



**UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

LUIGGY FERREIRA DIAS SILVA

**PLPPOOLWEB:
UM SISTEMA PARA SUPORTE A MONITORIA DA DISCIPLINA
PARADIGMAS DE LINGUAGENS DE PROGRAMAÇÃO**

CAMPINA GRANDE - PB

2023

LUIGGY FERREIRA DIAS SILVA

PLPPOOLWEB:

**UM SISTEMA PARA SUPORTE A MONITORIA DA DISCIPLINA
PARADIGMAS DE LINGUAGENS DE PROGRAMAÇÃO**

**Trabalho de Conclusão Curso
apresentado ao Curso Bacharelado em
Ciência da Computação do Centro de
Engenharia Elétrica e Informática da
Universidade Federal de Campina
Grande, como requisito parcial para
obtenção do título de Bacharel em
Ciência da Computação.**

Orientador: Professor Dr. Everton Leandro Galdino Alves.

CAMPINA GRANDE - PB

2023

LUIGGY FERREIRA DIAS SILVA

PLPPOOLWEB:

**UM SISTEMA PARA SUPORTE A MONITORIA DA DISCIPLINA
PARADIGMAS DE LINGUAGENS DE PROGRAMAÇÃO**

**Trabalho de Conclusão Curso
apresentado ao Curso Bacharelado em
Ciência da Computação do Centro de
Engenharia Elétrica e Informática da
Universidade Federal de Campina
Grande, como requisito parcial para
obtenção do título de Bacharel em
Ciência da Computação.**

BANCA EXAMINADORA:

**Professor Dr. Everton Leandro Galdino Alves
Orientador – UASC/CEEI/UFCG**

**Professor M.Sc. Roberto Medeiros de Faria
Examinador – UASC/CEEI/UFCG**

**Professor Dr. Tiago Lima Massoni
Professor da Disciplina TCC – UASC/CEEI/UFCG**

Trabalho aprovado em: 14 de Fevereiro de 2023.

CAMPINA GRANDE - PB

ABSTRACT

A professor's extra-class working time is usually dedicated to preparing classes, correcting tests, review of materials and research of teaching strategies. However, regularly, the amount of these activities can become notably demanding and negatively affect both the quality of the teaching and the professor's personal life. The subject of Paradigms of Programming Languages develops practical programming laboratories and the monitors elaborate questions that can be selected by the teacher. Currently, the questions are submitted on Github, but this is not advantageous both for the professor, as it makes it difficult to select questions, since there are no filters to search for questions of interest, as well as for the monitors, who do not have an easy way and practice of analyzing questions that were created in previous semesters. Based on these problems, this research proposed to develop a system to support professors and monitors of the Paradigms of Programming Languages subject at the Federal University of Campina Grande (UFCG, in Portuguese). This is the development of plpPoolWeb, an application that will allow the management of questions, notifications for monitors and other activities in a quick and agile way.

plpPoolWeb: Um Sistema para Suporte a Monitoria da Disciplina Paradigmas de Linguagens de Programação

Luiggy Ferreira Dias Silva

luiggy.silva@ccc.ufcg.edu.br

Universidade Federal de Campina Grande
Campina Grande, Paraíba, Brasil.

Orientador: Everton Leandro Galdino Alves

everton@computação.ufcg.edu.br

Universidade Federal de Campina Grande
Campina Grande, Paraíba, Brasil.

RESUMO

O tempo de trabalho extraclasse de um professor é normalmente dedicado a preparação de aulas, correção de provas, revisão de materiais e pesquisa de estratégias de ensino. Porém, em muitos momentos o volume destas atividades pode se tornar muito demandante e afetar negativamente tanto a qualidade do ensino quanto a vida pessoal do professor. A disciplina de Paradigmas de Linguagens de Programação trabalha laboratórios práticos de programação e monitores elaboram questões que podem ser selecionadas pelo professor. Atualmente, as questões são submetidas no Github, mas isso não é vantajoso tanto para o professor, pois dificulta sua seleção de questões, já que não existem filtros para buscar questões do seu interesse, como também para os monitores, que não tem uma maneira fácil e prática de analisar as questões que foram criadas em períodos anteriores. A partir desses problemas, o trabalho apresentado neste documento se propôs a desenvolver um sistema para dar suporte ao professor e monitores da disciplina de Paradigmas de Linguagens de Programação da Universidade Federal de Campina Grande (UFCG). Trata-se do desenvolvimento do plpPoolWeb, uma aplicação que permitirá o gerenciamento de questões, notificações para monitores e outras atividades de maneira rápida e ágil.

