UNIVERSIDADE FEDERAL DO RIO DE JANEIRO
ESCOLA POLITÉCNICA

RELATÓRIO - TRABALHO 1
Disciplina: Linguagens de Programação

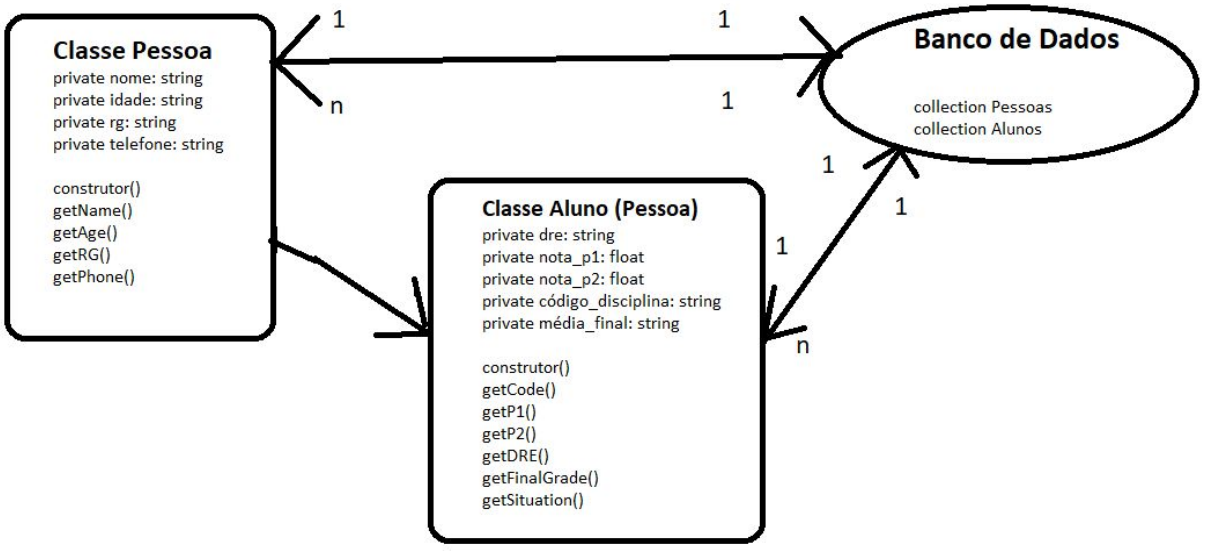
Aluno: William Barbosa de Macedo
DRE: 113090099
Prof: Claudio Miceli

OUTUBRO/2018

INTRODUÇÃO

O presente trabalho destina-se a implementar uma Lista de contatos, onde estes são categorizados em dois grupos, que se relacionam de forma hereditária: Pessoas e Alunos. O trabalho foi realizado em linguagem Python, com o uso do editor de texto Atom e o ambiente de desenvolvimento foi uma máquina com o sistema operacional Ubuntu 18.04. A seguir no diagrama do modelo ER, observa-se como os grupos citados acima se relacionam, juntamente com seus métodos e atributos.

MODELO ENTIDADE-RELACIONAMENTO



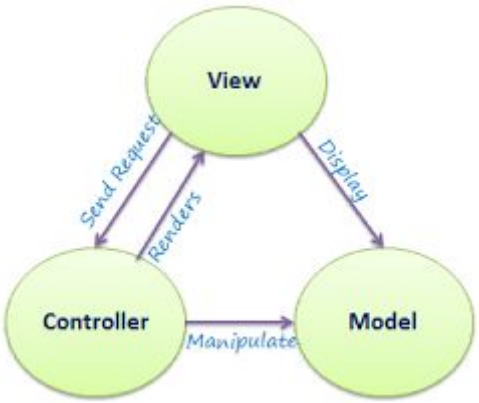
O modelo ER possui duas classes principais: Pessoa e Aluno, onde Aluno é classe-filha de Pessoa, herdando seus métodos e atributos. Por sua vez, Aluno possui métodos e atributos que o caracterizam de forma mais fidedigna, especificando ainda mais as suas características, não só como Pessoa mas efetivamente como Aluno.

A maioria dos atributos das classes foram configurados como **privados**, para só poderem ser modificados por suas respectivas classes. Dessa forma, o acesso a tais atributos é implementado através de métodos *getters*.

Objetos das classes Aluno e Pessoa são inseridos e removidos no Banco de Dados que possui 2 collections. Uma para cada classe. Cada collection pode ser interpretado como um array de objetos das suas respectivas classes.

