

## DESENVOLVIMENTO DE UMA APLICAÇÃO DESKTOP DE ACESSO WEB COM DESIGN RESPONSIVO UTILIZANDO *FRAMEWORK* COM FLASK E PYQT5

Artur Cantu de Oliveira<sup>1</sup>, José Rafael Pilan<sup>2</sup>

<sup>1</sup>Graduando do Curso de Análise e Desenvolvimento de Sistemas – Fatec Botucatu.  
artur.ca.oliveira@gmail.com

<sup>2</sup>Professor Mestre – Fatec Botucatu.

### RESUMO

Em razão da evolução das tecnologias de desenvolvimento *web* e devido ao crescimento do mercado *mobile*, surgiu a necessidade de se ter *sites* com *layouts* dinâmicos que apresentem diferentes formas de visualização. Com o surgimento de ferramentas voltadas para o desenvolvimento de *sites* responsivos, como *media queries* e *CSS Grid*, percebe-se que o foco de criação de ferramentas voltadas para sistemas responsivos é a *web* e não *softwares desktop*, embora aplicações *standalone* também possam acessar serviços desenvolvidos de modo responsivo através de recursos disponibilizados por ferramentas como o PYQT5 que simulam um web browser. O objetivo do presente trabalho é demonstrar uma maneira de se desenvolver uma aplicação *desktop* que acessa um *website* local com *layout* responsivo, com a utilização do Python, Flask, PyQt5 e um banco de dados MariaDB. O Python, em conjunto a seus módulos, abre um *Web Server*, uma janela e controla o banco de dados. O *layout* do sistema se dá a partir de arquivos HTML, CSS, JS e, como resultado, as telas funcionam em *displays* de diferentes tamanhos, sem ter que alterar a estrutura do HTML. A partir deste trabalho é compreendido que aplicações *desktop* podem ser desenvolvidas com a ideia de serem responsivas e que há a possibilidade de se criar diferentes sistemas seguindo este método.

**Palavras-chave:** Python. *Web Application*. *Responsive Design*.

### ABSTRACT

#### *DEVELOPMENT OF RESPONSIVE DESKTOP SOFTWARE WITH FLASK*

*Due to evolution of web development technologies as well as growth of the mobile market, a need for websites with responsive layouts that provide different ways of being visualized emerged. Owing to the advent of responsive website development tools, such as the Media*



*mainly for the web and not for desktop software, although standalone applications can also access responsive webservices. The objective of this study is to present a way to develop a desktop application that access a local website with responsive layout by using Python, Flask, PyQt5 and a MariaDB database. Python, together with its modules, opens a Web Server, a window, and manages the database. The layout of the system is created based on HTML, CSS, JS files and consequently the pages work on different size displays without having to change the HTML structure. Based on this study, it is understood that desktop applications can be developed responsively and that it is possible to create a framework using such method.*

**Keywords:** Python. *Web Application. Responsive Design.*

## 1 INTRODUÇÃO

O desenvolvimento de um *website*, seja ele responsivo ou não, vem recebendo mais e mais ferramentas, como o *CSS Grid* e *media queries*, que tem como objetivo auxiliar em um projeto voltado para a *web*, porém quando falamos de aplicações *desktop* esta ideia é raramente vista.

Com o passar do tempo, e com a evolução dos dispositivos móveis, foi visto a necessidade de se ter uma ferramenta que torna a utilização de *websites* em *mobiles* mais viável, isto é deixar o *site* mais organizado e mais bem estruturado para um *display* pequeno, e arrumar uma solução para a ideia de desenvolver *sites* que funcionem somente para iPhones (MARCOTTE et al., 2010).

Uma maneira de solucionar esse problema é a utilização de tecnologias *web* que sejam acessadas por uma aplicação *desktop*. Assim os *layouts* responsivos são usados para *desktop*, lembrando que um modelo que fica otimizado em um *desktop* não ficará necessariamente otimizado em um *notebook* e vice-versa.

