

ENGENHARIA DE SOFTWARE

ESTÁGIO SUPERVISIONADO

Daniel Menezes

PROJETO ORÇAMENTO

Projeto Final apresentado como requisito parcial para aprovação na disciplina de Estágio Supervisionado do Curso de Engenharia de Software, sob a orientação da professora Stephany Martins.

Palmas – TO
2025

SUMÁRIO

1.	1
2.	2
3.	3
4.	4
5.	10
6.	11

1. INTRODUÇÃO

O Estágio Supervisionado desenvolvido no curso de Engenharia de Software teve como área de concentração a análise e desenvolvimento de sistemas, com foco na melhoria de processos administrativos por meio da tecnologia da informação. O presente trabalho foi realizado a partir de uma demanda identificada no setor de Tecnologia da Informação de uma instituição de ensino superior, relacionada à dificuldade na elaboração e consolidação do orçamento anual de eventos e projetos institucionais.

Atualmente, o processo de planejamento orçamentário ocorre de forma descentralizada, onde cada setor da instituição elabora uma planilha própria contendo informações sobre seus projetos, justificativas e respectivos custos. Após essa etapa, um gestor de cada setor aprova os dados e encaminha a planilha para um responsável que realiza a unificação manual de todas as informações em um único documento final. Esse modelo apresenta limitações significativas, como retrabalho, maior possibilidade de erros, lentidão no processo e dificuldade de padronização das informações.

Diante desse cenário, surgiu a necessidade de propor uma solução tecnológica que otimizasse o fluxo de dados e tornasse o processo mais ágil e confiável. Assim, foi desenvolvido um sistema baseado em formulário único, capaz de atender às necessidades de todos os setores da instituição, coletar as informações de maneira padronizada e gerar automaticamente uma planilha unificada com todos os projetos submetidos.

A motivação para o desenvolvimento deste projeto está relacionada à aplicação prática dos conhecimentos adquiridos ao longo do curso, aliando teoria e prática na resolução de um problema real enfrentado pela instituição. A justificativa do trabalho fundamenta-se na importância da automação de processos administrativos, contribuindo para maior eficiência, organização e tomada de decisão por parte da gestão.

2. OBJETIVOS

O objetivo deste projeto consiste em aplicar os conhecimentos adquiridos ao longo da formação acadêmica, propondo uma solução eficiente para a organização dos orçamentos elaborados ao longo do ano letivo na instituição. Embora o modelo atualmente adotado seja funcional, ele apresenta limitações, como lentidão no processamento das informações e maior suscetibilidade a erros.

A solução proposta baseia-se no desenvolvimento de um formulário único, contendo todas as informações necessárias para a geração da planilha orçamentária, independentemente do setor solicitante. O sistema realiza o armazenamento dos dados dos projetos submetidos e converte automaticamente as informações coletadas em uma planilha do Excel, otimizando o processo de consolidação, análise e aprovação dos projetos.

3. DESCRIÇÃO DO LOCAL DE TRABALHO

O estágio supervisionado teve como local de referência a instituição de ensino superior UniCatólica, na fábrica de software (Primeiro andar - Bloco 1), uma sala designada para os estudantes da instituição utilizarem os computadores disponíveis no local para o desenvolvimento de competências técnicas e educacionais.

Inicialmente, o estágio estava previsto para ser realizado de forma presencial na própria instituição. No entanto, devido à incompatibilidade de horários entre os estagiários e o horário de funcionamento do setor, optou-se pela realização das atividades de forma remota. Dessa maneira, os integrantes do projeto desenvolveram suas atividades a partir de suas residências, utilizando ferramentas digitais para comunicação, desenvolvimento e versionamento do sistema.

Apesar do formato remoto, houve acompanhamento constante por parte do gestor de TI, o senhor Darley, que participou de reuniões virtuais periódicas. Nessas reuniões, foram discutidos os requisitos do sistema, realizadas validações das funcionalidades propostas e sugeridas melhorias alinhadas às necessidades reais da instituição. Esse modelo de trabalho possibilitou uma comunicação eficiente, mesmo à distância, garantindo a qualidade e aderência do projeto aos objetivos institucionais.

