

GESTARQ, MODELAGEM DE APLICAÇÃO WEB PARA GERENCIAMENTO DE MICRO ESCRITÓRIOS DE ARQUITETURA

Táisia Cruz de Souza Honório Delgado¹
Ana Carolina Costa de Oliveira²

RESUMO

Um dos principais desafios na execução de projetos de arquitetura é o relacionamento com clientes, incluindo a comunicação e o gerenciamento de documentos relacionados ao projeto. Nesse contexto, sistemas de gerenciamento de projetos de arquitetura podem auxiliar nesse aspecto e em outros relacionados à execução. No entanto, é importante verificar os requisitos específicos de escritórios de arquitetura de diferentes portes, uma vez que uma solução que funciona para grandes escritórios com equipes compostas por muitos profissionais talvez não se aplique de maneira satisfatória em escritórios pequenos, que contam com um ou poucos arquitetos e colaboradores envolvidos no projeto. Este trabalho de conclusão de curso se propõe a realizar a especificação e modelagem de uma aplicação web para gerenciamento de micro escritórios de arquitetura, visando atender às características específicas dos escritórios com esse porte e oferecer uma solução para o aumento de eficiência e produtividade na execução de projetos de arquitetura.

Palavras-chaves: Aplicação web; Gerenciamento de projetos; Escritórios de arquitetura.

ABSTRACT

One of the main challenges in the execution of architectural projects is the relationship with clients, including the communication and management of documents related to the project. In this context, architectural project management systems can help in this aspect and others related to execution. However, it is important to check the specific requirements of architectural offices of different sizes, since a solution that works for large offices with teams composed of many professionals may not be satisfactorily applied in small offices, which have one or few architects and collaborators involved in the project. This course completion work proposes to specify and model a web application for managing micro architecture offices, aiming to meet the specific characteristics of offices of this size and offer a solution to increase efficiency and productivity in the execution of architecture projects.

¹ Graduanda do Curso de Sistemas para Internet. E-mail: taisiahonorio@gmail.com

² Professor Orientador, Tecnóloga em processamento de dados pela Faculdade Paraibana de Processamento de Dados, Doutorado em Modelo de Decisão e Saúde pela Universidade Federal Da Paraíba , Mestrado em Engenharia de Produção pela Universidade Federal da Paraíba , Especialização em Informática e Saúde pela Unifesp, Especialização em Engenharia e Administração de Dados pela Uniesp, Docente do Curso de Sistemas para Internet do Uniesp da Disciplina Trabalho de Conclusão de Curso. E-mail: prof2121@iesp.edu.br

1 INTRODUÇÃO

O gerenciamento adequado de projetos é peça fundamental para obtenção de bons resultados, uma vez que ele traz mais eficácia e eficiência para o processo de desenvolvimento dos projetos. Segundo o *PMBOK*, “O gerenciamento de projetos é a aplicação de conhecimentos, habilidades, ferramentas e técnicas às atividades do projeto a fim de cumprir os seus requisitos.” (PMI, 2017 p.10). Mais especificamente, no contexto de projetos de arquitetura, as deficiências de gestão nas empresas de projeto de arquitetura ocorrem principalmente no relacionamento com os clientes, na documentação em geral e na comunicação, advindos muitas vezes da informalidade no processo (MELHADO, 2001).

Nesse sentido, de acordo com Miranda (2022, p. 19), “[...] a adoção de sistemas computacionais se mostra necessária para mediar tais relações e necessidades, permitindo que o profissional atenda com eficiência o seu cliente, dedicando maior parte do tempo a ele”. Há no mercado aplicações capazes de auxiliar o gerenciamento de projetos para escritórios de arquitetura de diversos portes, como o ProjeTools³ e o Doit⁴. No setor de serviços, como é o caso de arquitetura, escritórios com até 9 pessoas são considerados micro escritórios pelo Serviço Brasileiro de Apoio às Micro e Pequenas Empresas (SEBRAE, 2013).

Mas o que as aplicações de gerenciamento de projeto de arquitetura atuais não entregam? Elas não são focadas nas demandas específicas de gestão dos micro escritórios. Mais especificamente, o foco dessas ferramentas é na gestão de grandes times de arquitetura e não no relacionamento entre arquitetos (ou pequenos grupos de arquitetos) com os clientes, o que, de acordo com Melhado (2001), é um dos principais gargalos da gestão de projetos das empresas de arquitetura.

