

## DESENVOLVIMENTO DE SOFTWARE BASEADO EM LINGUAGEM PYTHON PARA UM MODELO “USER-FRIENDLY” DE ESTIMATIVAS DE FLUXOS DE RADIAÇÃO ULTRAVIOLETA

Luiz Gustavo de Oliveira<sup>1</sup> (IC), Marcelo de Paula Correa (PQ)<sup>1</sup>

<sup>1</sup>Universidade Federal de Itajubá.

**Palavras-chave:** Exposição Solar. Interface de Usuário. Radiação Ultravioleta. Uvboost

### Introdução

O uso crescente de *softwares*, estende-se aos mais diversos setores da sociedade. Segundo a Brasscom, até o final de 2025, haverá cerca de 797 mil vagas para profissionais de tecnologia da informação (TI) (Barbosa, 2023). Um exemplo do uso dessas ferramentas é no campo da saúde, essenciais para auxiliar na prevenção de doenças. No caso específico a exposição à radiação solar, os softwares podem auxiliar como ferramentas para prevenção de doenças ligadas ao excesso de exposição ao sol, como eritema, queimaduras e câncer de pele, além de facilitar diagnósticos e monitoramento. Essas condições de saúde são particularmente agravadas em regiões tropicais devido à maior disponibilidade de radiação solar (Syrigos, 2005; Pires, 2017). O aumento do número de casos de câncer de pele e os avanços na pesquisa sobre os efeitos da radiação ultravioleta (RUV) nos seres humanos aumentam a atenção pública sobre o problema de exposição ao sol (Ferlay, 2021; Oliveira, 2008).

Doenças causadas pela RUV, podem ser prevenidas com a maior conscientização a respeito da exposição adequada ao sol e uso de protetores solares. Por essa razão, a informação do nível de intensidade RUV é importante para redução de riscos à saúde. O índice ultravioleta (IUV) é a ferramenta recomendada pela Organização Mundial da Saúde (OMS) para representar, a medida da intensidade da RUV (tabela 1) (WHO, World Health Organization; 2023).

Tabela 1 - Classificação do IUV

CATEGORIA	ÍNDICE ULTRAVIOLETA
BAIXO	< 2
MODERADO	3 a 5
ALTO	6 a 7
MUITO ALTO	8 a 10
EXTREMO	> 11

Fonte: CPTEC (2023)

Os modelos de transferência radiativa (MTRs) são a principal opção para cálculos da RUV, sendo precisos e de baixo custo computacional. No entanto, os MTRs são complexos para o usuário leigo e, além disso, podem ser limitados para cálculos climatológicos (i.e., grandes bases de dados) ou aplicações web que demandem respostas rápidas.

Nesse contexto, uma alternativa aos MTRs são estimadores baseados em técnicas de *machine learning* (ML). Um exemplo é o estimador UVBoost que apresenta bom desempenho em emular cálculos mais complexos, sendo desenvolvido a partir de métodos de ML combinados com um banco de dados de MTRs. Torna-se, portanto, uma boa opção para obtenção de informações de IUV e da irradiância ponderada para síntese de vitamina D com fácil acesso e baixo custo computacional (Corrêa, 2023). Mesmo assim, esses modelos teóricos não possuem interface simples e, por essa razão, estão distantes do uso para usuários leigos.

Assim, esse estudo tem como o objetivo o desenvolvimento de uma interface que permita o uso de um *software* estritamente científico por parte de usuários que desejam avaliar o IUV em situação cotidiana. Essa ferramenta permitirá à população em geral, incluindo médico e tomadores de decisões para políticas de saúde pública, estimar valores de IUV em escala local ou global em diferentes condições temporais, geográficas e atmosféricas.

### Metodologia

#### Linguagem Python

A linguagem de programação Python foi utilizada no desenvolvimento do *software* proposto, tendo em vista sua sintaxe fácil e suporte a pacotes de funções disponibilizadas por terceiros, denominadas bibliotecas. Para a meteorologia, o uso do Python é comum às mais diversas aplicações, obtendo destaque no tratamento de grande banco de dados. Um exemplo

do uso da linguagem é o trabalho de Buriol (2015), que desenvolveu um programa para visualização em tempo real de dados de radar meteorológico.