PALAVRAS-CHAVE

Suporte ao professor, Trabalho extraclasse, Monitores, Sistema Web.

REPOSITÓRIO

<http://github.com/LuiggySilva/plpPoolWeb-TCC>

1. INTRODUÇÃO

O tempo de trabalho extraclasse de um professor é o período em que o professor pode atender aos alunos individualmente ou em pequenos grupos, ajudando-os a compreender melhor os conteúdos abordados em sala de aula e a desenvolver habilidades importantes, como a resolução de problemas e a capacidade de trabalhar em equipe. Além disso, o tempo extraclasse é muito importante para o professor, pois é o momento em que ele pode dedicar-se à preparação de aulas, correção de provas e trabalhos, bem como à revisão de materiais e à pesquisa de novas estratégias de ensino.

Esse trabalho extra-classe pode ser bastante demandante e afetar negativamente tanto a qualidade do ensino quanto a vida pessoal do professor. Essas atividades exigem muito tempo e energia, o que pode prejudicar a sua capacidade de se dedicar integralmente às suas funções docentes durante o horário de aula. Além disso, essas tarefas também podem interferir na sua vida pessoal, uma vez que consomem tempo que deveria ser destinado à sua família, amigos e outros interesses pessoais [6].

Na disciplina de Paradigmas de Linguagens de Programação, do curso de Ciência da Computação da UFCG, são trabalhados laboratórios práticos de programação. Os monitores ajudam na elaboração de questões dentro dos prazos definidos pelo professor. Atualmente, esse processo é realizado com a submissão de questões dos monitores no Github [5], que é apenas um serviço de versionamento e hospedagem de código. Este processo não apresenta vantagens para os monitores, uma vez que não permite uma análise rápida e prática das questões que foram elaboradas anteriormente para evitar repetições. Além disso, a seleção das questões de interesse do professor também é dificultada, pois não existem maneiras práticas de buscar, filtrar e testar as questões elaboradas pelos monitores.

Sabendo disso, foi desenvolvido o **plpPoolWeb**, uma aplicação que tem como objetivo dar suporte ao professor e aos monitores da disciplina Paradigmas de Linguagens de Programação. A aplicação irá permitir que o professor realize as atividades de seleção de questões e a notificação de atividades para os monitores da disciplina de maneira rápida e ágil. Além disso, os monitores da disciplina vão ter a possibilidade de analisar as questões que já foram criadas anteriormente de maneira mais simples, com isso tendo uma melhor noção de que tipos de questões já existem e quais novas podem ser criadas.

2. SOLUÇÃO

Nesta seção, apresentaremos a descrição da solução proposta por esse trabalho, as telas da aplicação em funcionamento, funcionalidades e arquitetura da aplicação.

2.1 DESCRIÇÃO

O plpPoolWeb é um sistema web para o professor e os monitores da disciplina Paradigmas de Linguagens de Programação na Universidade Federal de Campina Grande (UFCG).

O sistema permite cadastrar monitores que podem realizar a submissão de questões, que envolvem a resolução de um problema com um código feito em um determinado paradigma de linguagem de programação [7]. Tanto o professor como o monitor podem realizar buscas e filtragens de questões por diversos parâmetros, permitindo que o monitor tenha uma noção de questões que já foram criadas e que o professor tenha mais rapidez e eficiência em selecionar questões que serão aplicadas para os alunos da disciplina. Também é possível que o professor teste as questões, ou seja, compile e execute as questões diretamente na aplicação, sem a necessidade de realizar esse procedimento em sua máquina localmente, além de fornecer feedback via e-mail dentro da aplicação. Além disso, há opções avançadas de gerenciamento, como gerenciamento de tags, que são usadas para indicar o conteúdo que é abordado no problema proposto pela questão e para realizar a filtragem das questões, monitores, períodos e atividades que são atribuídas aos monitores pelo professor (e.g., correção de projetos, criação de material complementar). O sistema também oferece backup de questões para garantir a segurança dos dados.

2.1.1 Telas do Monitor

Durante um período, o monitor da disciplina pode realizar o login na aplicação (Figura 1) e ter acesso a todas questões que já foram criadas em períodos anteriores, além disso, filtrar as questões com base no seu enunciado, período em que foi criada, tags, linguagem e tipo (Figura 2) e visualizar todos atributos de uma questão em específico (Figura 3). Além disso, o monitor pode visualizar todas atividades que lhe estão atribuídas juntamente dos seus deadlines, como também submeter, modificar ou remover questões na aplicação (Figura 4).