4. ATIVIDADES REALIZADAS

Nos primeiros dias criamos um grupo para facilitar o contato entre a equipe e o colaborador (Darley), após isso fizemos algumas reuniões para o mesmo explicar sobre o processo para a criação das planilhas orçamentárias e a principal dificuldade para ser resolvida, onde no caso foi a da demora para fazê-la, pois precisaria de procurar o modelo, substituir os dados célula por célula para, no final, salvar uma cópia da nova planilha, essa tarefa é feita em todos os setores.

A semana seguinte foi a escolha da tecnologia para ser utilizada, onde pensamos em algumas como python com Django, projetos node, projetos com typescript e react, mas o Darley sugeriu utilizarmos a ferramenta da microsoft chamada Microsoft SharePoint, então começamos a estudar e desenvolver nela.

1º dificuldade: A ferramenta da microsoft, além de ser algo relativamente novo para os membros da equipe, que não tinham experiência, acabou por ser extremamente limitado o seu uso, embora tenhamos insistido na ferramenta, não tivemos sinais bons de progresso.

Solução: Fizemos um protótipo básico utilizando a linguagem Python com o framework Django e apresentamos ao Darley, que foi compreensivo quanto a situação da dificuldade relatada anteriormente e ficou satisfeito com o protótipo desenvolvido pela equipe para exemplificar, a partir daí recomeçamos o projeto utilizando essas ferramentas.

```

MESES = [
    ('janeiro', 'Janeiro'),
    ('fevereiro', 'Fevereiro'),
    ('marco', 'Março'),
    ('abril', 'Abril'),
    ('maio', 'Maio'),
    ('junho', 'Junho'),
    ('julho', 'Julho'),
    ('agosto', 'Agosto'),
    ('setembro', 'Setembro'),
    ('outubro', 'Outubro'),
    ('novembro', 'Novembro'),
    ('dezembro', 'Dezembro'),

    # Campos mensais opcionais
    for mes, label in MESES:
        locals()[mes] = forms.DecimalField(
            label=label,
            required=False,
            min_value=0,
            widget=forms.TextInput(attrs={
                'placeholder': 'Digite o valor (ex: 1200)',
                'inputmode': 'decimal',
            })
        )

```

Lógica para adicionar os campos de mês e adicionar a informação no banco somente se o campo foi preenchido.

```

<script src="https://cdnjs.cloudflare.com/ajax/libs/select2/4.1.0-rc.0/js/select2.min.js"></script>
<script>
    $(document).ready(function() {
        $('#id_codigo_produto').select2({
            placeholder: 'Selecione as contas...',
            allowClear: true
        });
    });

```

Utilização da biblioteca django select para a criação do dropdown dinâmico e com múltipla seleção, pois o django não tem essa funcionalidade nativamente.

```

100, 19 seconds ago | 2 authors (Citriniti and one other)
<!DOCTYPE html>
{% load static %}
<html>
<head>
  <title>Criar Projeto</title>
  <link rel="stylesheet" href="{% static 'css/projetos_form.css' %}">
  <link href="https://code.jquery.com/jquery-3.6.0.min.js" rel="stylesheet" />
  <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
  <!-- Estilos movidos para projetos_form.css -->
</head>
<body>
  {% if edit %}
    <h1>Editar Projeto</h1>
  {% else %}
    <h1>Criar Projeto</h1>
  {% endif %}

  <form method="post">
    {% csrf_token %}
    <p>
      <label for="id_nome_projeto">Projeto / Atividade</label>
      {{ form.nome_projeto }}
    </p>

    <p>
      <label for="id_justificativa">Objetivo / Justificativa</label>
      {{ form.justificativa }}
    </p>
  </form>