No desenvolvimento de *softwares*, bibliotecas como TKinter, PyQt, Kivy, possuem a função de desenvolvimento de interfaces gráficas a partir de elementos do próprio sistema operacional. O uso para o desenvolvimento de *softwares* na meteorologia com o Python foi aplicado por Araújo (2017) no desenvolvimento de um aplicativo para visualização de dados meteorológicos diários, o CLIMAP, desenvolvido com a biblioteca TKinter. No desenvolvimento *web*, Costa (2022) apresenta o sistema PAP Meteor, um sistema para análise e processamento de dados meteorológicos, o sistema foi desenvolvido a partir da linguagem Python com o *framework* Streamlit e as bibliotecas Pandas, Numpy, Scipy, Scikit-learn e Plotly.

No entanto, em comparação com outras linguagens de programação, o Python possui algumas desvantagens, sendo as principais o tempo de execução, devido à necessidade de um interpretador (Zehra, 2020) e o consumo de memória, não sendo ideal para tarefas que exigem muita memória, logo, Python não é uma boa linguagem para desenvolvimento móvel, limitando-se a plataformas de *desktop* e servidor (Zehra, 2020).

Tendo em vista a aplicabilidade da linguagem Python no desenvolvimento de *softwares*, o presente trabalho visa desenvolver uma interface amigável para o uso do estimador de RUV o UVBoost, e assim, possibilitar o uso amplo e comum de uma ferramenta científica.

### Desenvolvimento da interface

Primeiramente, definiu-se um desenho conceitual da interface e das funcionalidades a serem desenvolvidas. O programa foi pensado para operar em dois modos de operação, sendo o primeiro modo para cálculo voltado para o cálculo rápido referente à localidade do usuário. Neste modo, o usuário fornece informações do conteúdo total de ozônio (em unidades Dobson), profundidade óptica do aerossol (Adimensional), latitude, longitude, data e hora, e o programa determina o ângulo zenital solar e o IUV para a localidade. No segundo modo, o usuário fornece informações a partir de um arquivo csv (Figura 1), onde os dados de entrada ficam dispostos em colunas com as seguintes informações: ângulo zenital solar (graus), conteúdo total de ozônio e profundidade óptica do aerossol.

Figura 1 - Exemplo csv de entrada

	A	B	C
1	SZA	CTO	AOD
2	69415	321.3565	0.0184
3	69415	321362	0.0184
4	69415	321.3742	0.0184
5	69415	321.3786	0.0184

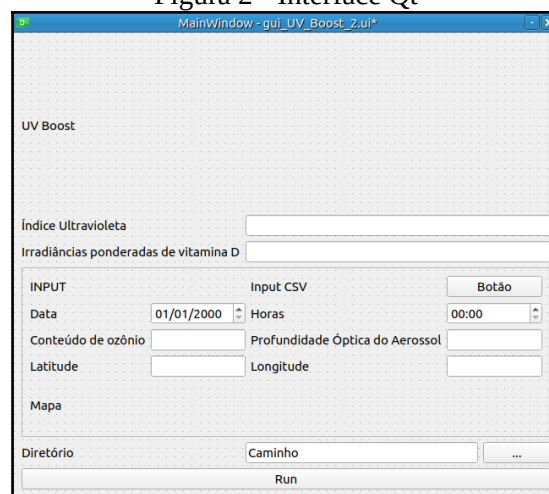
Fonte: Corrêa (2023)

Os resultados dos cálculos são apresentados conforme o modo de operação. Para o primeiro modo os resultados aparecem diretamente na interface para o usuário, já com a entrada csv o programa retorna o arquivo com duas colunas adicionais contendo o IUV e irradiância ponderada para síntese de vitamina D.

Inicialmente, o programa foi desenvolvido no ambiente Qt, ferramenta que permite a criação de e manipulação de elementos de interação (*widjets*) tais como janelas, botões, caixas de diálogo, dentre outros. Tais elementos podem ser integrados facilmente na programação, por meio de sinais e mecanismos de *slots*.

O arquivo gerado pelo Qt Designer é um arquivo com extensão “.ui” com base XML (Extensible Markup Language); linguagem de marcação utilizada para armazenamento e estrutura de dados em formato legível por máquina. Seu arquivo pode ser manipulado por meio da linguagem Python mediante sua conversão para formato “.py”, tal conversão e manipulação é possibilitado pela biblioteca PyQt, a figura 2 apresenta a versão desenvolvida no ambiente Qt.