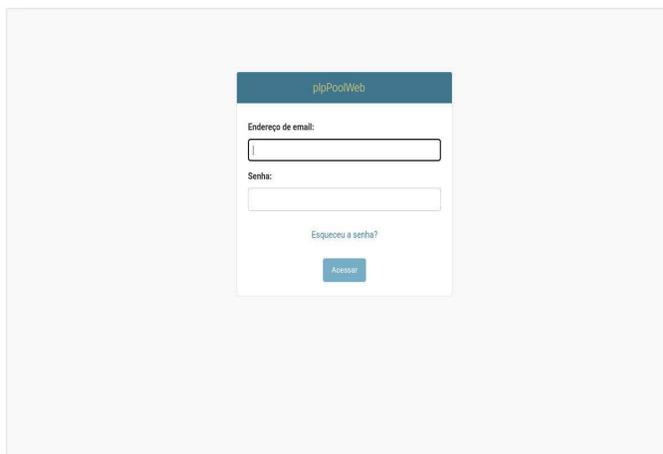


FIGURA 1: Tela de login do monitor:

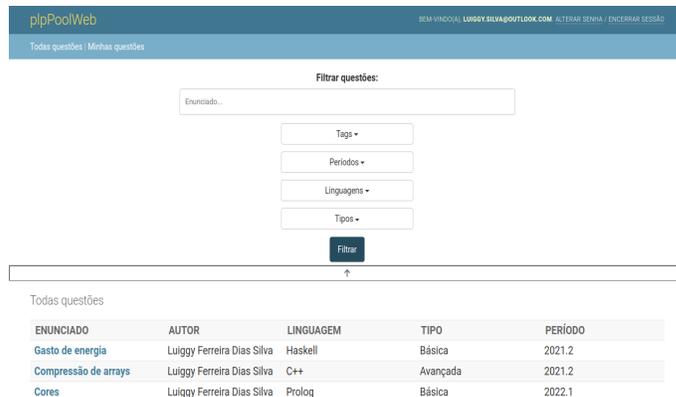


FIGURA 2: Tela de visualização de todas questões.

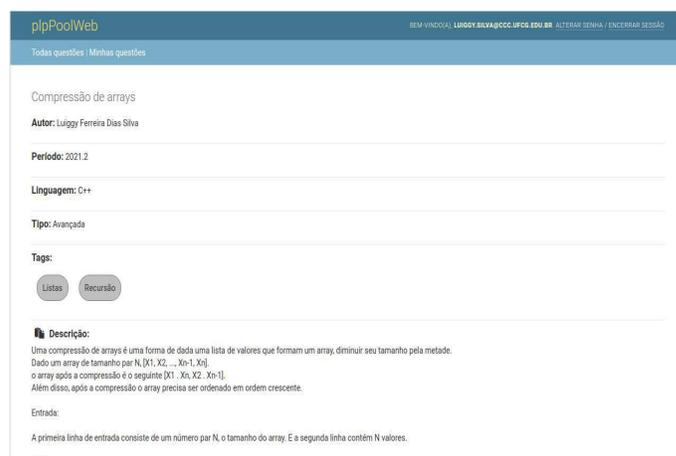
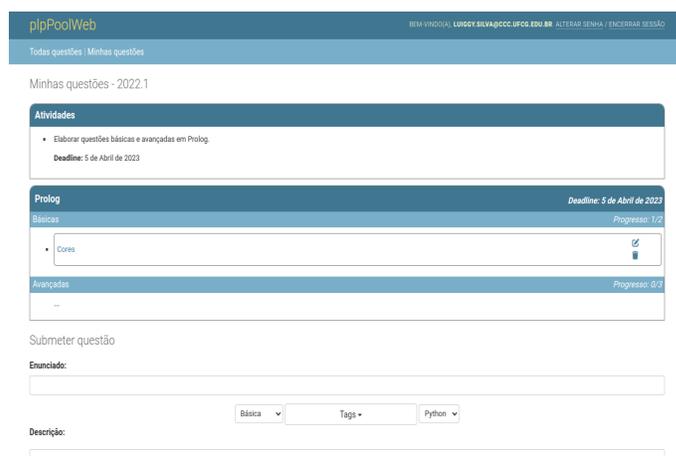


FIGURA 3: Tela de visualização de uma questão.



