

UNIVERSIDADE VIRTUAL DO ESTADO DE SÃO PAULO

Fabiana Cristina Gomez de Almeida Silva
João Robson Ramalho
Jefferson Batista Fogaça
Mario Fernandes Vasquez Junior
Murilo Leonel Pinton

SOS - Serviços

Taquarituba - SP
2023

UNIVERSIDADE VIRTUAL DO ESTADO DE SÃO PAULO

SOS - Serviços

Relatório Técnico-Científico apresentado na disciplina de Projeto Integrador para o curso de Ciência de Dados e Engenharia da Computação da Universidade Virtual do Estado de São Paulo (UNIVESP).

SILVA, Fabiana Cristina Gomez de Almeida; RAMALHO, João Robson; FOGAÇA, Jefferson Batista; JUNIOR, Mario Fernandes Vasquez; PINTON, Murilo Leonel. **SOS - Serviços**. 00f. Relatório Técnico-Científico. Ciência de Dados e Engenharia da Computação – **Universidade Virtual do Estado de São Paulo**. Tutor: (Isabela Wanderez Dutra Patriota). Polo...(Taquarituba), 2023.

RESUMO

O projeto em questão tem como objetivo criar uma aplicação web para cadastro de fluxos de serviços, tais como professores, pedreiros, pintores, entre outros, em uma região onde o emprego informal e pouco conhecimento técnico são predominantes. A melhoria visa atender novos desafios do projeto, incluindo a disponibilização de análise de dados por meio de gráficos simples e uma interface amigável e intuitiva para usuários com pouco conhecimento técnico. A arquitetura MTV do framework Django foi escolhida para o desenvolvimento do projeto, além das linguagens HTML, CSS e JavaScript para o desenvolvimento dos templates e programação orientada a objeto em Python para o desenvolvimento do modelo e das views. A integração com o banco de dados MySQL foi realizada para garantir a continuidade dos dados. O sistema CRUD foi implementado para permitir que os usuários realizem as operações básicas de criação, leitura, atualização e exclusão de registros. A aplicação permite o cadastro de pedidos de serviços, incluindo informações como nome, serviço oferecido, diária e descrição.

PALAVRAS-CHAVE: aplicação web, cadastro de fluxo de serviços, análise de dados.

SUMÁRIO

1 INTRODUÇÃO	1
2 DESENVOLVIMENTO.....	2
2.1 Objetivos.....	2
2.2 Justificativa e delimitação do problema	3
2.3 Fundamentação teórica	4
2.4 Metodologia.....	6
REFERÊNCIAS	8

1 INTRODUÇÃO

Este artigo descreve o processo de desenvolvimento de uma aplicação web que tem como objetivo permitir o cadastro de prestadores de serviços, como pintores, encanadores, professores entre outros serviços. A aplicação é desenvolvida utilizando várias ferramentas como a linguagem de programação Python, o framework Django, banco de dados MySQL, a linguagem de programação para a Web Java Script e GitHub para controle de versionamento.

Esta aplicação é uma melhoria em relação a um Projeto Integrador anterior, desenvolvido no quarto semestre da disciplina Projeto Integrador I e foi desenvolvida para enfrentar novos desafios do projeto. Um desses desafios é fornecer análise de dados para os usuários da aplicação, e isso será realizado com a biblioteca Plotly do Java Script.

Neste artigo, discutiremos os detalhes do projeto e a utilização das diferentes ferramentas para a construção da aplicação. Também abordaremos as vantagens do uso de cada ferramenta, e como elas trabalham juntas para fornecer uma aplicação web eficiente e fácil de usar.

O uso do Python e do Django permite o desenvolvimento rápido e eficiente de aplicações web, fornecendo um conjunto de ferramentas abrangente para criar recursos poderosos e extensíveis. O MySQL é um sistema de gerenciamento de banco de dados confiável e escalável, que garante a integridade e a segurança dos dados do projeto.

O JavaScript é uma linguagem de programação fundamental para criar interatividade em páginas da web e fornece recursos avançados para o usuário final. O Plotly é uma biblioteca JavaScript que fornece ferramentas de visualização de dados sofisticadas para tornar a análise de dados mais atraente e acessível.

Por fim, o uso do GitHub para controle de versionamento permite que a equipe de desenvolvimento trabalhe em colaboração, mantendo um registro preciso das alterações no código-fonte do projeto. Combinando essas tecnologias, a equipe pode criar uma aplicação web poderosa e escalável para gerenciar cadastros de prestadores de serviços e fornecer análise de dados para os usuários.