Figura 2 - Interface Qt



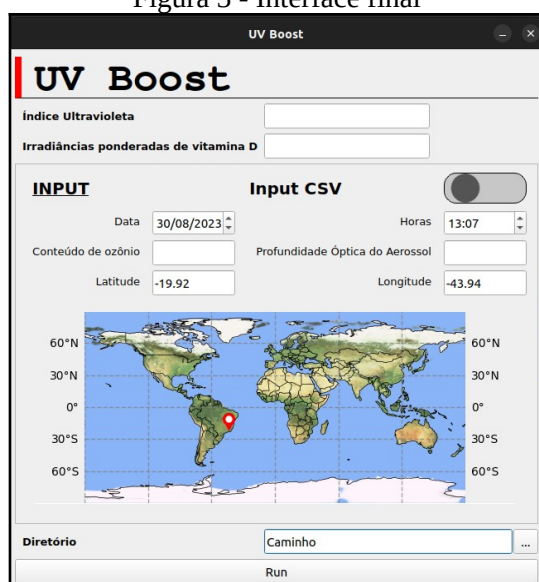
Fonte: Autoria própria

Para atender as necessidades propostas na idealização do programa, uma gama de módulos e bibliotecas do Python foram utilizadas, apresentada na tabela 2, tais bibliotecas amplificam as funções base da linguagem Python, possibilitando o desenvolvimento de aplicações mais complexas com um código mais simples.

Com a conversão do arquivo inicial desenvolvido no ambiente Qt para Python, pode-se realizar a comunicação entre os objetos e estilização da interface a ser desenvolvida mediante a linguagem Python. A linguagem em questão não possui suporte ao desenvolvimento de *softwares* nativamente, portanto, utilizaram-se os módulos do PyQt5 que possibilita a utilização de elementos gráficos do próprio sistema operacional. A interação do usuário com a interface é possibilitada por meio de sinais que comunicam a ação do usuário com as funções do programa, retornando para o usuário uma resposta à ação.

Atribuído as funções aos objetos, adicionou-se ao código folhas de estilo, permitindo a alteração do *layout* inicial, resultando em uma interface amigável ao usuário. A figura 3 apresenta a versão final da interface, já redimensionado e estilizado por meio de um arquivo externo de extensão “.qss” (Qt Style Sheet) de sintaxe muito próxima ao do CSS (Cascading Style Sheets).

Figura 3 - Interface final



Fonte: Autoria própria

A distribuição do código para sistemas operacionais Linux e Windows é possível devido ao Python ser uma linguagem multiplataforma, ou seja, por necessitar de

um interpretador a linguagem pode ser executada em qualquer sistema operacional sem depender do mesmo, apenas do interpretador. No entanto, mesmo se tratando de uma linguagem multiplataforma, o uso para uma aplicação pode não ser usual, pois há a necessidade de instalação de bibliotecas externas ou até mesmo o próprio pacote base do interpretador para que se possa executar o programa sem erros.

Por essa razão, foi gerado arquivos executáveis que contem todas as dependências em um único arquivo, facilitando o uso e aplicação. Para sistemas operacionais Windows, além do arquivo executável, foi gerado um arquivo instalador, permitindo empacotar arquivo externos e instalar no sistema, facilitando o uso por usuários leigos.

## Resultados e discussão

O programa final desenvolvido atingiu os objetivos propostos. Entretanto, vale ressaltar aspectos positivos e negativos encontrados durante o desenvolvimento do programa. Em um aspecto geral, a interface apresentou bons resultados em seu funcionamento, apresentando resposta rápida aos comandos e atendendo a proposta inicial de uma interface amigável para o usuário leigo. No entanto, no estágio final de desenvolvimento, o programa apresentou demorar em sua inicialização, relacionada ao uso da linguagem Python, inicialmente apresentada como uma linguagem “lenta”. Entretanto, é necessária uma análise mais detalhada para compreender os mecanismos de funcionamento da aplicação em sua inicialização.

Durante o processo de desenvolvimento, também foram identificados alguns problemas em relação ao suporte aos sistemas operacionais Linux e Windows. O programa, apesar de ser construído em uma linguagem multiplataforma, apresentou discrepâncias entre os sistemas devido a dependências externas. Desse modo, o programa, construído inicialmente em um ambiente Linux, teve de ser reformulado para sistema Windows. Contornado tal problema, o programa pode ser disponibilizado para o ambiente Windows. Porém, outro problema apresentado para o sistema é em relação ao instalador gerado para o UVBoost, reconhecido pelo sistema Windows como um *Malware*. Para sistema operacional Linux, o programa apresentou resultados melhores em relação ao tempo de execução e inicialização, no entanto, o pacote de instalação para o Linux apresentou falhas no processo, não funcionando corretamente quando instalado no sistema.