2.1.2 Telas do Professor

No início de um período, o professor da disciplina realizará o login na aplicação (Figura 4) e irá cadastrar os monitores da disciplina (Figura 5). Após realizar o cadastro dos monitores, o professor iniciará um novo período alocando os monitores atribuindo uma atividade atribuída a (e.g., proposição de duas questões básicas e três avançadas em Prolog (Figura 6 e Figura 7)). Depois dos monitores submeterem suas questões, o professor poderá buscar uma questão utilizando diversos filtros (Figura 8), escolhendo uma questão o professor ver todos seus atributos como também tem a opção de enviar feedbacks para o autor (Figura 9), compilar a questão passando uma entrada arbitrária (Figura 10) ou executar todos testes propostos pelo autor da questão (Figura 11).

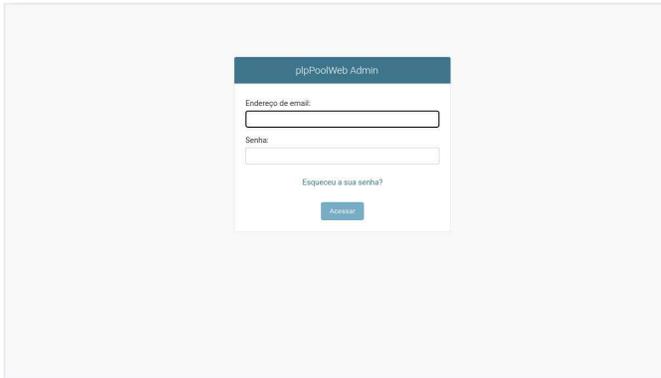


FIGURA 4: Tela de login do professor

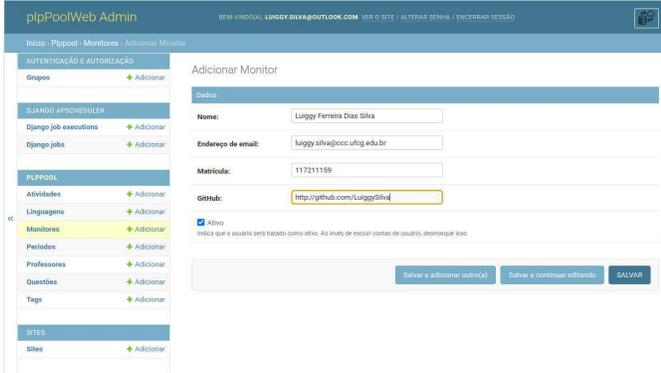


FIGURA 5: Tela de cadastro dos monitores.

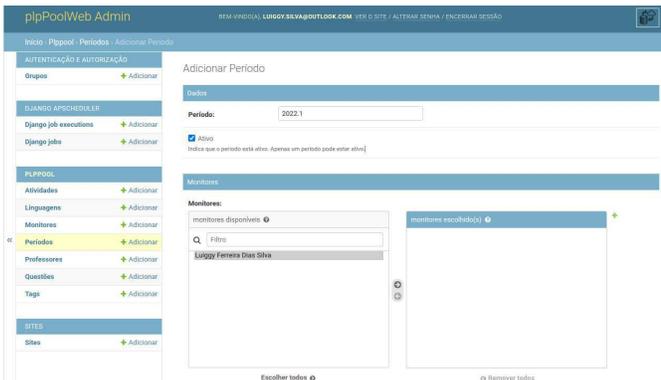


FIGURA 6: Tela de cadastro de um período.

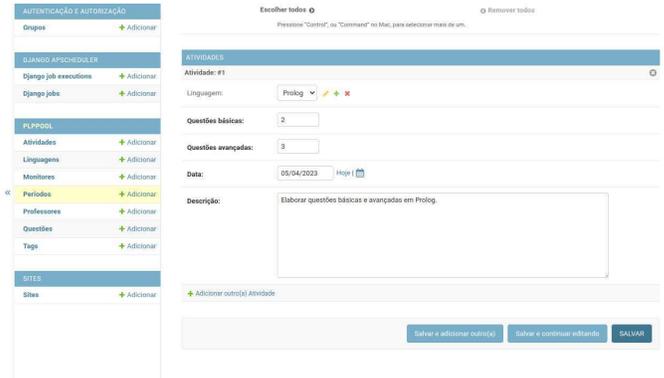


FIGURA 7: Atividades de um período.

