

Segundo trabalho de Linguagens de Programação

Professor Claudio Miceli

Thiago R. Mendonça Andre A. Schlaepfer, Thiago L. Almeida

¹Centro de Ciências Matemáticas e da Natureza – Universidade Federal do Rio de Janeiro (UFRJ)
RJ – Brazil

thiagomendonca2801@poli.ufrj.br

Abstract. *Second Programming Language course activity, which involves creating a better version of the previous presented website. This project uses Flask framework for python. Bug fix, error treatment and a new logout button were made. This report will be focused on Flask, since the code was developed by me.*

Resumo. *Segundo trabalho da aula de Linguagens de Programação, que envolve o aprimoramento do website criado para o primeiro trabalho. Nesse projeto foi utilizado o framework Flask para Python. Foi feita correção de bugs, tratamento de erros e inserção de um botão para logout. Esse relatório terá ênfase na parte do Flask, visto que o código foi desenvolvido por mim.*

1. Introdução

O objetivo do website é promover uma melhoria no sistema de agendamento de refeições atual do restaurante universitário do Centro de Tecnologia da UFRJ. Desde a implantação do sistema atual, alunos reclamam frequentemente do prejuízo causado por outros que faltam seus agendamentos. A melhoria proposta evita esses prejuízos, como o desperdício de comida e o impedimento de outros utilizarem o restaurante, liberando as vagas dos que não compareceram.

O website conta com um sistema de registro, login, agendamento e administração através de um login exclusivo, tudo isso armazenado em um banco de dados relacional, o SQLite.

2. Código

O código é dividido em 2 partes, o back-end e o front-end. O back-end utiliza a linguagem Python aplicada ao framework Flask. Já o front-end utiliza a linguagem de marcação HTML e a linguagem de estilo CSS.

2.1. MVC

O padrão MVC separa a arquitetura do software em 3 partes: modelo, visão e controle. O controle é encarregado de lidar com as requisições do usuário através de ações estabelecidas no código e usar os modelos para buscar informações necessárias no banco de dados. Essas informações são tratadas e carregadas na visão do usuário, no caso de um website, a tela do navegador.

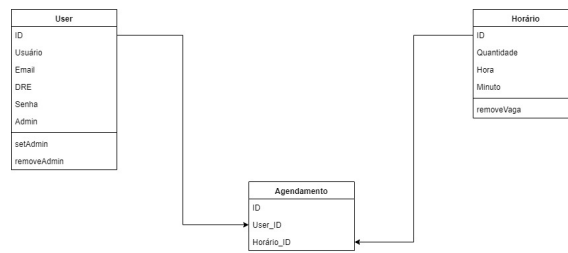


Figura 1. Diagrama UML do modelo

2.2. O paradigma cliente-servidor

O cliente faz uma solicitação a um servidor utilizando o protocolo HTTP, que é interpretado e devolvido pelo servidor em forma de array, novamente utilizando HTTP.

3. A visão final do usuário

Através da aplicação do HTML e CSS foi possível criar uma interface limpa e amigável ao usuário.

3.1. Login

Para utilizar o website, o usuário é obrigado a se identificar através de login, sendo encaminhado a tela responsável. Nessa tela pode ser feito o login ou se o utilizador não possuir conta, basta criar clicando no link de cadastro. O formulário preenchido é enviado ao servidor para que o Flask compare as informações com o banco de dados e aprove ou não o login.

The screenshot shows a login form with an orange header containing the word "Login". Below the header, there are two input fields: one labeled "DRE" and another labeled "Senha". Below these fields is an orange button labeled "Entrar". At the bottom of the form, there is a link that says "Não possui uma conta? [Cadastre-se](#)".

Figura 2. Tela de login

3.2. Cadastro

Caso o usuário não possua conta, pode clicar em "Cadastre-se" na tela de login para ser encaminhado a um formulário para preenchimento e cadastro. Ao concluir, os dados são enviados ao servidor para armazenagem.

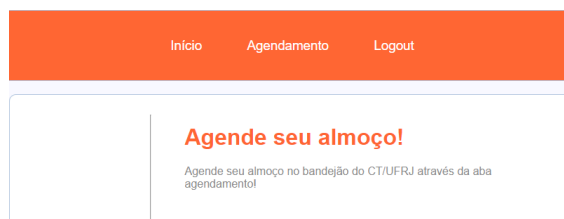


O formulário de cadastro possui um cabeçalho laranja com o título "Cadastro" em branco. Abaixo, há campos de entrada para "Usuário", "Email", "DRE", "Senha" e "Confirmação de senha", todos com bordas arredondadas. Um botão laranja "Cadastrar" está posicionado abaixo dos campos. No rodapé, há o texto "Já é um membro?" seguido por um link azul "Fazer Login".

Figura 3. Tela de cadastro

3.3. Início

Após autenticação do usuário, a tela de início pode ser acessada. Essa tela pode ser utilizada para exibir notícias e informações importantes sobre o restaurante universitário. A partir dela é possível navegar para outras páginas utilizando a barra de navegação superior.



A tela de início possui uma barra de navegação laranja no topo com os links "Início", "Agendamento" e "Logout" em branco. Abaixo, há uma seção com o título "Agende seu almoço!" em laranja. Abaixo do título, há um texto menor: "Agende seu almoço no bandeirão do CT/UFRJ através da aba agendamento!".

Figura 4. Tela de início

3.4. Agendamento

Ao acessar a página de agendamento, o usuário é capaz de visualizar os horários disponíveis e escolher o que mais lhe agrada, podendo somente agendar uma vez ao dia. O horário selecionado é enviado ao servidor que armazena e exibe na página de administração o agendamento criado.

Início Agendamento Logout

Agendamento

Para realizar o agendamento, informe seus dados abaixo:

Horário
Selecione o horário ▼

Agendar

Figura 5. Tela de agendamento

3.5. Administração

A administração é feita através de um login exclusivo ao gerente do sistema. Ao acessar esse login, o usuário encontra uma nova opção na barra de navegação, a página de administração. Nessa página são mostrados os agendamentos feitos pelos utilizadores do restaurante com seus respectivos horários e DRE. Ao clicar em um agendamento, esse é concretizado e some da página. Se o sistema detectar que existe um agendamento não concretizado em um horário passado ao atual, esse contará as ocorrências e mostrará na parte superior.

Início Agendamento Administração Logout

Faltaram: 1

12345
16.00

Figura 6. Tela de administração

4. Conclusão

O objetivo do projeto foi alcançado. Ao implementar o website desenvolvido, benefícios à comunidade escolar serão alcançados. Essa segunda versão conta com melhorias de bug e tratamento de erros.