As necessidades de gerenciamento se assemelham em muitos pontos para todos os tipos de porte de escritório de arquitetura. Mas há uma grande diferença nas demandas de gestão de micro escritórios dos de portes maiores. Os micro escritórios encontram dificuldades relacionadas ao desenvolvimento em paralelo de múltiplas tarefas de diversas naturezas por um único indivíduo, diferente do que ocorre com escritórios maiores, que necessitam controlar diversas atividades em torno de uma equipe mais numerosa. Essa discrepância de quantidade de pessoas envolvidas no trabalho e sua divisão de tarefas é uma lacuna nas aplicações existentes.

³ <http://www.projetools.com.br/>

⁴ <https://www.doit.com.br/>

De acordo com o II Censo divulgado em 2020 pelo Conselho de Arquitetura e Urbanismo no Brasil, o nosso país possui atualmente cerca de 212.000 arquitetos, sendo 51% autônomos trabalhando diretamente com arquitetura (CAU-BR,2020). Os números indicam um possível mercado em aberto, com demandas ainda não atendidas.

Nesse contexto, este trabalho tem como objetivo geral modelar e implementar funcionalidades básicas de um sistema web visando auxiliar o gerenciamento de projetos de arquitetura em micro escritórios, bem como a comunicação com parceiros e clientes, além da disponibilização de documentos para o seu manejo e controle. Por entender que este porte de empresa têm demandas de gerenciamento diferentes das de portes maiores, o trabalho busca criar uma aplicação mais dedicada às necessidades desses micro escritórios e capaz de atender este espaço existente no mercado.

A seguir, o trabalho apresenta uma fundamentação teórica, capaz de embasar etapas posteriores, seguida da apresentação da metodologia adotada. Posteriormente, são descritas as etapas de desenvolvimento realizadas no contexto deste trabalho e, por fim, são descritas as considerações finais.

2 FUNDAMENTAÇÃO TEÓRICA

“A comunicação do projeto consiste em múltiplas atividades e consome a maior parte do tempo do gerente de projetos que está paralelamente, monitorando todas as outras atividades administrativas ...” (MIRANDA,2022, p.18). De forma complementar, Sousa (2010 *apud* GRILO, 2002) coloca que a comunicação foi apontada por 25% dos clientes, como a razão básica para a ocorrência de falhas em empreendimentos de arquitetura.

Em micro escritórios de arquitetura, o gerente de projetos é também o responsável pelo desenvolvimento do projeto bem como acompanhamento da obra, controle financeiro, além de estratégias de marketing e comunicação direta com clientes e parceiros, que acabam por gerar uma sobrecarga de múltiplas atividades que devem ser desenvolvidas em paralelo.

Para Fabrício e Melhado (2002) uma dificuldade significativa criada pelo emprego de meios eletrônicos de comunicação é o vertiginoso aumento na quantidade e no fluxo de informações entre as pessoas juntamente com falta de procedimentos e de normas de comportamento . Isso gera uma sobrecarga informacional que faz dados importantes serem desconsiderados e perdidos dentre as muitas informações recebidas.

Com o desenvolvimento da tecnologia temos hoje em dia diversas ferramentas de comunicação. Esta variedade de opções, embora seja uma facilitadora, pois atende melhor mais pessoas, pode acarretar em problemas quando não é bem manejada, visto que a falta de controle sobre ela é capaz de implicar na perda de dados importantes. Este talvez seja um dos grandes desafios encontrados atualmente por micro escritórios de arquitetura. Os dados podem estar espalhados por diversas ferramentas, como e-mail, WhatsApp, Telegram, serviços na nuvem e locais físicos, relacionados com a informalidade citada na introdução e ausência de uma padronização. Quando se há uma centralização dos meios de comunicação pode-se realizar buscas de maneira mais ágil. Dessa forma a padronização pode ser um caminho para evitar esses problemas e retirar o caráter informal já mencionado.