```

Tela de formulário para criação de projetos, que recebia os widgets para a construção dos campos texto e dropdown para exibir na tela do usuário.

```

views.py  excel_constructor.py  projetos_form.html  settings.py  projetos_lista.html  projetos_form.js
main > views.py
1  from django.shortcuts import render, redirect
2  from django.http import HttpResponse
3  from .forms import ProjetoForm, CAMPOS_EXCLUIR_MESES
4  from .models import Projeto
5  from .excel_file.excel_constructor import create_projetos_excel_response
6  import pandas as pd
7
8  def criar_projeto(request):
9      if request.method == 'POST':
10         form = ProjetoForm(request.POST)
11         if form.is_valid():
12             instance = form.save()
13             # processar contas e valores enviados via POST
14             processar_contas_post(request.POST, instance)
15             return redirect('listar_projetos')
16         else:
17             form = ProjetoForm()
18
19         return render(request, 'page/projetos_form.html', {
20             'form': form,
21             'campos_excluir': CAMPOS_EXCLUIR_MESES,
22             'contas_data': []
23         })
24
25
26  def listar_projetos(request):
27     projetos = Projeto.objects.all()
28     return render(request, 'page/projetos_lista.html', {'projetos': projetos})
29

```



Criação das funções responsáveis de renderizar/construir as telas do sistema (lista de projeto e criar projetos), retornando os dados e urls necessários para cada página.

No dia seguinte a essa reunião a equipe separou as suas tarefas, Mateus Citriniti ficando responsável pelo desenvolvimento diretamente na tela, criando campos de entrada de texto, botões e os demais componentes, enquanto eu, Daniel Menezes, fiquei responsável por aplicar a lógica para guardar os dados inseridos na tela e inserir em células específicas do excel, além de criar o executável.

Nas semanas que se seguiram, fizemos um total de 3 reuniões, todas com a finalidade de apresentação do projeto até o ponto que foi desenvolvido, em todas houveram os devidos apontamentos, onde foi elogiado o que ficou funcional e colocado o que falta.

No dia 05/12, tivemos a última reunião online onde foi apresentado o sistema completo e tivemos a devida aprovação do Darley, assim só faltando ele fazer o devido processo para dar-nos a autorização de instalarmos (da forma que ele achar melhor) o sistema para utilização prática por parte dos setores.

Projetos Cadastrados

Nome do Projeto	Objetivo / justificativa	Tipo de Conta	Código do Produto	Valores Mensais	Total	É Recorrente?	Obrigação Legal?	Ações
teste 6	justificativa 2	Investimentos	5.1.2.1.01.02 - Copa e Alimentos	Novembro: R\$ 1000.00	R\$ 1000.00	Não	Não	 

Informações Adicionais
Nome do Gestor:

Setor:

📄 Exportar para Excel
➕ Novo Projeto

Tela inicial onde é possível visualizar os projetos criados localmente, além das opções “criar projeto” e “importar para excel”.

Criar Projeto

Projeto / Atividade

Objetivo /
Justificativa

Tipo de Conta

Conta Contábil
(para DGA)

Gastos Mensais

+ Janeiro

+ Fevereiro

+ Março

+ Abril

+ Maio

+ Junho

+ Julho

+ Agosto

The screenshot shows a web form for creating a project. It features four light gray input fields for the months 'Setembro', 'Outubro', 'Novembro', and 'Dezembro', each with a plus icon on the left. Below these are two dropdown menus: 'É Recorrente?' and 'Obrigação Legal?', both currently set to 'Não'. A 'Total:' label is followed by a wide, empty light gray input field. At the bottom is a prominent blue button labeled 'Salvar Projeto'.

Tela de criação, onde usuário preenche com as informações necessárias e salva os dados no banco, as informações cadastradas são exibidas na tela anterior “lista de projetos” os campos de mês são adaptáveis e o dropdown permite pesquisa e múltipla seleção.

