto por esse módulo.

3. Materiais, Métodos e Proposta do Sistema

A metodologia científica utilizada no trabalho foi pesquisa exploratória com revisão bibliográfica auxiliada por estudo de caso. E como metodologia de projeto e desenvolvimento de sistema, foram utilizadas boas práticas de SCRUM [Pagotto et al. 2016], com auxílio da técnica Kanban (via ferramenta Trello) para gestão do plano de atividades. As ferramentas utilizadas foram: linguagem de programação Python; *framework* Django [Alchin 2013]; *framework* Bootstrap; Git (Github) como controle de versionamento de código [Github 2019]; bancos de dados MySQL e SQLite; Astah para modelagem do sistema; Bizagi para modelagem de negócios. Para a geração de relatórios foi utilizada a biblioteca *django-easy-pdf*. Ressalta-se o uso dos dois bancos de dados, pois por se utilizar gerenciamento de versões, o banco de dados local é SQLite (para o processo de desenvolvimento), enquanto o do servidor é o MySQL⁷.

No contexto deste trabalho, o 'dono do produto' ou a figura que representa o cliente é o gestor do COMIC. O time de desenvolvimento é composto pelo aluno e pelo seu orientador, que desempenham papéis de desenvolvedores, analistas e testadores. Por fim, mas não menos importante, o *scrum master*, representado pelo professor orientador, que é o responsável em remover obstáculos à capacidade de entregar metas e os resultados do produto. O *scrum master* não é um gerente de projetos tradicional, mas atua como um facilitador entre a equipe e quaisquer influências perturbadoras. *Scrum master* garante que a estrutura da metodologia (*framework*) seja seguida [Sutherland 2016].

A Figura 5 ilustra o Diagrama de Casos de Uso do SISGEP-COMIC, ou seja, todas as funcionalidades existentes.

Já a funcionalidade "informar prazos" permite que o usuário administrador relate os projetos com prazos especificados manualmente, por exemplo. A funcionalidade "ca-

⁷É uma prática do Django fornecer e habilitar o banco de dados SQLite para desenvolvimento local. O arquivo gerado do banco também é compartilhado no Git, facilitando os testes com o sistema em construção. Entretanto, a documentação Django não recomenda seu uso em servidores com o sistema em operação.

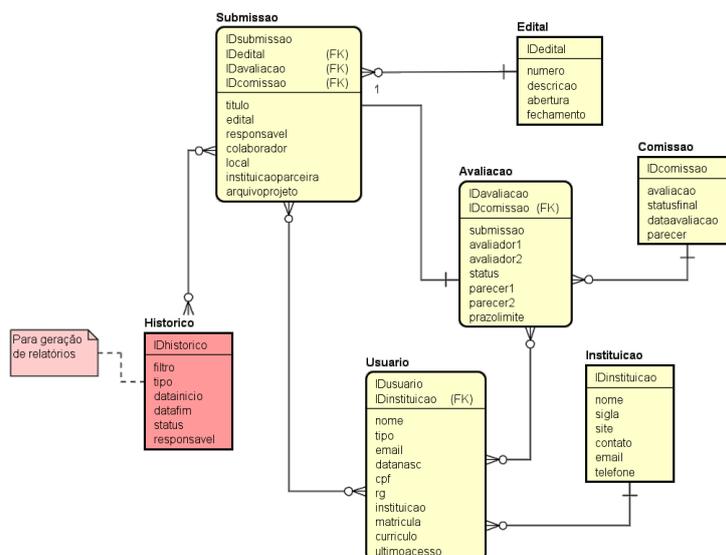


Figura 6. Diagrama de entidade e relacionamento do SISGEP-COMIC com a funcionalidade proposta em destaque.

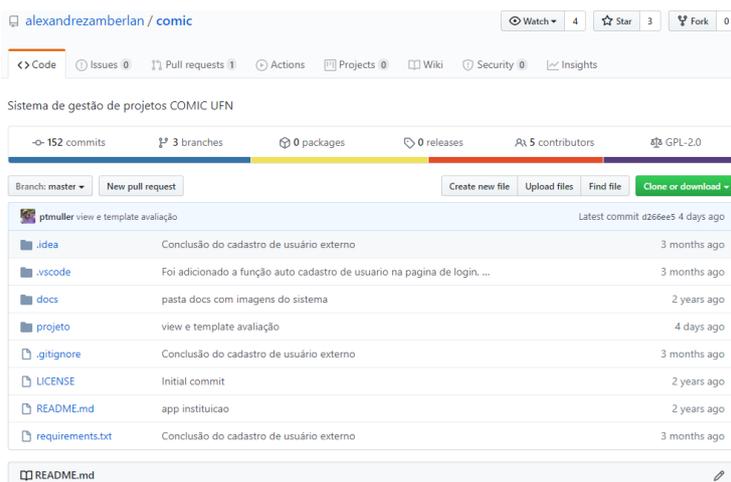


Figura 7. GitHub do projeto COMIC.