ESTRUTURA MVC

O código pode ser analisado seguindo a estrutura Modelo-Visão-Controle. O framework utilizado foi o Flask.



- No controller é feito toda a parte de recebimento das requisições HTTP e tomada de decisões. Essas decisões acionam o model.
- No model, são instanciados objetos que são inseridos e removidos no banco de dados, conforme a operação realizada. Este por sua vez, devolve ao controller, os dados que são necessários para a renderização das views.
- As views são páginas HTML alimentadas por dados variáveis (renderizadas conforme a solicitação do usuário).

VIEWS UTILIZADAS

Foram 3 arquivos HTML para serem renderizados com os dados provenientes do banco de dados.

1. home.html

Esta é a homepage do site, onde é disponibilizado para o usuário o conteúdo atual das duas listas (de alunos e de pessoas). Além disso, após cada inserção/remoção, o usuário é redirecionado para esta página, onde pode visualizar o efeito da sua ação imediatamente.

2. insert.html

Esta é a página onde foram inseridos formulários que recebem os dados característicos de cada entidade (Alunos e Pessoas). Estes formulários enviam os dados para o Backend via requisição HTTP (método POST).

3. remove.html

Nesta página, o usuário pode remover unidades específicas de cada lista. Para a entidade Aluno, o seletor é o DRE. Já para a entidade Pessoa, o seletor é o RG.

FOLHA DE ESTILO DO SITE

Para a estilização das páginas, foi utilizado o framework **BULMA.** A escolha foi motivada pelo fato de possuir boa documentação, com bons exemplos e ilustrações, além de que é um framework bem leve e compacto, se comparado com outros frameworks, como por exemplo o Bootstrap.

BANCO DE DADOS

Para o armazenamento dos dados foi utilizado o Banco de Dados não-relacional MongoDB (No-SQL), através do pacote PyMongo. A escolha foi motivada por ser um banco de dados que o autor já utilizou previamente e que é de uso fácil e com boa documentação. Foi utilizado o site **mlab.com** para ser o hospedeiro do banco de dados na nuvem. Dessa forma, os dados ficam mais protegidos do que no caso de serem armazenados na máquina local.

Duas collections foram implementadas para armazenar cada tipo de dado (Alunos e Pessoas).

ROTAS

Foram utilizadas rotas de nomes intuitivos e sem extensão, seguindo uma tendência que vem se estabelecendo no segmento de desenvolvimento WEB, facilitando a memorização do usuário.

- “/” - rota raiz, exibe a tabela com os dados presentes no banco de dados atualmente
- “/insert” - rota que exibe a página com os formulários para inserção de novos objetos ao banco de dados
- “/remove” - rota que exibe a página para remoção de objetos
- “/in1” - rota para o método POST, que recebe os dados (para objetos da entidade Pessoa) provenientes do usuário e os direciona para o model. Após isso, redireciona para a homepage.
- “/in2” - semelhante a “/in1”, porém para a entidade Aluno.
- “/out1” - semelhante a “/in1”, porém para remoção de dados.
- “/out2” - semelhante a “ou1”, porém para a entidade Aluno.

AMBIENTE VIRTUAL E *PACKAGES*

A documentação padrão do framework Flask recomenda que o desenvolvimento seja feito através de um ambiente virtual, de forma que os pacotes necessários para os arquivos estejam garantidos e assim se dificulte problemas futuros com atualização de novas versões e possíveis incompatibilidades.

Os pacotes utilizados/instalados foram:

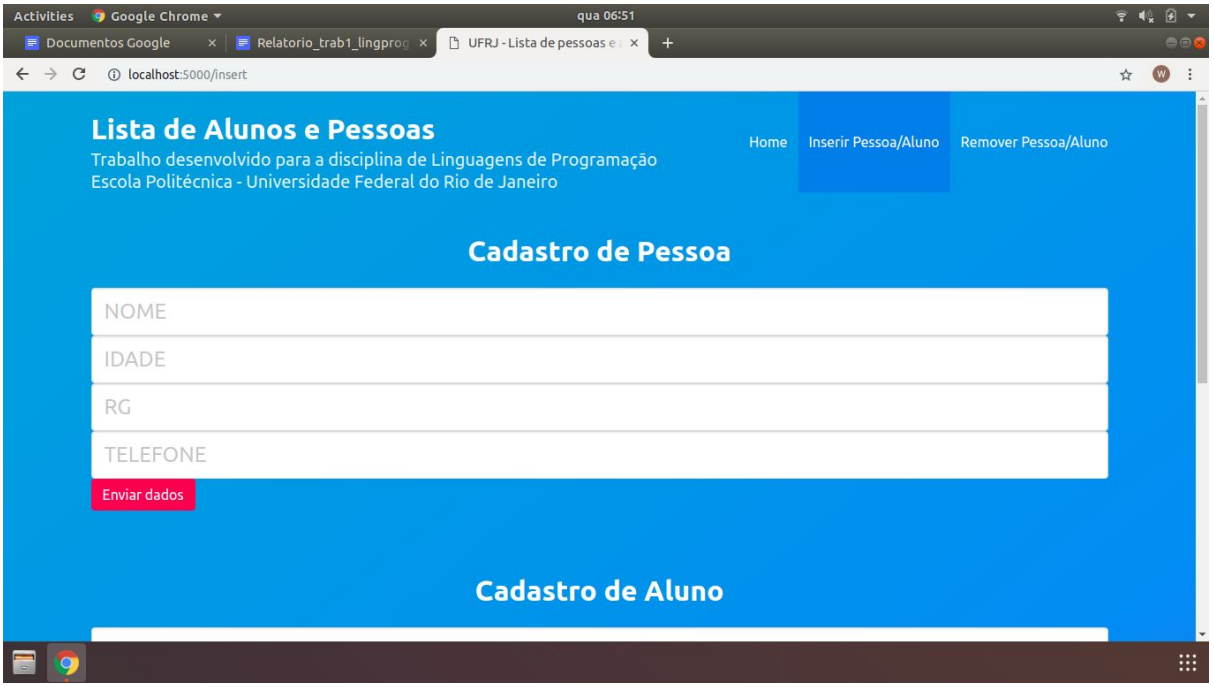
- flask
- flask_pymongo

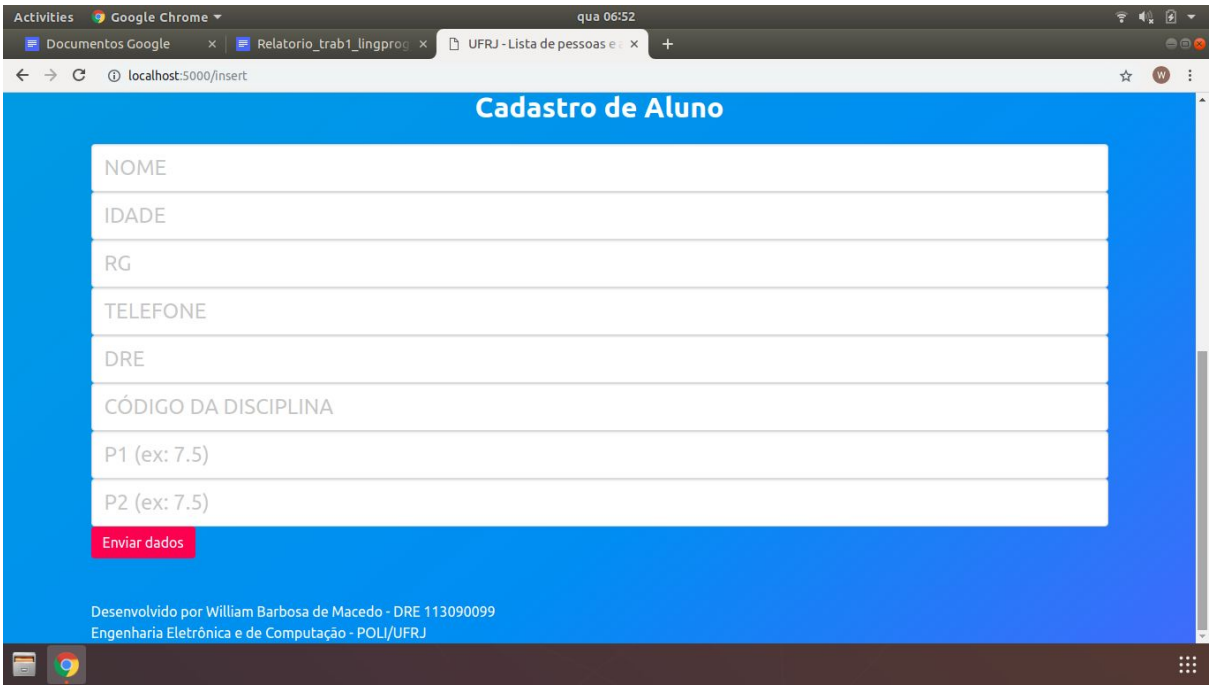
VISUALIZAÇÃO DO SITE

Nos printscreens abaixo, visualizamos como ficou o conteúdo das páginas.