O *layout* da aplicação desenvolvida para este trabalho, foi gerada com *templates* feitos em HTML, CSS e Java Script. Desta forma o desenvolvedor possui toda a liberdade de criação de um *website*, porém terá como resultado um *site* local ou uma aplicação híbrida (que é o programa ter sua própria janela de acesso ao serviço).

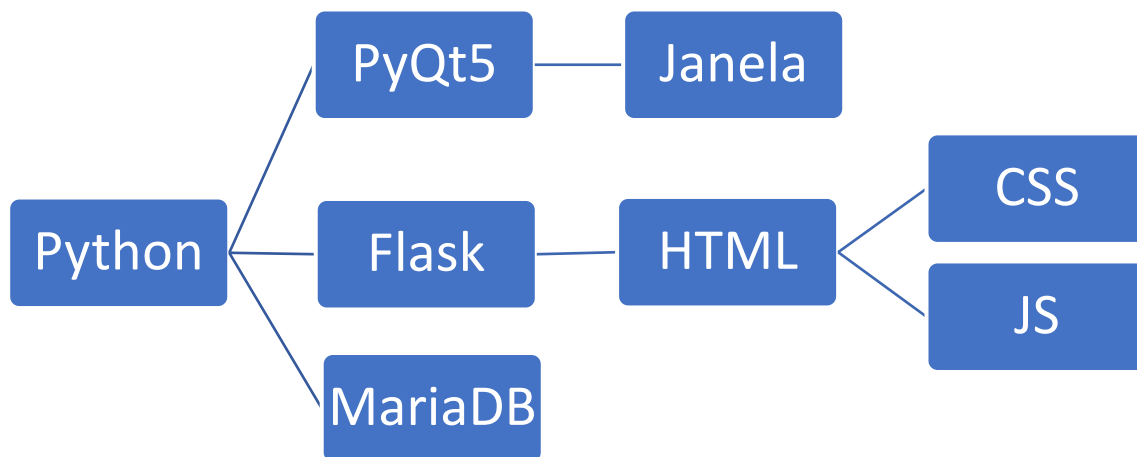
Tendo isso em vista e utilizando duas bibliotecas em Python (Flask e PyQt5), o objetivo do presente trabalho é o desenvolvimento de um *software desktop* de acesso a um

operacional, por exemplo, o acesso a um banco de dados MariaDB.

## 2 MATERIAL E MÉTODOS

Na figura 1 é demonstrado através de um diagrama a interação das ferramentas para o desenvolvimento do projeto. Como demonstrado o Python é responsável por gerenciar as telas utilizando o PyQt5 para abrir uma janela e o Flask que faz o *setup* de um servidor *web*, que possibilita a dinâmica do HTML com CSS e JavaScript e o Python realiza a ligação ao banco de dados, assim como outras ações no nível do sistema operacional, por exemplo acesso a um banco MariaDB.

Figura 1 Fluxograma do sistema.



Fonte: O autor, 2018.

### 2.1 Python

O Python por ser uma linguagem de programação interpretada, pode rodar em qualquer plataforma seja Linux, Windows, Mac OS ou até mesmo em Android com algumas *builds* não oficiais. Isso torna o desenvolvimento de aplicações multiplataformas mais acessível. Sua sintaxe em si é uma de suas melhores características, que é deixar o código facilmente legível a nível humano. Permite desenvolver de modo imperativo, orientado a objetos ou funcional e possui tipagem dinâmica e forte. Além do mais, possui uma vasta quantidade de módulos/bibliotecas que já são *built in*, e se preciso de bibliotecas de terceiros basta utilizar o pip, um módulo *built in* do Python, para instala-las, outro exemplo de uma biblioteca interna é o *multiprocessing* que é utilizada para se atingir o objetivo de se executar duas funções simultaneamente neste projeto. (PYTHON FOUNDATION, 2018).