4. Resultados

A interface do protótipo na seção Avaliações de Projetos da visão Administrador (Figura 10) mostra a posição em que o administrador do SISGEP-COMIC tem acesso aos status de avaliação de cada projeto enviado, quem já avaliou, quem está em atraso, por exemplo.

As Figuras 11 e 12 mostram a possibilidade do administrador do COMIC acompanhar a situação de projetos aprovados/reprovados por edital.

Classe *Edital* (Figura 13) contém as *properties* para listagens de projetos aprovados e reprovados. Uma *propertie* em Python-Django é um método que ao final de sua execução transforma-se em atributo. É considerada uma forma de acessar um valor depois de um processamento de código. Observe as duas últimas *properties* da figura (*submissoes_aprovadas* e *submissoes_reprovadas*), há chamadas de duas consultas (*que-*

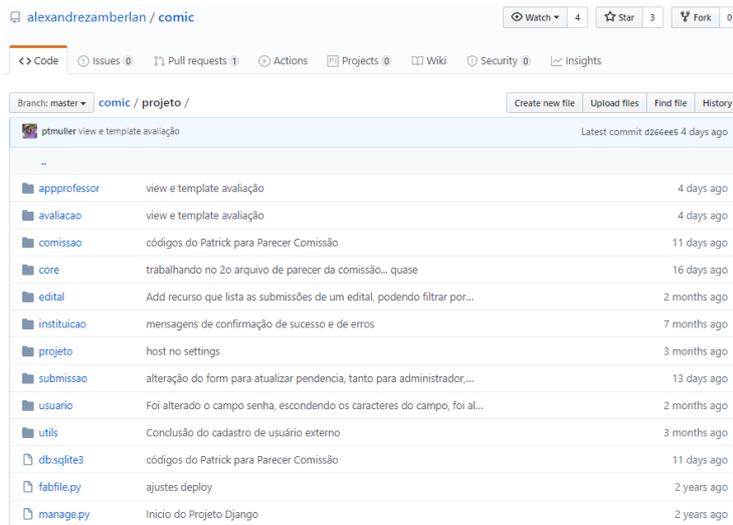


Figura 8. GitHub do projeto com a distribuição dos *apps*.

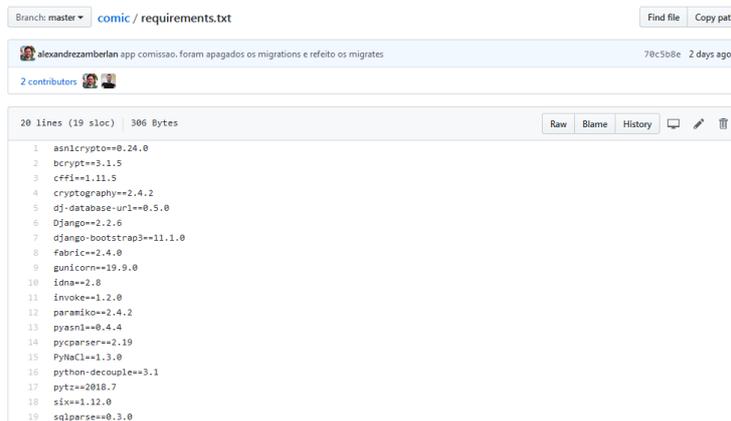


Figura 9. Arquivo *requirements.txt* com todas as especificações de bibliotecas e software necessários para executar SISGEP-COMIC.



Figura 10. Tela SISGEP-COMIC visão do Administrador com o protótipo do menu de Relatórios.

| Numero | Descrição | Abertura | Fechamento | Submissões | Editar | Excluir |
|--------|---|-------------------------|------------------------|------------|--------|---------|
| 2020 | FLUXO CONTÍNUO DE PROJETOS DE PESQUISA 2020 | 14 de Fevereiro de 2020 | 10 de Dezembro de 2020 | 7 | G | E |

Figura 11. Relatório em editais para acompanhar aprovados e reprovados.

| Situação | Quantidade |
|------------|------------|
| Aprovadas | 0 |
| Reprovadas | 0 |
| Todas | 7 |

Figura 12. Relatório de aprovados e reprovados em edital específico.