2.1 APLICAÇÕES EXISTENTES PARA GERENCIAMENTO DE PROJETOS DE ARQUITETURA

Conforme as descrições apresentadas a seguir, de duas grandes plataformas de gestão de projetos na área de arquitetura, e após a análise detalhada de cada uma delas, foi observado que as aplicações, embora atendam e automatizem partes importantes do processo de gerenciamento dos escritórios de arquitetura, estão mais voltadas para a administração interna e financeira. O problema da comunicação entre profissional e cliente abordado na introdução, que será o foco da aplicação proposta neste trabalho de conclusão de curso, não é plenamente atendido. A seguir é fornecida uma breve descrição sobre as duas ferramentas mencionadas.

2.1.1 DOit

De acordo com a página virtual da aplicação em questão, o DOit⁴ é um sistema on-line desenvolvido para auxiliar escritórios de arquitetura e interiores na administração do dia a dia, desde o primeiro contato do cliente até a entrega final do projeto. A aplicação conta com os seguintes módulos : financeiro (controle de contas emissão relatórios), projeto (controla as horas trabalhadas), faturamento (emite boletos e notas fiscais), cadastro (cria um banco de dados de parceiros e clientes com informações pessoais e histórico de eventos vinculados ao financeiro), agenda, e-mail (envia emails automáticos com cobranças etc) e listas (direcionam as malas diretas) (DOIT,2022).

2.1.2 PROJE TOOLS

De acordo com informações obtidas na página da plataforma, o ProjeTools³ é um software criado visando automatizar o gerenciamento de informações dos projetos de Arquitetura ou Engenharia, controlando as fases do projeto como a parte inicial de cadastros de clientes, tarefas, programas de necessidades, acompanhamento, gerenciamento de arquivos e controle financeiro, além de possibilitar o acesso às informações em qualquer lugar.

Algumas de suas funcionalidades apresentadas são: gerenciamento de clientes e profissionais envolvidos com os dados da obra; criação de programas de necessidades e acompanhamento de prazos; emissão de recibos, requerimentos e contratos, integração com órgãos públicos a depender da região, servidor para backup e acesso remoto (PROJETTOOLS,2022).

2.2 PROCESSO DE DESENVOLVIMENTO DE SOFTWARE

Pressman (2021) afirma que para a elaboração de um sistema é pertinente seguir uma série de etapas como forma de auxiliar a qualidade nos resultados, denominando esse conjunto de passos de “processo de software”. Usualmente segue-se um modelo de processo de desenvolvimento de software que desenvolve uma modelagem do sistema. De acordo com Pressman (2021, p. 100) “O objetivo dos modelos é solidificar a compreensão do trabalho a ser feito e providenciar orientação técnica aos implementadores do software”. Sommerville (2019) descreve a modelagem de sistema como um processo de desenvolvimento de modelos abstratos onde cada modelo apresenta uma perspectiva diferente do sistema. Essas modelagens se utilizam de notações gráficas e normalmente são baseadas em notações de UML (linguagem de modelagem unificada, do inglês *Unified Modeling Language*).

Para o desenvolvimento da aplicação, algumas fases serão realizadas, como o levantamento de requisitos e o projeto do sistema. Durante essas fases, alguns modelos podem ser criados, a exemplo dos diagramas de casos de uso e diagramas de classes. Também pode-se criar protótipos que auxiliam no refinamento dos requisitos e do projeto do sistema. De acordo com Sommerville (2019), os requisitos de um sistema são as descrições do que o sistema deve fazer, bem como dos serviços que oferece e das restrições a seu funcionamento. Eles devem refletir as necessidades dos clientes e por isso dependem do tipo de software a ser desenvolvido e de quem são seus possíveis usuários.

Os casos de uso identificam os atores envolvidos em uma interação dando um nome para ela. Essa é, então, complementada com informações adicionais que descrevem a interação com o sistema. Os diagramas de classe são usados no desenvolvimento de um modelo de sistema orientado a objetos para mostrar as classes de um sistema e as associações

entre elas. Um protótipo é uma versão inicial de um sistema de software, usado para demonstrar conceitos, experimentar opções de projeto e descobrir mais sobre o problema e suas possíveis soluções. Ele pode ser útil nos estudos de soluções específicas para o software e no projeto de interface com o usuário. Protótipos do sistema permitem aos usuários ver quão bem o sistema atende a necessidade. Através dele pode se perceber pontos fortes e fracos do software revelando erros e omissões nos requisitos propostos (SOMMERVILLE, 2019).