2 DESENVOLVIMENTO

2.1 Objetivos

A busca por profissionais autônomos para a realização de serviços em residências ou empresas é uma necessidade comum na vida moderna. No entanto, nem sempre é fácil encontrar pessoas qualificadas para realizar esses serviços, especialmente para aqueles que têm pouco conhecimento sobre o assunto. É nesse contexto que uma solução de aplicativo web de fácil utilização se torna essencial.

A importância de uma solução de aplicativo web de fácil utilização é evidente quando consideramos que muitos usuários não têm experiência em tecnologia ou em aplicativos de busca de serviços. Por isso, é importante que a interface do aplicativo seja intuitiva e fácil de usar, para que o usuário não se sinta desencorajado ou intimidado ao utilizá-lo.

Além disso, é importante que o aplicativo ofereça uma análise de dados dos usuários por gráficos simples. Isso porque muitos usuários podem não ter conhecimento em análise de dados e, por isso, precisam de uma forma clara e acessível de visualizar as informações relevantes. Gráficos simples e intuitivos podem ajudar o usuário a entender as informações de maneira rápida e fácil.

No caso de um aplicativo web que cadastra usuários que são prestadores de diversos serviços, como professores, pedreiros, pintores, entre outros, a análise de dados pode ser uma ferramenta muito útil para melhorar a qualidade do serviço oferecido. Por exemplo, ao analisar os dados de avaliação dos usuários sobre os prestadores de serviços, é possível identificar padrões e tendências que podem ser usados para melhorar o desempenho dos prestadores ou até mesmo para fornecer treinamento adicional.

Além disso, os gráficos podem ajudar os usuários a visualizar informações importantes, como o número de profissionais cadastrados em cada categoria de serviço, a média de preço para cada serviço, e outros dados relevantes para a escolha do prestador de serviço.

Em conclusão, uma solução de aplicativo web de fácil utilização é fundamental para garantir a acessibilidade e a qualidade dos serviços oferecidos. Além disso, a análise de dados por meio de gráficos simples pode ser uma ferramenta poderosa para melhorar a qualidade dos serviços prestados e fornecer informações valiosas aos usuários. Por isso, a adoção de tecnologias que oferecem interfaces intuitivas e análises de dados simples deve ser prioridade para empresas que buscam oferecer serviços de qualidade aos seus clientes.

2.2 Justificativa e delimitação do problema

O desenvolvimento de um aplicativo web que cadastra prestadores de serviços em uma região com pouco conhecimento técnico e alto índice de emprego informal é muito importante por diversas razões.

Primeiramente, o uso da tecnologia para conectar prestadores de serviços com clientes pode ajudar a melhorar a qualidade dos serviços oferecidos e, consequentemente, aumentar a confiança do público nos serviços prestados. Ao ter acesso a um banco de dados com informações sobre os prestadores de serviço, como suas habilidades e qualificações, os usuários podem tomar decisões mais informadas sobre qual prestador escolher. Isso pode levar a um aumento na satisfação do cliente e, por sua vez, na demanda pelos serviços.

Além disso, a tecnologia pode ser uma ferramenta poderosa para combater a informalidade e a falta de transparência no mercado de trabalho. Ao cadastrar prestadores de serviços em uma plataforma online, é possível registrar informações importantes, como suas habilidades, qualificações e histórico de trabalho. Isso pode ajudar a formalizar o mercado de trabalho na região e incentivar o desenvolvimento de novos negócios e oportunidades.

Outra razão pela qual o desenvolvimento de um aplicativo web é importante é que ele pode ajudar a melhorar a vida das pessoas que vivem na região, oferecendo novas oportunidades de trabalho e renda. Ao cadastrar prestadores de serviços na plataforma online, é possível conectar esses trabalhadores com novos clientes e oportunidades de trabalho que, de outra forma, poderiam não estar disponíveis.

O uso da tecnologia para conectar prestadores de serviços com clientes pode ajudar a democratizar o acesso a serviços de qualidade na região. Ao tornar o processo de busca e contratação de prestadores de serviços mais fácil e acessível, o aplicativo web pode ajudar a melhorar a vida de muitas pessoas, especialmente aquelas que têm menos conhecimento técnico e recursos financeiros limitados.

Em resumo, o desenvolvimento de um aplicativo web que cadastra prestadores de serviços em uma região com alto índice de emprego informal e pouco conhecimento técnico pode ser uma ferramenta poderosa para melhorar a qualidade dos serviços oferecidos, formalizar o mercado de trabalho, oferecer novas oportunidades de trabalho e renda, e democratizar o acesso a serviços de qualidade.

2. 3 Fundamentação teórica

O Com base nas informações do projeto, o desenvolvimento de uma aplicação web para o cadastro de retiradas de serviços em uma região com forte presença de emprego informal e pouco conhecimento técnico. O artigo abordará a arquitetura MTV (Model-Template-View) e o framework Django, que foram utilizados no desenvolvimento do projeto.