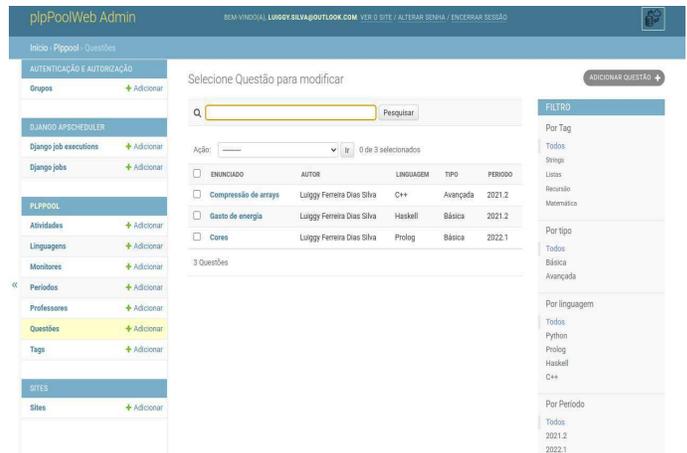


FIGURA 8: Tela de visualização das questões

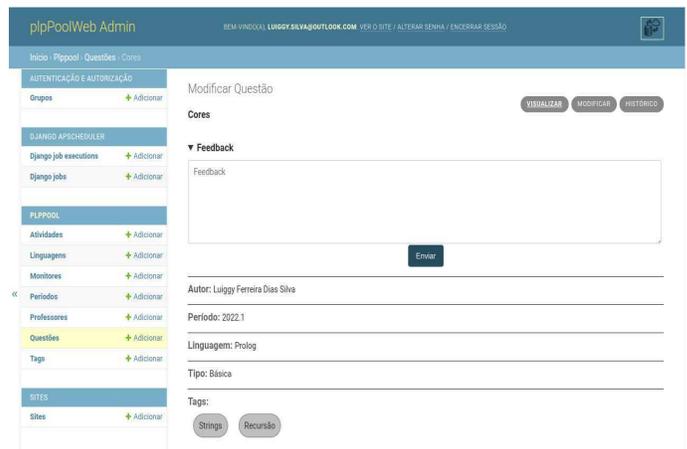


FIGURA 9: Tela de visualização de uma questão: feedback

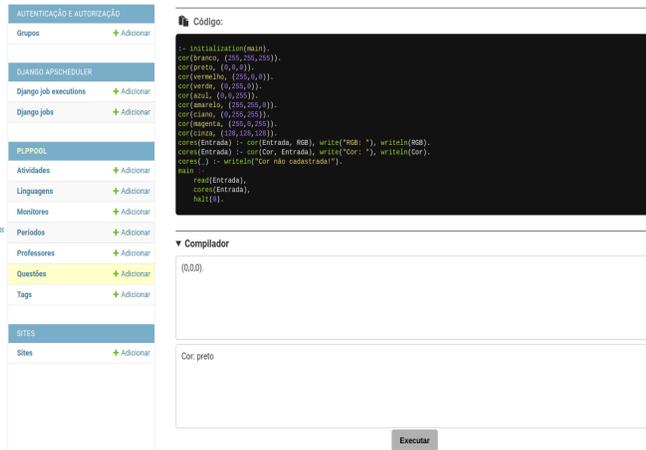


FIGURA 10: Tela de visualização de uma questão: compilador

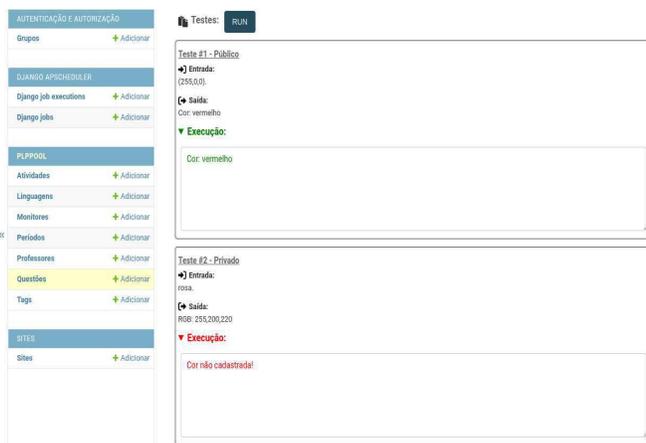


FIGURA 11: Tela de visualização de uma questão: testes

2.2 Funcionalidades

Nesta seção serão descritas as principais funcionalidades desta aplicação.