3 METODOLOGIA

O trabalho em questão se enquadra em uma pesquisa aplicada, com objetivos exploratórios e procedimentos bibliográficos. As práticas metodológicas utilizadas neste trabalho são :

1. Revisão bibliográfica para identificação de problemáticas apontadas por demais autores e reconhecimento do que já é oferecido nas aplicações existentes;
2. Levantamento de requisitos: consiste em entender melhor o funcionamento de um micro escritório para então montar uma lista especificando tudo aquilo que é necessário existir na aplicação a fim de sanar o problema existente;
3. Modelagem inicial do Sistema: A técnica utilizada será o uso de diagramas UML, aplicando os seguintes modelos: Diagrama de Caso de uso, Diagrama de Classes e Prototipação de telas;
4. O desenvolvimento da aplicação será feito apenas de modo inicial, utilizando a linguagem Python para implementar um CRUD inicial de cadastro com auxílio Framework Flask .

4.2.3 FLASK

O Flask consiste em um micro-framework para o desenvolvimento de aplicações Web em Python. Por ser um micro-framework, ele provê apenas as funcionalidades básicas para o desenvolvimento de uma aplicação Web simples, mas novos módulos podem ser adicionados à medida que sejam necessários no projeto (TREINAWEB, 2019). Por exemplo, no protótipo das funcionalidades básicas implementado para este trabalho foram utilizados módulos para facilitar a comunicação e configuração do Banco de Dados, bem como para facilitar a realização de login e autenticação de usuário.

4 RESULTADO E DISCUSSÕES

4.1 LEVANTAMENTO DE REQUISITOS FUNCIONAIS

O levantamento de requisitos apresentado no quadro foi elaborado com base na experiência da autora como arquiteta autônoma e leva em consideração as dificuldades encontradas para gerir os processos de projetos e relacionamento com clientes e parceiros ao longo do ciclo de vida dos projetos.

Quadro 01-Requisitos funcionais

REQUISITOS FUNCIONAIS	
Identificador	Descrição
RF01	Abertura de chamados por clientes e colaboradores externos
RF02	Acesso ao portfólio de projetos por qualquer usuário, cadastrado no sistema ou não
RF03	Upload de documentos por usuários autenticados, com registro de data
RF04	Solicitação de orçamentos por qualquer usuário, cadastrado no sistema ou não
RF05	Agendamentos de reuniões por clientes e colaboradores externos cadastrados
RF06	Cadastro de clientes, colaboradores e projetos pelo arquiteto
RF07	Envio de lembretes e confirmações para clientes, arquitetos e colaboradores
RF08	Criação de atas de reunião por meio de templates pré-estabelecidos
RF09	Cadastro de informações e imagens sobre projetos para composição do portfólio
RF10	Criação de listas de atividades a serem realizadas nos projetos
RF11	Formulário de avaliação e geração de relatório de feedback de clientes ao final do projeto
RF12	Acompanhamento do desenvolvimento do projeto por parte do cliente

Fonte: Elaborado pela autora

4.2 MODELAGEM UML

4.2.1 DIAGRAMA DE CASO DE USO

O diagrama de caso de uso considera cinco personagens, para sua elaboração a ferramenta utilizada foi o Miro.⁵ Um possível cliente é retratado como *user:externo* e não necessita de autenticação no sistema e, portanto, utiliza apenas funcionalidades como

⁵ www.miro.com

visualização de portfólio e solicitação de orçamento. É importante destacar nesse caso a presença de um portfólio, que vai além das redes sociais, como um meio seguro e independente de divulgação, visto que é comum a perda de acesso das contas nas redes sociais em virtude de ataques hackers.

Figura 01: Diagrama de Caso de Uso User.Ext



Fonte: Elaborado pela autora utilizando a ferramenta Miro

O ator *colab.interno* equivale, neste caso, a estagiários ou projetistas que não têm qualquer participação administrativa no escritório, sendo responsáveis apenas por realizar trabalhos técnicos e, portanto, com acesso limitado a funcionalidades dessa natureza, como visualização e inserção de arquivos relacionados a esse tipo de trabalho.