Django, um framework para criação de aplicações Web escrito em Python, criado em 2005 por um grupo de programadores do Lawrence Journal-World com a intenção de tornar mais rápido o desenvolvimento de aplicações Web. Este framework tornou-se conhecido por fornecer soluções para grande parte dos problemas tradicionais em desenvolvimentos Web, possuindo dezenas de tarefas comuns já prontas para serem reutilizadas, como por exemplo autenticação de usuário, administração de conteúdo, mapas de site, entre outras. (Santiago, Veras, Amaral, 2020, p.10)

O Django é um framework web de alto nível que utiliza a arquitetura MTV para desenvolvimento de aplicações em Python. A arquitetura MTV é semelhante à arquitetura MVC (Model-View-Controller), mas com algumas diferenças importantes. No MTV, o Model é responsável pela camada de dados, o Template pela camada de apresentação e o View pela camada de lógica de negócios. Essa arquitetura torna o desenvolvimento da aplicação mais organizado, escalável e fácil de manter.

A arquitetura do Django corresponde a uma variação do padrão Model-ViewController (MVC), utilizado por outros frameworks Web desenvolvidos em diversas linguagens, como Ruby on Rails (Ruby), Spring MVC (Java), ASP.NET MVC (C#) e Zend Framework (PHP). O Django pode ser frequentemente considerado um framework MVC, porém algumas diferenças conceituais fazem com que a nomenclatura mais adequada, no caso do Django, seja “MTV” (Model-Template-View). (Fernades, 2015, p.35)

Para o desenvolvimento dos templates, foram utilizadas as linguagens HTML, CSS e JavaScript. O HTML é a linguagem padrão para criar páginas web, o CSS é utilizado para definir a aparência visual dos elementos da página e o JavaScript é uma linguagem de programação utilizada para adicionar interatividade e dinamismo ao site. Essas ferramentas foram essenciais para a criação de um layout agradável e de fácil utilização para os usuários.

Esta é a camada de apresentação, onde as informações são visualizadas pelos usuários. Um template consiste de partes estáticas do arquivo HTML de saída e de partes com uma sintaxe especial que descrevem como o conteúdo dinâmico será apresentado. O Django tem um caminho de pesquisa para templates, o qual permite minimizar a redundância entre eles. Geralmente, uma view recupera dados de acordo com os parâmetros de pesquisa, carrega um template e o renderiza com os dados recuperados [Foundation 2020a]. (Santiago, Veras, Amaral, 2020, p.11)

A programação orientada a objeto em Python foi utilizada para o desenvolvimento do Model. Essa abordagem permite organizar o código em classes e objetos, o que torna o desenvolvimento mais modular, reutilizável e fácil de manter. O Model é responsável por gerenciar os dados da aplicação, sendo utilizado para fazer operações no banco de dados e fornecer dados para as Views.

Django já vem com uma solução para mapeamento objeto-relacional (ORM, do inglês Object-Relational Mapping) no qual o esquema do banco de dados é descrito em código Python [Foundation 2020a]. Neste caso, abstrações em Python podem ser usadas para criar consultas complexas sem que seja necessário realizar ações diretas no banco. (Santiago, Veras, Amaral, 2020, p.10)

Por fim, a lógica para criação das Views foi desenvolvida em Python. As Views são responsáveis por receber como aceitação do usuário, processar os dados e retornar como respostas adequadas. As Views trabalham em conjunto com os Templates para apresentar os dados de forma clara e organizada para o usuário.

As views recebem a informação e o tipo da requisição (“POST” ou “GET”) do lado do cliente e, em seguida, formatam os dados para que sejam armazenados no banco através dos models da camada Model. As views também se comunicam via models com o banco para recuperar dados que são transferidos posteriormente aos templates, para a visualização do usuário. Cada view é responsável por fazer uma entre duas coisas: devolver um objeto `HTTPResponse` contendo o conteúdo para a página requisitada ou levantar uma exceção como `Http404` [Foundation 2020a]. (Santiago, Veras, Amaral, 2020, p.11)

Com o uso dessas ferramentas e metodologias, foi possível desenvolver uma aplicação web eficiente, organizada e de fácil utilização para os usuários. O Django, em conjunto com as outras tecnologias, permitiu o desenvolvimento rápido e eficiente da aplicação, além de facilitar a manutenção e aprimoramento da mesma.