ries), como se fossem comandos *select* da linguagem SQL, entretanto, são em um nível mais abstrato, independente do banco de dados utilizado pela aplicação. No caso da *property submissoes_reprovadas*, a consulta retorna um conjunto de submissões que foram avaliadas como *reprovadas* e que pertencem a este edital, ou ao edital vigente.

```

class Edital(models.Model):
    numero = models.CharField(_(u'Número do Edital *'), max_length=20, help_text='Campo obrigatório como todos os que tiverem *')
    descricao = models.CharField(_(u'Descrição *'), max_length=200)
    abertura = models.DateField(_(u'Abertura do edital *'), blank = False, null = False, help_text='dd/mm/aaaa')
    encerra = models.DateField(_(u'Encerramento do edital *'), blank = False, null = False, help_text='dd/mm/aaaa')

    class Meta:
        ordering = ['numero']

    def __str__(self):
        return '%s - Encerramento: %s' % (self.numero, self.encerra.strftime('%d/%m/%Y'))

    def save(self, *args, **kwargs):
        self.descricao = self.descricao.upper()
        self.numero = self.numero.upper()
        super(Edital, self).save(*args, **kwargs)

    @property
    def get_submissoes_todas_list_url(self):
        return reverse('submissao_edital_list', kwargs={'pk_edital': self.pk, 'situacao': 'TODAS'})

    @property
    def get_submissoes_aprovadas_list_url(self):
        return reverse('submissao_edital_list', kwargs={'pk_edital': self.pk, 'situacao': 'APROVADAS'})

    @property
    def get_submissoes_reprovadas_list_url(self):
        return reverse('submissao_edital_list', kwargs={'pk_edital': self.pk, 'situacao': 'REPROVADAS'})

    @property
    def submissoes(self):
        return Submissao.objects.filter(edital=self)

    @property
    def submissoes_aprovadas(self):
        return self.submissoes.filter(id__in=[avaliacao.submissao.id for avaliacao in Avaliacao.objects.filter(status='APROVADO', submissao__edital=self)])

    @property
    def submissoes_reprovadas(self):
        return self.submissoes.filter(id__in=[avaliacao.submissao.id for avaliacao in
        Avaliacao.objects.filter(status='REPROVADO', submissao__edital=self)])

```

Figura 13. Classe Edital dentro de *models* (camada *Model* do modelo MVT).

Outra contribuição significativa ao projeto SISGEP-COMIC foram as funcionalidades apresentadas na Figura 14 e Figura 15. A primeira mostra a seção do Administrador com a possibilidade da comissão alterar status do projeto, bem como enviar arquivo com a resposta ao professor responsável. Já a Figura 15, em que tanto professor pesquisador, quanto administrador do COMIC podem acessar o arquivo contendo o parecer gerado pela comissão.

A Figura 16 apresenta a visão da seção da Comissão (COMIC) na emissão de pareceres pós avaliação dos projetos submetidos. A comissão pode conceder uma pendência

COMIC Home Gestão básica Submissões de Projetos Avaliações de Projetos Pareceres COMIC Relatórios Sobre Login

Pareceres COMIC

Submeter parecer comissão

| Avaliação | Status final projeto | Data avaliação comissão | Parecer da comissão | Editar | Excluir |
|---|----------------------|-------------------------|---------------------|--------|---------|
| ESTUDO DAS COMPLICAÇÕES DA TIMPANOTOMIA POR TÉCNICA ENDOSCÓPICA EM UM HOSPITAL D - ÉLISSON KRUG OLIVEIRA - - PENDENTE | PENDENTE | 15/05/2020 21:54 | | | |
| PERCEÇÃO E VIVÊNCIA DOS ENFERMEIROS ACERCA DOS CUIDADOS PALIATIVOS NO CONTEXTO - CLÁUDIA ZAMBERLAN - - PENDENTE | PENDENTE | 15/05/2020 22:02 | | | |

Figura 14. Tela da parte do Administrador para pareceres da comissão.