Figura 02: Diagrama de Caso de Uso COLAB.INT



Fonte: Elaborado pela autora utilizando a ferramenta Miro

O *colab.ext* equivale nesse caso aos engenheiros e projetistas contratados por clientes para realização de trabalhos complementares ao projeto, que não são ofertados pelo escritório. Dessa forma, é necessário a permissão apenas para as funcionalidades de acesso aos arquivos, agendamento de reuniões e abertura de chamados.

Figura 03: Diagrama de Caso de Uso COLAB.Ext



Fonte: Elaborado pela autora utilizando a ferramenta Miro.

Os dois últimos atores (cliente e arquiteto) são considerados os principais, em virtude do sistema proposto ser dedicado principalmente a solucionar problemas de ruído de comunicação entre eles, facilitando a documentação e gestão de todo o trabalho. Sendo assim, o personagem *cliente* pode, após fechamento de contrato, dispor de login e senha que permitam um acesso autenticado capaz de permitir que este veja todos os arquivos relacionados ao seu projeto, abra chamados, agende reuniões e visitas técnicas através da visualização de disponibilidade de agenda do sistema, além de acompanhar o andamento do projeto através do cronograma e da verificação das etapas já realizadas, com suas respectivas datas.

Figura 04: Diagrama de Caso de Uso Cliente

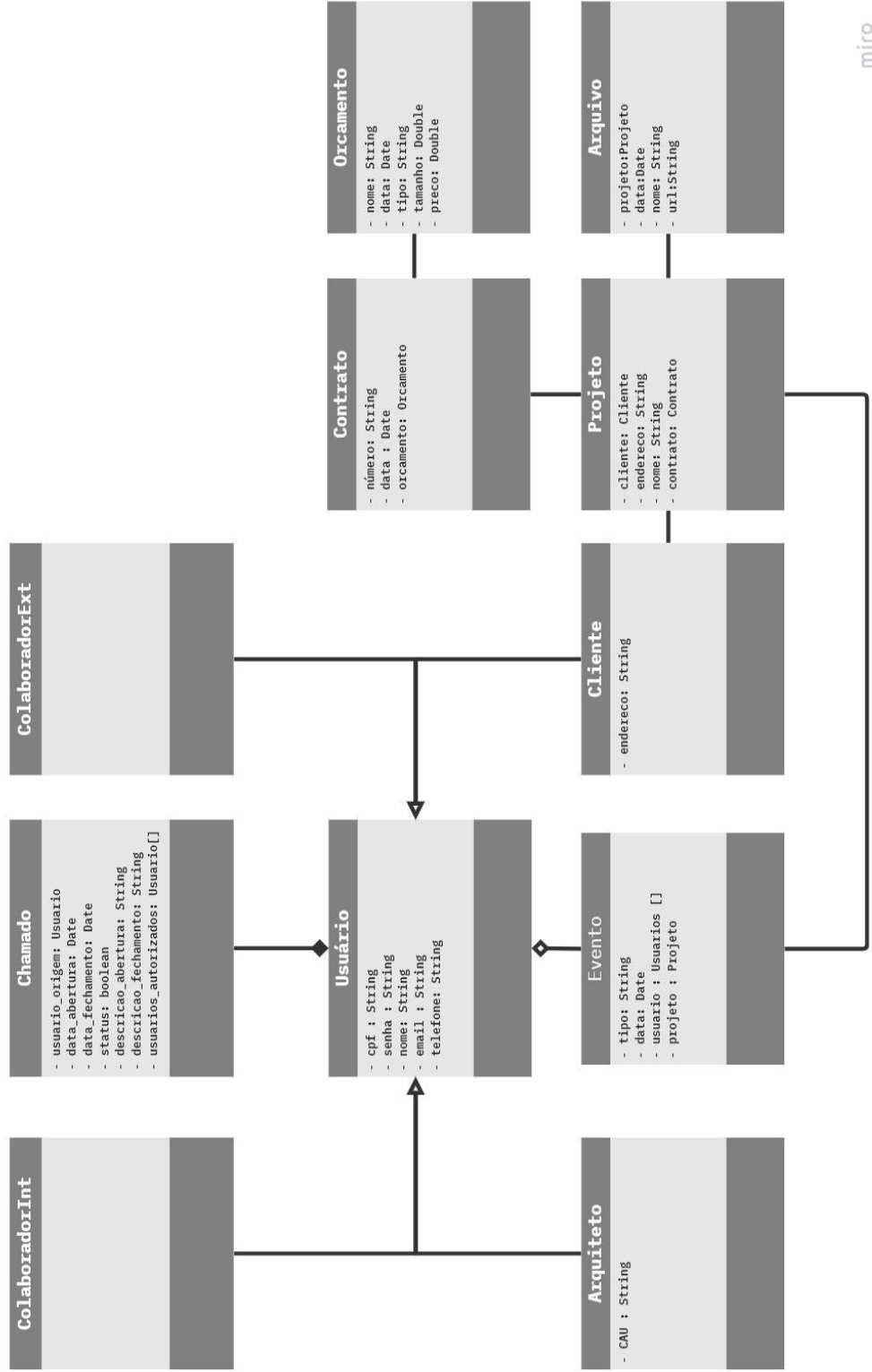


Fonte: Elaborado pela autora utilizando a ferramenta Miro

desenvolvidas para este trabalho, mais detalhes sobre as outras classes implementadas são fornecidos.

O diagrama de classes a seguir demonstra que existe a superclasse *Usuario* e que as classes *Arquiteto*, *UsuarioExterno*, *ColaboradorExt*, *ColaboradorInt*, e *Cliente* possuem relação de herança perante ela. A classe *Chamado* tem relação de composição com *Usuario* e classe *Evento* relação de agregação também com *Usuario*. Demais classes estão representadas apenas com associações simples.

Figura 05:Diagrama de Classe



Fonte: Elaborado pela autora utilizando a ferramenta Miro

4.2.3 IMPLEMENTAÇÃO DE FUNCIONALIDADES BÁSICAS E PROTOTIPAGEM DE TELAS