- **Login:** É possível que tanto o professor como os monitores realizem o login no sistema. A senha de ambos podem ser alteradas ou recuperadas via e-mail.
- **Filtragem de questões:** Tanto na tela inicial do monitor como na tela de questões do professor é possível realizar a filtragem de questões por título, tags associadas à questão, período em que a questão foi criada, tipo de questão e a linguagem que ela utiliza.
- **Submissão de questões:** Na tela de submissão de questões do monitor é possível submeter uma questão passando título, tags, descrição, código, testes, tipo de questão e linguagem.
- **Compilação de questões:** Na tela de visualização do professor além de visualizar e modificar, também é possível compilar o código da questão passando uma entrada arbitrária e obtendo a saída da sua execução. Também é possível executar todos os testes que foram propostos na questão e verificar suas saídas.

- **Feedback de questões:** Na tela de visualização de questão do professor é possível enviar um feedback sobre a questão para o email do monitor autor da questão.
- **Gerenciamento de questões:** Na tela de gerenciamento de monitores do professor é possível cadastrar um novo monitor passando nome, e-mail, matrícula e um flag que indica se o monitor está ativo. Também é possível atualizar os dados de um monitor ou removê-lo.
- **Gerenciamento de tags:** Na tela gerenciamento de tags do professor é possível cadastrar uma nova tag passando um nome. Também é possível atualizar os dados de uma tag ou removê-la.
- **Gerenciamento de professores:** Na tela gerenciamento de professores do professor é possível cadastrar um novo monitor passando um nome, matrícula e e-mail. Também é possível atualizar os dados de um monitor ou removê-lo.
- **Gerenciamento de monitores:** Na tela de monitores do professor é possível cadastrar um novo monitor passando um nome, matrícula, github e e-mail. Também é possível atualizar os dados de um monitor ou removê-lo.
- **Gerenciamento de períodos:** Na tela gerenciamento de períodos do professor é possível cadastrar um novo período passando um identificador, monitores, atividades e uma flag que indica se o período está ativo. Também é possível atualizar os dados de um período ou removê-lo.
- **Gerenciamento de atividades:** Na tela gerenciamento de atividades do professor é possível cadastrar uma nova atividade passando uma descrição, deadline e outros atributos opcionais. Também é possível atualizar os dados de uma atividade ou removê-la. Os monitores que estão alocados no período da atividade serão notificados automaticamente dias antes do deadline ser atingido.
- **Backup:** Para o professor é possível realizar o backup de todos dados que foram cadastrados na aplicação em um arquivo JSON, permitindo que possa recuperar todos dados com esse arquivo.

2.3 Arquitetura

Nesta seção apresentamos as principais decisões arquiteturais utilizadas para a criação do sistema para o problema proposto.

Para o desenvolvimento do projeto foi utilizado Django [1], que é um framework web Python [2] de alto nível que incentiva o desenvolvimento rápido e um design limpo e pragmático.

Um projeto Django possui como padrão de projeto o MTV (Model, Template, View), onde:

- **Model:** Representa o mapeamento dos dados da aplicação no banco de dados.
- **Template:** Representa a apresentação visual da aplicação, escrita em HTML.
- **View:** Lógica de negócio da aplicação, processando as solicitações dos usuários e fornecendo respostas.

O ciclo de uma requisição em Django (Figura 12) segue os seguintes passos:

1. Recebimento da requisição HTTP pelo servidor web.
2. Roteamento da requisição para URL correspondente na aplicação.
3. Execução da View associada à URL, processando a lógica de negócio e acessando o Model para obter ou armazenar dados.
4. Renderização do template HTML com os dados obtidos.
5. Envio da resposta HTTP com o conteúdo gerado para o cliente.

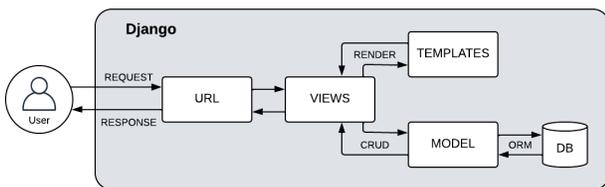


FIGURA 12: Ciclo de uma requisição no Django

O MTV do Django é uma abordagem que tem várias vantagens na criação de aplicações web. A separação clara de responsabilidades permite uma manutenção e escalabilidade mais fáceis do código, enquanto a reutilização de templates reduz o tempo de desenvolvimento. Além disso, o MTV permite a criação de módulos reutilizáveis, o que aumenta a eficiência do desenvolvimento. Por fim, o Django tem fácil integração com outras ferramentas, como o ORM e o formulário de validação, tornando o desenvolvimento mais fácil e eficiente.