COMIC Home Gestão básica Submissões de Projetos Avaliações de Projetos Pareceres COMIC Relatórios Sobre Login

Submissões de Projetos

Submeter projeto

| Edital | Título | Responsável | Colaboradores | Local execução | Cursos de graduação | Arquivo projeto | Arquivo parecer comissão | Editar | Excluir |
|---------------------------------|--|------------------------|------------------------------|---|-----------------------|-----------------|--------------------------|--------|---------|
| 2020 - Encerramento: 10/12/2020 | A RELAÇÃO ENTRE AS LICENÇAS MATERNIDADE E PATERNIDADE E OS ACIDENTES DE TRABALHO | LJANA BOHRER BERNI | Nenhum colaborador informado | HOSPITAL CASA DE SAÚDE E HOSPITAL SÃO FRANCISCO | PSICOLOGIA | | | | |
| 2020 - Encerramento: 10/12/2020 | ANOSOGNOSIA EM INDIVÍDUOS PÓS ACIDENTE VASCULAR CEREBRAL NA FASE AGUDA | MORGANA CHRISTMANN | Nenhum colaborador informado | UNIVERSIDADE FRANCISCANA | FISIOTERAPIA | | | | |
| 2020 - Encerramento: 10/12/2020 | ASSISTÊNCIA FARMACÉUTICA NOS CUIDADOS PALIATIVOS EM PACIENTES ONCOLÓGICOS DE HOSPITAL FILANTRÓPICO DE SANTA MARIA, -RS | JANE BEATRIZ LIMBERGER | Nenhum colaborador informado | HOSPITAL CASA DE SAÚDE | FARMÁCIA | | | | |
| 2020 - Encerramento: 10/12/2020 | ESTUDO DAS COMPLICAÇÕES DA TIMPANOTOMIA POR TÉCNICA ENDOSCÓPICA EM UM HOSPITAL DE ATENÇÃO SECUNDÁRIA DO SISTEMA ÚNICO DE SAÚDE | ÉLISSON KRUG OLIVEIRA | Nenhum colaborador informado | HOSPITAL CASA DE SAÚDE | FACULDADE DE MEDICINA | | | | |

Figura 15. Tela alterada para que Administrador ou Professor Pesquisador possa acessar o arquivo de resposta emitido pela comissão.

a um projeto, para que os responsáveis possam corrigi-lo e submetê-lo novamente.

Na seção de Parecer do COMIC, há 3 sub-seções: detalhes da submissão (com informações referentes à submissão que se está realizando parecer); detalhes da avaliação (com os dois textos dos avaliadores alocados para avaliação da submissão); área COMIC (em que a comissão gera os pareceres, tanto inicial, quanto o final).

O parecer é emitido em reuniões mensais ou quinzenais entre os membros do COMIC, em que analisam os textos dos avaliadores.

A Figura 17 mostra parte do código (camada *Template* do modelo MVT). Entre as linhas 74 e 99 há o código da subseção "Detalhes da avaliação", e é possível visualizar a mescla de código *html* com código Python. Por exemplo, o código da linha 82, trata do *object* parecer que tem uma avaliação (objeto *avaliacao_comissao*), que por sua vez tem um texto de avaliador (objeto *parecer_avaliador_responsavel*). Isso mostra uma das vantagens do *framework* Django, integrando as camadas *Model* e *Template*.

Outro exemplo está entre as linhas 90 e 94, em que surge uma condição que só vai mostrar o texto do avaliador suplente se o texto existir (suplente alocado há avaliação, já que é opcional sua presença).

☀ Parecer do COMIC

Detalhes da submissão

Título:
SISTEMA DE MINERAÇÃO DE DADOS PARA PREDIÇÃO DE PACIENTES ONCOLÓGICOS.

Local execução: HOSPITAL SÃO FRANCISCO **Área:** CIÊNCIAS TECNOLÓGICAS

Resumo:
O estudo busca analisar os dados do banco contendo informações de pacientes e seus respectivos tratamentos oncológicos.

Palavras-chave:
Mineração de dados; Redes Neurais; Classificação; Treinamento supervisionado

Arquivo do projeto:

Arquivo de correção do projeto após pendência:

Detalhes da avaliação

Texto do avaliador Responsável:

Texto do avaliador Suplente:
Sem avaliador suplente ou texto não enviado.

Área COMIC

| Descrição | Resposta |
|--|--|
| Status final do projeto * | <input type="button" value="Aprovado"/> ▼ |
| Arquivo inicial de resposta à submissão do projeto (Use arquivo .pdf para enviar seu projeto) * | Atualmente: midias/parecerInicialComissao_gk8vDee.txt <input type="button" value="Limpar"/> Modificar: <input type="button" value="Choose file"/> No file chosen |
| Arquivo de parecer depois das correções solicitadas pela comissão (Use arquivo .pdf para enviar seu projeto) | Atualmente: midias/Screenshot_2020-05-14_at_14.47.47_25A1pq1.png <input type="button" value="Limpar"/> Modificar: <input type="button" value="Choose file"/> No file chosen |

ATENÇÃO: há dois arquivos que a comissão pode emitir.