Nesta seção é descrita a implementação das telas e funcionalidades básicas do sistema, que puderam ser desenvolvidas dentro do tempo hábil para este trabalho. Outras funcionalidades e telas serão desenvolvidas como trabalhos futuros. A implementação se baseou no tutorial disponível em (DIGITAL OCEAN,2020).

Para implementar essas funcionalidades básicas, foi utilizado o conceito de *Blueprint* de Flask para a divisão das funcionalidades em módulos diferentes. Mais especificamente, foi desenvolvida a *Blueprint* para a página inicial e a *Blueprint* para a realização de login.

Inicialmente, foi implementada a *Classe Usuario*, de acordo com o Diagrama de Classes mostrado na Seção 4.2.2. A implementação dessa classe foi necessária para permitir a implementação da funcionalidade de *login*. A Figura 06 mostra uma parte do código da *Classe Usuario*, contendo os atributos. Por meio da biblioteca de banco de dados do *Flask*, a partir da criação dessa classe, a tabela correspondente é criada no MySQL.

Figura 06. Código do arquivo *models.py* contendo os atributos da *Classe Usuario*.

```
class Usuario(UserMixin, db.Model):
    """
    Cria uma tabela Usuario
    """
    __tablename__ = 'usuarios'

    id = db.Column(db.Integer, primary_key=True)
    cpf = db.Column(db.String(11), unique=True)
    password_hash = db.Column(db.String(128))
    nome = db.Column(db.String(120), index=True)
    email = db.Column(db.String(60), index=True, unique=True)
    telefone = db.Column(db.String(11), index=True, unique=True)
    is_admin = db.Column(db.Boolean, default=False)
```

Fonte: elaborado pela autora

Cada *Blueprint* possui um arquivo chamado *views.py*, que é responsável por tratar as requisições HTTP e retornar a página HTML que deve ser renderizada no *browser*. A Figura 07 a seguir mostra o código do arquivo *views.py* da *Blueprint* responsável por implementar a página inicial. O código completo para essas funcionalidades básicas pode ser encontrado no repositório disponibilizado no GitHub⁶.