No dia 10/12, Darley decidiu que queria uma forma simples de execução, onde não precisasse de implementar o projeto no sistema administrador da católica ou de um domínio próprio, então eu criei um arquivo .bat para execução direta no terminal.

```

planilha_orcamentaria_exe.bat X
planilha_orcamentaria_exe.bat
1  @echo off
2  REM Verifica se Python está instalado
3  where python >nul 2>nul
4  if errorlevel 1 (
5      echo Python nao encontrado! Instale o Python 3.13 antes de rodar este sistema.
6      pause
7      exit /b
8  )
9  REM Verifica se as bibliotecas essenciais estao instaladas
10 set PACKAGES=django openpyxl django_select2 django-crispy-forms
11 set NEED_INSTALL=0
12 for %%P in (%PACKAGES%) do (
13     python -c "import %%P" 2>nul
14     if errorlevel 1 (
15         echo Biblioteca '%%P' nao encontrada! Instalando dependencias...
16         set NEED_INSTALL=1
17     )
18 )
19 if %NEED_INSTALL%==1 (
20     pip install -r requirements.txt
21     if errorlevel 1 (
22         echo Falha ao instalar dependencias! Verifique o Python/pip.
23         pause
24         exit /b
25     )
26 )
27 REM Ativa o ambiente virtual, inicia o servidor Django e abre o navegador
28 cd /d %~dp0
29 start "Django" cmd /c ".\env\Scripts\activate.bat && python manage.py runserver"
30 REM Aguarda alguns segundos para o servidor iniciar
31 ping 127.0.0.1 -n 3 >nul
32 start http://127.0.0.1:8000/

```

Este código é do arquivo bat onde verifica se o python está instalado, verifica se os requisitos do sistema está instalado, ativa o ambiente virtual, inicia o servidor localhost pelo ping 127.0.0.1 e inicia o mesmo.

5. CONCLUSÃO

O desenvolvimento do presente projeto permitiu a aplicação prática dos conceitos aprendidos ao longo do curso de Engenharia de Software, proporcionando uma experiência enriquecedora tanto do ponto de vista técnico quanto profissional. A solução proposta atendeu ao objetivo principal de otimizar o processo de organização e unificação do orçamento institucional, tornando-o mais ágil, padronizado e menos suscetível a erros.

A substituição do modelo baseado em múltiplas planilhas independentes por um formulário único integrado demonstrou-se eficaz, pois reduziu significativamente o retrabalho e facilitou a consolidação dos dados em uma planilha final. Além disso, a nova abordagem contribui para uma melhor

organização das informações, agiliza o processo de aprovação e melhora a tomada de decisão por parte da gestão.

6. REFERÊNCIAS

Django documentation | Django documentation. Disponível em: <<https://docs.djangoproject.com/en/6.0>>.

Openpyxl - A Python library to read/write Excel 2010 xlsx/xlsm files — openpyxl 3.0.3 documentation. Disponível em: <<https://openpyxl.readthedocs.io/en/stable/>>.

Django-Select2 — Django-Select2 8.4.4.dev7+gbc49a5275 documentation. Disponível em: <<https://django-select2.readthedocs.io/en/latest/>>.

PANDAS. Python Data Analysis Library. Disponível em: <<https://pandas.pydata.org/>>.

APÊNDICES

Código fonte: https://github.com/DanielMenezes21/projeto_catolica

```

MESES = [
    ('janeiro', 'Janeiro'),
    ('fevereiro', 'Fevereiro'),
    ('marco', 'Março'),
    ('abril', 'Abril'),
    ('maio', 'Maio'),
    ('junho', 'Junho'),
    ('julho', 'Julho'),
    ('agosto', 'Agosto'),
    ('setembro', 'Setembro'),
    ('outubro', 'Outubro'),
    ('novembro', 'Novembro'),
    ('dezembro', 'Dezembro'),

    # Campos mensais opcionais
    for mes, label in MESES:
        locals()[mes] = forms.DecimalField(
            label=label,
            required=False,
            min_value=0,
            widget=forms.TextInput(attrs={
                'placeholder': 'Digite o valor (ex: 1200)',
                'inputmode': 'decimal',
            })
        )

```

Lógica para adicionar os campos de mês e adicionar a informação no banco somente se o campo foi preenchido.

```

<script src="https://cdnjs.cloudflare.com/ajax/libs/select2/4.1.0-rc.0/js/select2.min.js"></script>
<script>
    $(document).ready(function() {
        $('#id_codigo_produto').select2({
            placeholder: 'Selecione as contas...',
            allowClear: true
        });
    });

```

Utilização da biblioteca django select para a criação do dropdown dinâmico e com múltipla seleção, pois o django não tem essa funcionalidade nativamente.