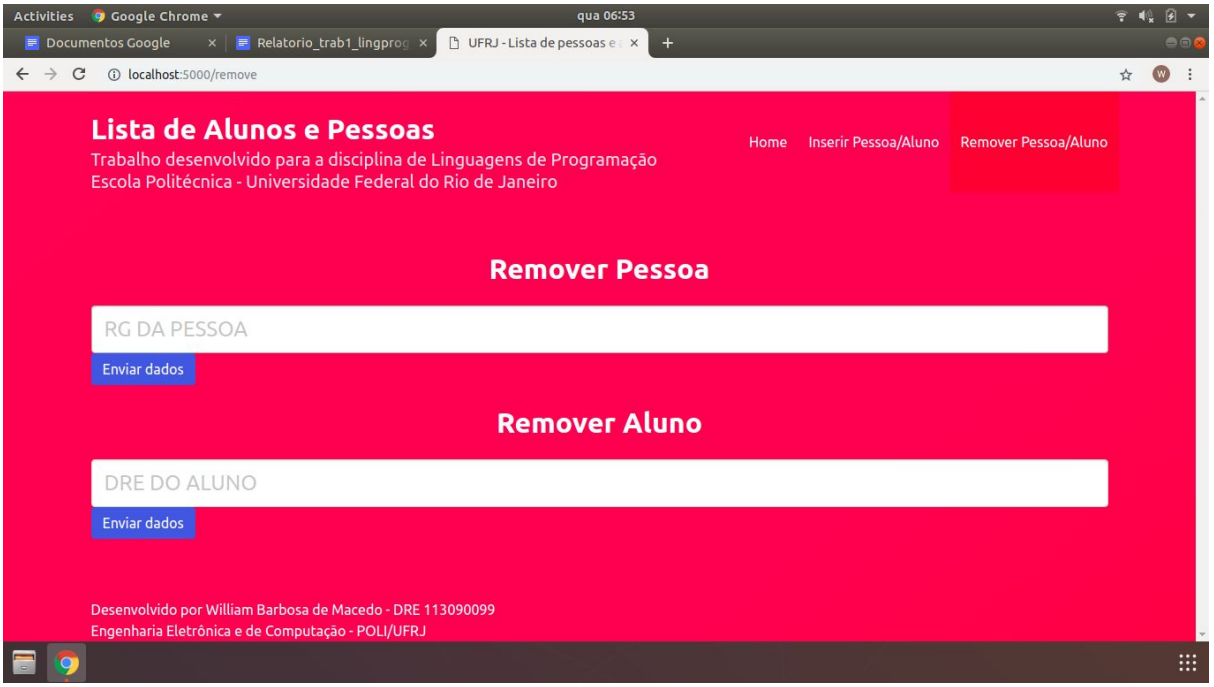
- homepage

- insert


Activities
Google Chrome
qua 06:51
Documentos Google
Relatorio_trab1_lingprog
UFRJ - Lista de pessoas e
localhost:5000/insert
Lista de Alunos e Pessoas
Trabalho desenvolvido para a disciplina de Linguagens de Programação
Escola Politécnica - Universidade Federal do Rio de Janeiro
Home
Inserir Pessoa/Aluno
Remover Pessoa/Aluno
Cadastro de Pessoa
NOME
IDADE
RG
TELEFONE
Enviar dados
Cadastro de Aluno


Activities
Google Chrome
qua 06:52
Documentos Google
Relatorio_trab1_lingprog
UFRJ - Lista de pessoas e
localhost:5000/insert
Cadastro de Aluno
NOME
IDADE
RG
TELEFONE
DRE
CÓDIGO DA DISCIPLINA
P1 (ex: 7.5)
P2 (ex: 7.5)
Enviar dados
Desenvolvido por William Barbosa de Macedo - DRE 113090099
Engenharia Eletrônica e de Computação - POLI/UFRJ

- remove



REPOSITÓRIO

Os códigos do trabalho estão disponíveis em:
https://github.com/wbm99/Trabalhos_LingProg_2018.2