## 2.2 Flask



7ª Jornada Científica e Tecnológica da Fatec de Botucatu  
29 de Outubro a 01 de Novembro de 2018, Botucatu - São Paulo, Brasil



Segundo Ronacher (2018) Flask é um *microframework* para Python baseado no Werkzeug, Jinja 2 e boas práticas. É usado para abrir um *web server*, que pode ser gerenciado por URLs customizadas, onde cada uma aponta para uma anotação no *script* em Python que executará sua devida função, sendo que cada função possui as definições de uma página ou ações padronizadas, assim como o tratamento de GET e POST *requests*, pelo fato de fazer parte do escopo de um *script* Python pode-se fazer alterações em nível de sistema operacional, o acesso a um banco de dados por exemplo.

## 2.3 PyQt5

PyQt5 é um módulo de interface gráfica para Python que utiliza do *framework* Qt que é suportado para uma variedade de plataformas incluindo Windows, OS X, Linux, IOS e Android. Este módulo possui mais de mil classes, seja para abrir uma janela simples ou a criação de uma *web browser*, que é utilizado como a *display* principal do programa. Atualmente a versão 5 é mantida pela *Riverbank Computing Limited* sob duas licenças, a GPL e uma de uso comercial (RIVERBANK COMPUTING LIMITED, 2018).

## 2.4 HTML

HTML é a sigla para *Hyper Text Markup Language*, ou seja, uma linguagem de marcação de hipertexto. Basicamente uma linguagem que nos permite criar estruturas para uma aplicação *web* possibilitando descrever e definir o conteúdo de um *site* assim como seu *layout*. E com a nova versão, o HTML5, foram adicionados novos elementos e tiveram vários aspectos melhorados como, Semântica, Conectividade a servidores, Gráficos 2D e 3D, performance e estilização.

## 2.5 JavaScript

O JavaScript é uma linguagem interpretada pelo próprio navegador, tem a intenção de tornar uma página simples, sem muitas funcionalidades, em uma página rica em funcionalidade e/ou comportamento, seja síncrono ou assíncrono (MILLS et al., 2018).

## 2.6 CSS

CSS é a sigla para *Cascade Styling Sheet*, uma folha de estilo em cascata. Esta ferramenta suportada por *web browsers*, faz com que o site ou esta aplicação *desktop*, possa ter mais cores. O HTML em si pode se tornar uma grande bagunça, porém o HTML pode ser muito mais elegante, e mais direto (simples de se entender), com a utilização do CSS Grid, que tem como conceito o *grid layout*.

e verticais onde um dos conjuntos define as colunas e o outro as linhas. Elementos podem ser colocados sobre o *grid*, respeitando suas colunas e linhas. E juntamente com o *media query* do CSS há a dinâmica de se reestruturar o *layout* de um *grid*, podendo inverter o *footer* com o *header* de uma página sem sequer tocar na estrutura do HTML.

Figura 2 Demonstração do poder do CSS Grid.



Fonte: Adaptado de Graham (2017).