A estrutura de arquivos padrão em um projeto Django (Figura 13) é composta por vários diretórios e arquivos, incluindo:

- Diretório do projeto: Este é o diretório principal que contém todos os arquivos relacionados ao projeto.
- Diretórios de aplicativos: Aqui ficam todos os aplicativos criados para o projeto, cada um com sua estrutura de arquivos própria.
- manage.py: Este é o script de gerenciamento do Django que permite a execução de comandos relacionados ao projeto.
- settings.py: Este arquivo contém todas as configurações do projeto, incluindo as configurações do banco de dados, tempo de execução, linguagem, entre outras.
- urls.py: Este arquivo define as rotas do projeto e controla como as URLs são tratadas pelo Django.
- wsgi.py: Este arquivo é responsável por fornecer uma interface entre o Django e o servidor web para garantir que as requisições sejam tratadas corretamente.
- static/: Este diretório armazena arquivos estáticos, como imagens, folhas de estilo CSS e arquivos JavaScript.
- templates/: Este diretório contém todos os modelos HTML usados pelo projeto.
- media/: Este diretório armazena arquivos enviados pelo usuário, como imagens e vídeos.

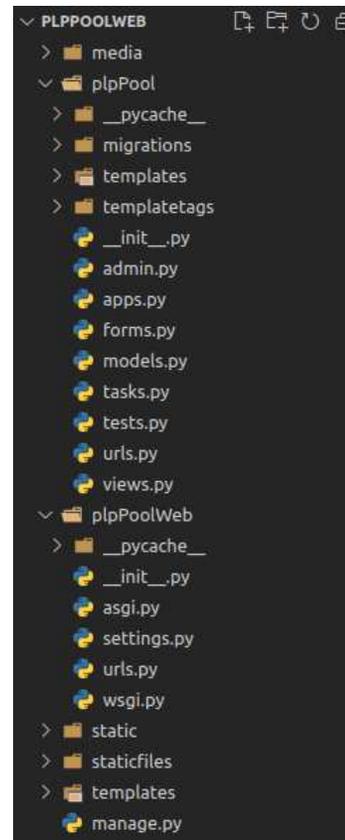


FIGURA 13: Organização da estrutura de diretórios

O banco de dados escolhido para armazenar e gerenciar os dados do sistema foi o PostgreSQL [3], um sistema de banco de dados relacional de código aberto conhecido por sua escalabilidade e desempenho. O Docker [4] foi utilizado para empacotar o projeto, ele é um software de código aberto usado para implantar aplicativos dentro de containers virtuais. Além disso, o código-fonte do sistema foi armazenado e gerenciado no Github [5], uma plataforma de hospedagem e versionamento de código fonte.

3. METODOLOGIA

Para o início do projeto, uma série de reuniões foram realizadas com o professor da disciplina de Paradigmas de Linguagens de Programação no curso de Ciência da Computação da UFCG. O objetivo principal dessas reuniões foi identificar todos os requisitos necessários para o sistema, a fim de garantir que ele atenda às necessidades do professor. Além disso, o desenvolvedor também contribuiu com sua experiência prévia como monitor da disciplina, fornecendo sugestões para a definição dos requisitos. Ao longo do desenvolvimento, alguns dos requisitos iniciais foram ajustados, de acordo com a evolução do projeto e com as necessidades que surgiram ao longo do caminho.

4. RESULTADOS

Referente a validação, a aplicação com todas as suas funcionalidades foi apresentada ao professor da disciplina. Duas questões negativas foram levantadas: o design visual da aplicação poderia ser melhorado e a falta de ambiente para os alunos da disciplina realizarem exercícios de programação. Por outro lado, pontos positivos foram levantados, tais como: a capacidade de atender às necessidades da disciplina, promover fácil interação entre o professor e monitores, e manter um histórico de questões úteis para estudo dos monitores. O professor planeja utilizar o sistema em próximos períodos.