⁶<https://github.com/taisiah/Gestarq>

Figura 07: Código do arquivo *views.py* da *Blueprint* responsável pela página inicial

```
from flask import render_template
from flask_login import login_required

from . import home

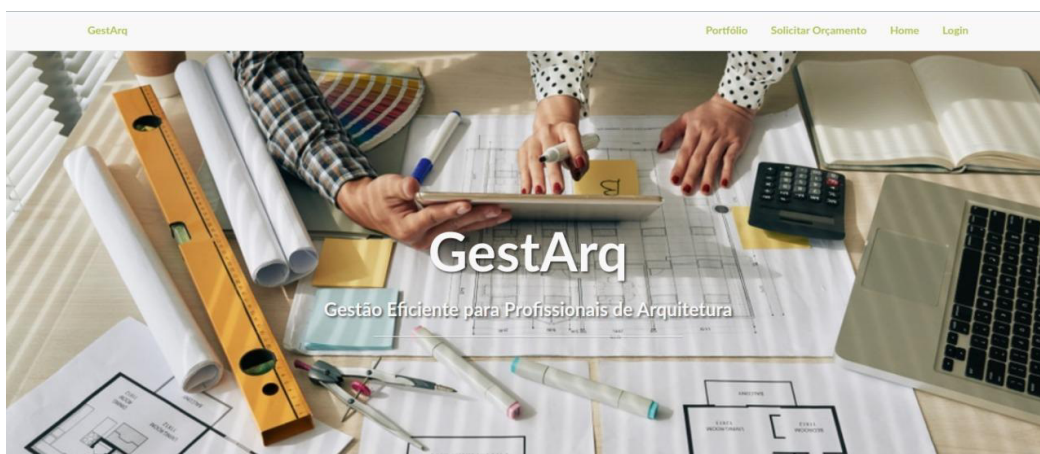
@home.route('/')
def homepage():
    """
    Retorna o HTML para a página inicial
    """
    return render_template('home/index.html', title="Welcome")

@home.route('/dashboard')
@login_required
def dashboard():
    """
    Retorna o HTML para o dashboard (página inicial após login)
    """
    return render_template('home/dashboard.html', title="Dashboard")
```

Fonte: elaborado pela autora

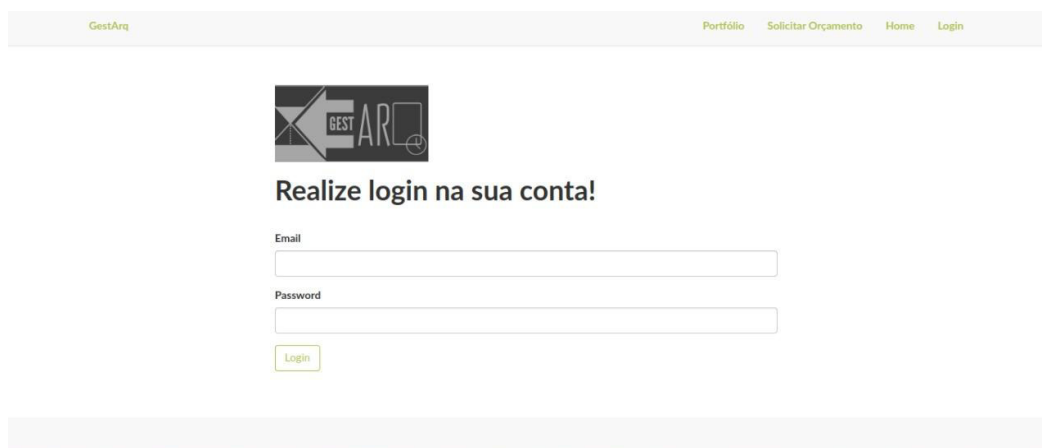
As Figuras 08 e 09 a seguir mostram as duas telas que foram implementadas. A primeira diz respeito à tela inicial do sistema, enquanto a segunda diz respeito à tela de *login*.

Figura 08: Tela inicial



Fonte: elaborado pela autora

Figura 09:Tela login



The image shows a web browser window displaying the login page of a system named 'GestArq'. The header bar is light gray with 'GestArq' on the left and four links: 'Portfólio', 'Solicitar Orçamento', 'Home', and 'Login' on the right. The main content area is white. It starts with the 'GestArq' logo, which consists of a stylized 'G' and 'A' with the text 'GEST ARQ' inside. Below the logo, the text 'Realize login na sua conta!' is displayed. Underneath this text is a login form with two input fields: 'Email' and 'Password'. Below the 'Password' field is a yellow 'Login' button. The browser's scrollbar is visible on the right side of the page.