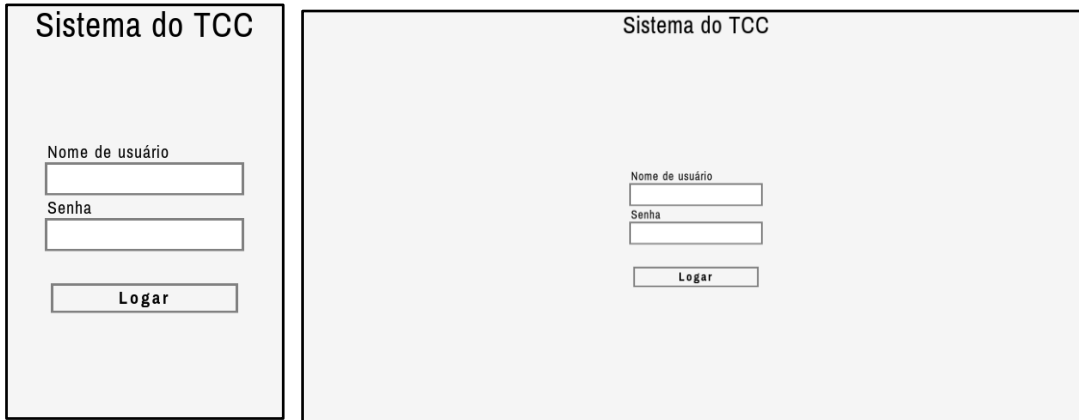
## 2.7 MariaDB

Criado pelos desenvolvedores do MySQL e mantido pela MariaDB *Foundation*, é uma base de dados relacional *open source* que transforma dados em informações estruturadas. É utilizado pela sua rapidez, escalabilidade, robustez e por possuir uma grande variedade de *engines* de armazenamento assim como *plugins* e várias outras ferramentas fazendo-o útil para diversos casos distintos (MARIADB FOUNDATION, 2018). É utilizado para criar e armazenar os dados de um banco de RG.

## 3 RESULTADOS E DISCUSSÕES

Durante o desenvolvimento de uma aplicação de visualização de um banco de dados utilizando tal método, é notado que o design das telas deve seguir o mesmo conceito aplicado a concepção de um *website*, que seria o *mobile first*, que implica que o *layout* para celulares seja definido antes do de *desktop*. Outra questão relevante é que cada tipo de visão terá estruturas distintas, assim tendo a necessidade de adaptar o JavaScript.

O sistema foi feito de modo bem simples (sem muitos componentes) e minimalista como demonstrado na tela de *login*, pedindo somente o usuário e senha do banco de dados (esse sistema foi idealizado com uma base um banco de dados local). A figura (a) demonstra em uma tela *mobile* enquanto a figura (b) é uma tela *desktop*.

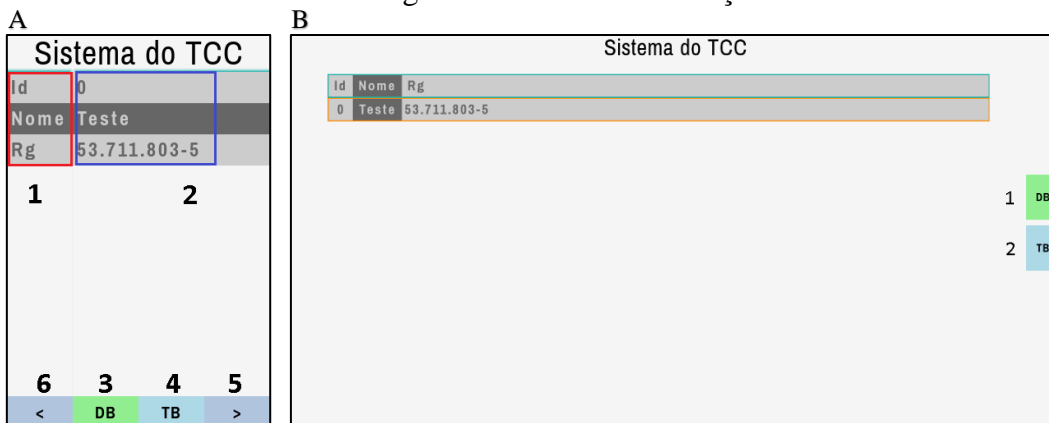


Fonte: O Autor, 2018.

A segunda tela é representada de duas maneiras sendo-as *mobile* (a) e *desktop* (b) a tela (a) apresenta o cabeçalho da tabela (a.1) ao lado esquerdo e um registro (a.2) a direita, enquanto a tela (b) demonstra a tabela de uma maneira mais comum que possibilita a ação de *scroll* quando houver um *overflow* de registros.

Ambas visualizações possuem botões de navegação que são: para acesso a lista de bancos disponíveis na conexão (a.3 e b.1), trocar por outra tabela disponível do banco selecionado (a.4 e b.2). O modo *mobile* possui dois botões extras em relação ao modo *desktop* que são (a.5) para avançar um registro e (a.6) para voltar um registro.

