Modelagem e Simulação

Segundo trabalho de programação
Modelagem e Simulação de um sistema de filas
Aluno: Arthur Borges

Relatório:

PYTHON

- **Forms.py**: módulo guarda o formulário da página contendo os campos relativos a todos os tipos de simulações
- **Models.py:** módulo onde estão presentes as classes que especificam os tipos de distribuições e simulações disponíveis, bem como seus atributos.
- **Simulation.py**: módulo que faz a simulação
    a. ***get_simulated_data***: retorna uma iteração de uma simulação o que corresponde a uma linha na tabela de simulação
    b. ***get_summary***: compila os dados gerados pela função acima, gerando um estatísticas que serão mostradas na tabela de sumário
    c. ***get_simulation_data***: gera as várias linhas e o sumário da simulação
    d. ***get_general_summary***: compila os dados de todas as simulações realizadas
    e. ***get_value_uniform*** e ***get_value_exponential***: retornam valores aleatórios nas distribuições uniform e exponencial, respectivamente. O valor em exponencial é gerado usando um valor aleatório uniforme como no método da transformada inversa.
- **App.py**: módulo que dá início a aplicação web e que contém as rotas disponíveis. É responsável por instanciar as classes de modelo com base no formulário recebido, além de chamar as funções para realizar a simulação

HTML

- **layout.html.j2**: layout geral da página
- **home.html.j2:** template da página principal, com o formulário
- **random.html.j2**: template da página de simulação do tipo aleatória, funciona para ambas distribuições

CSS

Estilos da página: bootstrap-4.3.1, master.css e simstyle.css

JAVASCRIPT

Bibliotecas utilizadas:

- Bootstrap-4.3.1
- Jquery-3.4.1
- Plotly-1.48.3
- Popper-1.15.0

Módulos:

- **master.js**: javascript da página principal, apenas controle de visibilidade
- **tabslider.js**: módulo para deixar barra de navegação de simulação responsiva, implementando o arraste
- **simulation.js**: módulo responsável por fazer as requisições ao servidor e provido da resposta (todos os dados de uma simulação) renderizar a página para o usuário como: a tabela de simulação, o sumário da simulação, os gráficos de simulação e o sumário geral da simulação

Manual:

Requisitos:

- Python >= 3.5
- Numpy - pip3 install numpy
- Flask - pip3 install flask
    - Flask_wtf - pip3 install flask_wtf
- Wtforms - pip3 install wtforms

Inicializando o programa:

- Depois de todos os requisitos instalados abra o terminal no local da pasta onde está o programa
- Execute com python app.py (python versão >=3.5)
- No navegador acesse: localhost:5000 para exibir a página principal

Rodando o programa:

- Escolha uma das distribuições disponíveis e escolha o número de clientes
- Para a distribuição escolhida informe todos os parâmetros necessários e clique em Simular
- Na página de simulação você pode escolher o intervalo de simulação (padrão 1s) e o número de simulações (padrão 20 simulações)
- Clique em simular para dar início as simulações
- Os gráficos serão gerados em tempo real para o acompanhamento

- As tabelas e estatísticas sobre cada simulação também serão geradas e podem ser acessadas pelas abas de simulação
- Ao final de todas simulações um relatório será exibido contendo estatísticas úteis sobre o processo de simulação