1. O primeiro que oferece um parecer inicial e dá a oportunidade de ajustes se a avaliação ficar no STATUS PENDENTE.
2. O último, que é o arquivo contendo o parecer final à submissão reenviada com correções.

Figura 16. Gestão de um parecer em que a comissão está analisando projeto.

```
comissao_form.html X
projeto > comissao > templates > comissao > comissao_form.html > div:panel:panel-default > div:panel-body > form:form
74 <div class="panel panel-default">
75 <div class="panel-heading">
76 <h3 class="panel-title">Detalhes da avaliação</h3>
77 </div>
78 <div class="panel-body">
79 <div class="row">
80 <div class="col-xs-12 col-sm-12 col-md-8 col-lg-8">
81 <strong>Texto do avaliador Responsável:</strong>
82 <br>{{ object.avaliacao_comissao.parecer_avaliador_responsavel }}
83 </div>
84 </div>
85 <br>
86 <hr>
87 <div class="row">
88 <div class="col-xs-12 col-sm-12 col-md-8 col-lg-8">
89 <strong>Texto do avaliador Suplente:</strong><br>
90 {{ if object.avaliacao_comissao.parecer_avaliador_suplente }}
91 {{ object.avaliacao_comissao.parecer_avaliador_suplente }}
92 {{ else }}
93 Sem avaliador suplente ou texto não enviado.
94 {{ endif }}
95 </div>
96 </div>
97 </div>
98 </div>
99 </div>
100
101 <div class="panel panel-default">
102 <div class="panel-heading">
103 <h3 class="panel-title">Área COMIC</h3>
104 </div>
```

Figura 17. Trecho html da seção Parecer COMIC (camada *Template* do modelo MVT).

5. Conclusões

Os sistemas de gestão de projetos auxiliam gestores na organização de submissões, avaliações, pareceres e acompanhamento de projetos, em diferentes áreas. O módulo construído e disponibilizado gerencia todo o processo de avaliação de projetos junto ao

setor que administra submissões, avaliações e acompanhamento de projetos científicos na Saúde. O módulo foi projetado e implementado por meio da metodologia Scrum Solo, técnica de gestão de atividades Kanban (ferramenta Trello), linguagem de programação Python, frameworks Django e Bootstrap e ferramentas de modelagem Bizagi e Astah. Além disso, os trabalhos relacionados fornecerem recursos para a construção de um módulo adequado, com uso de boas práticas e ferramentas de desenvolvimento.

Acredita-se que todos os objetivos assumidos foram alcançados, uma vez que o módulo está em funcionamento. Houve adequações no modelo do sistema, no banco de dados, no fluxo de funcionamento do SISGEP.

Por se tratar de um projeto já em andamento, estruturalmente e funcionalmente, o SISGEP-COMIC encontra-se disponível e em uso por professores da instituição, entretanto, ele tem comportamento exclusivo de sistema de informação gerencial, pois somente armazena e relata dados de usuários, pesquisadores, projetos de pesquisas e suas avaliações.

Referências

- Alchin, M. (2013). *Pro Django*. Apress, New York.
- Brasil (2013). *Resolução n 466, de 12 de dezembro de 2012. Dispõe sobre diretrizes e normas regulamentadoras de pesquisas envolvendo seres humanos*. Diário Oficial [da] República Federativa do Brasil, Brasília, DF.
- Da Cunha, A. G. (2019). *Dicionário etimológico da língua portuguesa*. Lexikon Editora.
- dos Santos, R. E., da Silva, R. F., and de Oliveira Zamberlan, A. (2015). Proposta de uma plataforma de cloud computing para disponibilização de um sistema online para consultórios e clínicas por meio do modelo saas. In *XIII Simpósio de Informática - SIRC*, page 90. Centro Universitário Franciscano.
- Freeman, E. and Freeman, E. (2009). *Use a Cabeça - Padrões de Projetos*. Alta Books Editora, 2nd edition.
- Github (2019). GitHub Oficial. Website: <http://github.com>.
- Laudon, J. P. and Laudon, K. C. (2014). *Sistemas de Informação Gerenciais*. Pearson, 11st edition.
- McLaughlin, B., Pollice, G., and West, D. (2007). *Use a Cabeça - Análise e Projeto Orientado ao Objeto*. Alta Books Editora.
- 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.
- Pressman, R. S. (2010). *Software Engineering: a Practitioner's Approach*. McGraw-Hill Education, New York, 7th edition.

- Ramos, P. P. (2013). Desenvolvimento de aplicação Web para gerenciamento dos trabalhos finais de graduação. page 16. Centro Universitário Franciscano.
- 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.