Fonte: elaborado pela autora

5 CONSIDERAÇÕES FINAIS

Este trabalho descreveu a concepção de um sistema para gerenciamento de projetos em micro-escritórios de arquitetura. O trabalho possibilitou uma análise aprofundada das necessidades pertinentes ao sistema proposto e através da construção de uma modelagem inicial obteve-se um primeiro protótipo de funcionamento com funcionalidades básicas. Tais procedimentos são alinhados com as práticas metodológicas do sistema ágil, este é feito através de incrementos ao longo do processo de desenvolvimento que serão feitos a posteriori com o objetivo de ajustar as expectativas, de modo que o resultado seja mais eficaz e eficiente. Dessa forma, servirá para uma implementação completa do sistema em trabalhos futuros.

REFERÊNCIAS

ANUÁRIO DO TRABALHO NA MICRO E PEQUENA EMPRESA 2013 SEBRAE [2013]

Disponível em:

<https://www.sebrae.com.br/Sebrae/Portal%20Sebrae/Anexos/Anuario%20do%20Trabalho%20Na%20Micro%20e%20Pequena%20Empresa_2013.pdf> Acesso em: 19/08/2022

BEBER, Michelle; SCHEER, Sérgio; WILLE, SA de C. **Uso da tecnologia da informação como auxiliadora da gestão da comunicação em escritórios de arquitetura.** Encontro De Tecnologia De Informação E Comunicação Na Construção Civil, v. 3, p. 1-10, 2007.

BUILD A CRUD WEB APP WITH PYTHON AND FLASK. Digital Ocean. Disponível em

<<https://www.digitalocean.com/community/tutorials/build-a-crud-web-app-with-python-and-flask-part-one>>. Acesso em: 28/11/2022.

CENSO DO ARQUITETOS E ARQUITETAS E URBANISTAS DO BRASIL [2020] CAU-BR
Disponível em : <<https://caubr.gov.br/censo2020/>>. Acesso em: 23/08/2022

DE SOUZA, Carolina Pereira Rosa. **Gestão Dos Processos De Projeto Em Escritórios De Arquitetura De Pequeno Porte: Estudo De Caso E Elaboração De Modelo De Gestão**. 2013.

FABRICIO, Márcio Minto; MELHADO, Silvio Burrattino. **Impactos da tecnologia da informação nos conhecimentos e métodos projetuais**. Seminário De Tecnologia Da Informação E Comunicação Na Construção Civil, v. 1, 2002.

DOIT Disponível em: <<https://www.doit.com.br/>> Acesso em : 03/10/2022

MELHADO, Silvio Burrattino, **Gestão, Cooperação E Integração Para Um Novo Modelo Voltado À Qualidade Do Processo De Projeto Na Construção De Edifícios**. São Paulo, 2001.

MIRANDA, Rafaela Silva. **Sistema Web Para Gerenciamento De Um Escritório De Arquitetura**. 2022

O QUE É FLAK?. TREINAWEB. Disponível em
<<https://www.treinaweb.com.br/blog/o-que-e-flask>>. Acesso em: 29/11/2022.

PMI - PROJECT MANAGEMENT INSTITUTE. **Guia PMBOK: Um Guia para o Conjunto de Conhecimentos em Gerenciamento de Projetos**, Sexta edição, Pennsylvania: PMI, 2017.

PRESSMAN, Roger S.; MAXIM, Bruce R. **Engenharia de software**. [AMGH, Porto Alegre]: Grupo A, 2021. E-book. ISBN 9786558040118. Disponível em:
<https://integrada.minhabiblioteca.com.br/#/books/9786558040118/>. Acesso em: 12 out. 2022.

PROJETOOLS Disponível em: <<http://www.projetoools.com.br/>> Acesso: em 03/10/2022

SOMMERVILLE, Ian. **Engenharia De Software**. 9ª. Edição. São Paulo: Pearson, 2019

SOUSA, Carolina Ribas de. **Procedimentos De Gestão Da Comunicação E Informação Para Escritórios De Arquitetura De Pequeno Porte**. 2010.