5. EXPERIÊNCIA

Nesta seção serão descritas experiências proporcionadas pelo processo de desenvolvimento da aplicação, bem como os desafios, limitações e propostas para a continuidade e futuro.

5.1 Processo de desenvolvimento

No processo de desenvolvimento dessa aplicação, primeiramente, foram selecionadas as tecnologias que seriam utilizadas. Elas foram escolhidas considerando alguns critérios: domínio prévio das tecnologias pelo desenvolvedor, facilidade de trabalho para desenvolvimento e correções de eventuais problemas, e, por fim, permitir a manutenção e atualização de forma simples.

Em seguida, após os requisitos e funcionalidades do sistema serem estabelecidos com base nas necessidades apontadas pelo professor, o desenvolvimento seguiu gradativamente, construído por cada funcionalidade ao decorrer do período acadêmico.

5.2 Principais desafios e limitações

Os principais desafios deste trabalho foram lidar com a criação de formulários dinâmicos para a criação e modificação de questões no contexto do monitor. Além disso, a criação da funcionalidade de compilar o código das questões também foi um desafio. Inicialmente, foi pensado em tornar essa funcionalidade disponível para os monitores e o professor, mas, por conta da complexidade, ela está disponível apenas para o professor.

5.3 Trabalhos futuros

Existe um potencial para a continuação deste trabalho e melhorias que sigam a ideia principal do sistema. Abaixo seguem algumas melhorias que podem ser feitas:

- Registro do histórico de escolha de questões: Com esta funcionalidade, o professor poderá ter uma visão mais clara dos tipos de questões que ele tem selecionado e utilizar essas informações para orientar os monitores com relação às questões que deseja aplicar. Além disso, ela evitará a repetição de questões em períodos futuros;
- Ampliar a funcionalidade do compilador para os monitores: Esta funcionalidade permitirá que os monitores possam compilar e testar o código de suas questões diretamente na aplicação, garantindo a consistência dos resultados da compilação tanto na máquina do monitor quanto na aplicação. Além disso, simplifica o processo de modificação do código, evitando a necessidade de baixá-lo e compilá-lo localmente antes de fazer submissões;

- Criação de uma visão para os alunos da disciplina: Com essa nova visão, o professor da disciplina será capaz de registrar os alunos da disciplina, aplicar as questões escolhidas por ele e realizar avaliações dessas questões de maneira automatizada, tudo em um único ambiente, com isso o sistema cobriria grande parte dos aspectos práticos da disciplina.
- Melhoria no design visual da aplicação: Realizar melhorias na interface do monitor e professor em suas versões desktop e mobile para tornar a experiência do uso mais agradável em ambas plataformas.

6. CRESCIMENTO PESSOAL

Este trabalho foi desenvolvido com o objetivo de aplicar os conhecimentos adquiridos durante o curso de Ciência da Computação na Universidade Federal de Campina Grande.

Ao longo da elaboração dos requisitos e desenvolvimento do sistema, foi colocado em prática os conhecimentos adquiridos em disciplinas, tais como Programação I e II, Laboratório de Programação I e II, Engenharia de Software e Projeto de Software. Estas disciplinas foram cruciais para minha formação e capacitação para realizar este trabalho.

7. REFERÊNCIAS

- [1] Django, About Django Software Foundation. Disponível em: <<https://www.djangoproject.com/foundation/>>.
- [2] Python, About Python™. Disponível em: <<https://www.python.org/about/>>.
- [3] PostgreSQL, PostgreSQL: About. Disponível em: <<https://www.postgresql.org/about/>>.
- [4] Docker, Why Docker?. Disponível em: <<https://www.docker.com/why-docker/>>.
- [5] Github: Let's build from here. Disponível em: <<https://www.github.com/>>.
- [6] NUNES, Celia Maria Fernandes. TEMPO DE TRABALHO EXTRA-CLASSE, **Gestrado**, 2020. Disponível em: <<https://gestrado.net.br/verbetes/tempo-de-trabalho-extraclasse/>>.
- [7] SENA, Franciele. Paradigmas de Programação, **LinkedIn**, 2021. Disponível em: <<https://pt.linkedin.com/pulse/paradigmas-de-programa%C3%A7%C3%A3o-franciele-sena>>.

Sobre o autor:

Luiggy Ferreira Dias Silva é graduando de Ciência da Computação na Universidade Federal de Campina Grande e atualmente está no 10º semestre. (<https://github.com/LuiggySilva>)