O sistema CRUD é uma abordagem comum para a implementação de operações básicas em bancos de dados, que incluem: Create (Criação), Read (Leitura), Update (Atualização) e Delete (Exclusão). Na implementação do projeto de cadastro de prestadores de serviços, o sistema CRUD foi utilizado para permitir a criação de novos cadastros, a leitura de informações existentes, a atualização de dados já cadastrados e a exclusão de informações que não sejam mais relevantes. Por meio das views do Django, foi possível criar as funcionalidades necessárias para cada uma dessas operações e integrá-las com o banco de dados MySQL. Dessa forma, os usuários da aplicação puderam realizar essas operações de forma simples e intuitiva, permitindo uma gestão eficiente dos cadastros de prestadores de serviços.

2.4 Metodologia

O primeiro passo para o desenvolvimento da aplicação foi a escolha da arquitetura e do framework. Optamos pelo framework Django, que utiliza a arquitetura MTV para o desenvolvimento de aplicações web em Python. Essa escolha se deveu à facilidade de uso, à grande comunidade de desenvolvedores e à vasta documentação disponível na internet.

Em seguida, iniciamos o desenvolvimento dos templates utilizando as linguagens HTML, CSS e JavaScript. Começamos definindo a estrutura básica da página usando HTML e em seguida adicionamos os estilos visuais com CSS. Para tornar a interface mais interativa e dinâmica, adicionamos funcionalidades utilizando JavaScript.

Com os templates prontos, começou o desenvolvimento do Model programação utilizando orientada a objeto em Python. Criamos classes para cada tabela do banco de dados e implementamos métodos para realizar as operações de inserção, atualização e exclusão de dados.

Com o Model pronto, passamos para lógica das Views. Criamos as Views em Python e definimos as rotas e os métodos HTTP necessários para cada funcionalidade da aplicação. Em seguida, associamos cada View a um Template para apresentar as informações de forma organizada e clara para o usuário.

Por fim, implementamos o sistema de autenticação utilizando o Django Authentication Framework. Criamos uma tela de login para os usuários e definimos os níveis de acesso para cada tipo de usuário.

Ao longo do desenvolvimento da aplicação, utilizamos o Git para controle de versionamento e o GitHub para armazenar o código fonte da aplicação. Com isso, participe de forma colaborativa e segura, mantendo um histórico de todas as alterações realizadas no código. Essa parte teve um cuidado especial, foi criada algumas ramificações, pois se algo desse errado no decorrer do projeto, essas ramificações seria nosso ponto de controle para mudar o rumo do projeto caso a implementação se mostrasse muito complexa.

Com o uso dessas ferramentas e metodologias, desenvolvemos uma aplicação web eficiente, organizada e de fácil utilização para os usuários. O Django, em conjunto com as outras tecnologias, permitiu o desenvolvimento rápido e eficiente da aplicação, além de facilitar a manutenção e aprimoramento da mesma.

A implementação do sistema CRUD permitiu a criação de cadastros de prestadores de serviços com diversas informações relevantes. Por exemplo, ao realizar a operação de criação,

os usuários puderam informar dados como nome, serviço prestado, valor da diária e descrição do serviço oferecido.

Por meio da operação de leitura, os usuários tiveram acesso a essas informações e puderam visualizar os cadastros existentes de prestadores de serviços, podendo filtrá-los por serviço, nome ou qualquer outra informação cadastrada.

A operação de atualização permitiu aos usuários modificar informações em cadastros já existentes, como alterar o valor da diária ou atualizar a descrição do serviço.

Por fim, a operação de exclusão possibilitou a remoção de informações que não fossem mais relevantes ou que tenham sido cadastradas erroneamente.

Dessa forma, o sistema CRUD implementado no projeto de cadastro de prestadores de serviços permitiu que os usuários pudessem gerenciar as informações cadastradas de forma eficiente, tornando o processo de gestão mais simples e ágil.

REFERÊNCIAS

Tecnoblog. O que é Django? Disponível em: <https://tecnoblog.net/414088/o-que-e-django/> . Acesso em: 22 abr. 2023.

Treinaweb. Entendendo o MTV do Django. Disponível em: <https://www.treinaweb.com.br/blog/entendendo-o-mtv-do-django/> . Acesso em: 22 abr. 2023.

ABNT – Associação Brasileira de Normas Técnicas. **NBR 14724**: Informação e documentação. Trabalhos Acadêmicos - Apresentação. Rio de Janeiro: ABNT, 2002.

SANTIAGO, Cynthia Pinheiro et al. Desenvolvimento de sistemas Web orientado a reuso com Python, Django e Bootstrap. **Sociedade Brasileira de Computação**, 2020.

FERNANDES, Rodrigo de Oliveira. **Desenvolvimento de uma plataforma de bookmarking social com Django e TDD**. 2015. Trabalho de Conclusão de Curso. Universidade Tecnológica Federal do Paraná.