Figura 4 Telas de visualização



Id	Nome	Rg
0	Teste	53.711.803-5

Fonte: O Autor, 2018.

Como apresentado, a tela de visualização claramente demonstra uma aplicação responsiva pois, quando o *display* apresenta um aspecto 3:4 (mais longo na vertical) o menu de navegação é realocado e o modo de visualização passa a ser de um único registro.

## 4 CONCLUSÕES





serviço *web* que tem características responsivas é viável através do Python com algumas bibliotecas e juntamente com o HTML, CSS, JS. Futuros trabalhos podem utilizar este como base de criação de uma aplicação que tenha como objetivo uma aplicação responsiva ou híbrida que tenha de fornecer telas tanto para computadores quanto para dispositivos móveis.

## 5 REFERÊNCIAS BIBLIOGRÁFICAS

- ANDREW, R. **Media queries**, [sl : sn], 2018. Disponível em: <[https://developer.mozilla.org/en-US/docs/Web/CSS/Media\\_Queries](https://developer.mozilla.org/en-US/docs/Web/CSS/Media_Queries)> Acesso em: 20 ago 18.
- GRAHAM, G. **CSS Grid: One Layout, Multiple Ways**, [sl : sn] 2017, <<https://css-tricks.com/css-grid-one-layout-multiple-ways/>> Acesso em: 20 ago 2018.
- JULIEN. **HTML: HyperText Markup Language**, [sl : sn], 2018, Disponível em: <<https://developer.mozilla.org/en-US/docs/Web/HTML>> Acesso em: 20 ago 18.
- MICHAEL, D. **HTML5**, [sl : sn], 2018, Disponível em: <<https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/HTML5>> Acesso em: 19 set 2018.
- MARCOTTE, E. **Responsive Web Design** [sl : sn], 2010, Disponível em: <<https://alistapart.com/article/responsive-web-design>> Acesso em: 07 set 2018.
- MARIADB FOUNDATION, **MariaDB**, [sl : sn], 2018, Disponível em: <<https://mariadb.org/about/>> Acesso em: 07 set 2018.
- MILLS, C. **Basic concepts of grid layout**, [sl : sn], 2018, Disponível em: <[https://developer.mozilla.org/en-US/docs/Web/CSS/CSS\\_Grid\\_Layout/Basic\\_Concepts\\_of\\_Grid\\_Layout](https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Grid_Layout/Basic_Concepts_of_Grid_Layout)> Acesso em: 12 ago 18.
- MILLS, C. **Java Script**, [sl : sn], 2018, Disponível em: <<https://developer.mozilla.org/en-US/docs/Glossary/JavaScript>> Acesso em: 20 ago 18.
- PYTHON SOFTWARE FOUNDATION. **The Python Standard Library**, [sl : sn], 2018, Disponível em: <<https://docs.python.org/3/library/>> Acesso em: 20 ago 2018.
- RIVERBANK COMPUTING LIMITED. **PyQt5**, [sl : sn], 2018, Disponível em: <<https://pypi.org/project/PyQt5/>> Acesso em: 20 ago 2018.
- RIVERBANK COMPUTING LIMITED. **What is PyQt?**, [sl : sn], 2018, Disponível em: <<https://riverbankcomputing.com/software/pyqt/intro/>> Acesso em: 16 set 18.

RONACHER, ARMIN. **Flask web development, one drop at a time**, [sl : sn], 2018,



*7ª Jornada Científica e Tecnológica da Fatec de Botucatu*  
29 de Outubro a 01 de Novembro de 2018, Botucatu - São Paulo, Brasil



<<http://flask.pocoo.org/>> Acesso em: 20 ago 2018.

WARD, C. **Jump Start Responsive Web Design**, [sl], 2 eds. 2017, 164 p. Disponível em: <<https://www.himason.com/docs/SitePointBooks/responsive2.pdf>> Acesso em: 17 ago 2018.