A distribuição final do programa para os dois sistemas, estão disponíveis no Drive em: <https://drive.google.com/drive/folders/1EogTYRrbtNuMKxyo8cpJ80pQ8XLroGBa?usp=sharing>, contendo o programa executável e suas dependências. Para o Windows, encontra-se no Drive o pacote de instalação do UVBoost, facilitando a implementação do programa. O código fonte do programa pode ser encontrado integralmente no GitHub em: <https://github.com/luizjezzuis/UVBoost.git>. Vale ressaltar, que atualizações para o programa serão realizadas, contribuindo para um melhor desempenho. Outro ponto a ser destacado é que o programa final estará disponível para download por período indeterminado.

### Conclusões

A interface desenvolvida atingiu os objetivos propostos. Apesar dos problemas encontrados, como a lentidão na inicialização devido à linguagem Python e as discrepâncias entre os sistemas operacionais Linux e Windows, o programa final cumpriu em grande parte a proposta inicial. A distribuição do UVBoost se encontra disponível no drive e no repositório do GitHub. Sugestões e críticas são bem-vindas e contribuirão com atualizações e demandas pelos usuários.

### Agradecimentos

Expresso meu agradecimento, primeiramente, ao meu orientador, Marcelo de Paula Corrêa, por ter me dado a oportunidade de me desenvolver em uma iniciação científica e aos meus amigos que me auxiliaram durante o processo. Agradeço também à Universidade Federal de Itajubá (Unifei) e ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) pelo recurso financeiro que possibilitou o desenvolvimento do projeto.

### Referências

ARAÚJO SALVADOR, Mozar. Climap–aplicativo para análise de dados climáticos-versão 3.0. *Revista Brasileira de Climatologia*, v. 20, 2017.

BARBOSA, L. **Até 2025 devem surgir quase 800 mil vagas de emprego para área de tecnologia.** Brasscom - Brasscom, , 21 mar. 2023. Disponível em: <<https://brasscom.org.br/ate-2025-devem-surgir-quase-800-mil-vagas-de-emprego-para-area-de-tecnologia/>>. Acesso

em: 16 set. 2023

BURIOL, Tiago Martinuzzi; BENETI, Cesar. Visualização científica interativa de dados de radar meteorológico. *Proceeding Series of the Brazilian Society of Computational and Applied Mathematics*, v. 3, n. 1, 2015.

CORRÊA, Marcelo Paula. UVBoost: An erythema weighted ultraviolet radiation estimator based on a machine learning gradient boosting algorithm. *Journal of Quantitative Spectroscopy and Radiative Transfer*, v. 298, p. 108490, 2023.

COSTA, Walingson da Silva et al. Sistema Web para pré-processamento e análise de dados meteorológicos. *Revista Brasileira de Climatologia*, v. 30, p. 591-610, 2022.

OLIVEIRA, Fernanda Spada Vaz Mano; FERREIRA, Erica Pontes Pereira. A EXPOSIÇÃO SOLAR PARA OBTENÇÃO DA VITAMINA DEO DESENVOLVIMENTO DO CÂNCER DE PELE: REVISÃO DE LITERATURA. *Revista Ibero-Americana de Humanidades, Ciências e Educação*, v. 9, n. 4, p. 294-308, 2023.

PIRES, Carla Andréa Avelar et al. Câncer de pele: caracterização do perfil e avaliação da proteção solar dos pacientes atendidos em serviço universitário. *Journal of Health & Biological Sciences*, v. 6, n. 1, p. 54-59, 2017.

SYRIGOS, Konstantinos N. et al. Skin cancer in the elderly. *in vivo*, v. 19, n. 3, p. 643-652, 2005.

WORLD HEALTH ORGANIZATION. The Known Health Effects of UV. Disponível em: <<http://www.who.int/uv/faq/uvhealthfac/en/index2.html>>. Data de Acesso: 21 de ago. de 2023.

ZEHRA, Farzeen et al. Comparative analysis of c++ and python in terms of memory and time. 2020.