

Cálculo e análise dos índices hematimétricos usando Python

CALCULATIONS AND ANALYSIS OF HEMATIMETRIC INDEX USING PYTHON

André Luís Fernandes dos Santos¹, Ellen de Camargo Campos², João Pedro Torres Alves³, Larissa Myrella Menaldo Dutra⁴, Laura Medeiros Tavares⁵, Rafaella Cristina Pontes Duca⁶, Tayná Meneses de Sousa⁷

¹ Doutor, Fundação Instituto de Educação de Barueri.
ORCID: 0000-0001-7053-4989

Email: andre.santos@docente.fieb.edu.br

² Técnico em Análises Clínicas, Fundação Instituto de Educação de Barueri.

ORCID: 0000-0002-4085-0961

Email: rm78847@estudante.fieb.edu.br

³ Técnico em Análises Clínicas, Fundação Instituto de Educação de Barueri.

ORCID: 0000-0003-1617-8965

Email: rm78289@estudante.fieb.edu.br

⁴ Técnico em Análises Clínicas, Fundação Instituto de Educação de Barueri.

ORCID: 0000-0001-9475-4361

Email: rm78284@estudante.fieb.edu.br

⁵ Técnico em Análises Clínicas, Fundação Instituto de Educação de Barueri.

ORCID: 0000-0003-1860-4649

Email: rm78242@estudante.fieb.edu.br

⁶ Técnico em Análises Clínicas, Fundação Instituto de Educação de Barueri.

ORCID: 0000-0002-9347-3645

Email: rm78541@estudante.fieb.edu.br

⁷ Técnico em Análises Clínicas, Fundação Instituto de Educação de Barueri.

ORCID: 0000-0002-2913-6097

Email: rm78815@estudante.fieb.edu.br

Correspondência: especificar o endereço para correspondência de pelo menos um autor (pode ser o endereço institucional).

Copyright: Esta obra está licenciada com uma Licença Creative Commons Atribuição-Não Comercial 4.0 Internacional.

Conflito de interesses: os autores declaram que não há conflito de interesses.

Como citar este artigo

Santos ALF dos, et al. Cálculo e análise dos índices hematimétricos usando Python. Revista de Saúde Digital e Tecnologias Educacionais. [online], volume 9, n. 1. Editor responsável: Luiz

Roberto de Oliveira. Fortaleza, abr de 2024.

Disponível em:

<http://periodicos.ufc.br/resdite/index>. Acesso em "dia/mês/ano".

Data de recebimento do artigo: 24/04/2023

Data de aprovação do artigo: 26/09/2023

Data de publicação: 18/04/2024

Resumo

Introdução: os índices hematimétricos (IH) avaliam os eritrócitos e são o VCM (volume corpuscular médio), CHCM (concentração de hemoglobina corpuscular média), HCM (hemoglobina corpuscular média) e o RDW (red cell distribution width). **Objetivo:** criar uma calculadora para os IH e fazer as análises dos resultados, utilizando a linguagem de programação *Python*. **Métodos:** a calculadora dos IH foi desenvolvida no Google Colab, com o *Python 3*. **Resultados e discussão:** no Google Colab é possível inserir os dados do paciente, realizar os cálculos e textos explicativos. Com o uso do *Google Colab* é possível disponibilizar a calculadora aos alunos, ou seja, é factível realizar testes e receber deles sugestões de melhoria. **Conclusão:** é possível utilizar o *Python* para calcular os IH e oportunizar aos educandos contatos com a programação.

Palavras-chave: Computação em Informática Médica. Técnicas de Laboratório Clínico. Anemia.

Abstract

Introduction: hematimetric indices (IH) evaluate erythrocytes and are MCV (mean corpuscular volume), MCHC (mean corpuscular hemoglobin concentration), HCM (mean corpuscular hemoglobin) and RDW (red cell distribution width). **Objective:** create a calculator for IH and analyze the results, using the Python programming language. **Methods:** the IH calculator was developed in Google Colab, with Python 3. **Results and discussion:** in Google Colab it is

possible to enter patient data, perform calculations and explanatory texts. Using Google Colab, it is possible to make the calculator available to students, that is, it is possible to carry out tests and receive suggestions for improvement from them. **Conclusion:** it is possible to use Python to calculate the IH and provide educated people with programming opportunities.

Keywords: *Medical Informatics Computing. Clinical Laboratory Techniques. Anemia.*

1. Introdução

O hemograma é um exame laboratorial qualitativo e quantitativo das séries branca (leucócitos), vermelha (hemácias) e plaquetária (plaquetas). Esses exames permitem o prognóstico de diversas doenças¹.

No passado a contagem das plaquetas, eritrócitos e leucócitos era manual, sendo lenta, trabalhosa e necessitava de diversos insumos, como pipetas diluidoras, câmaras de contagem, capilares, centrifugadores e alguns reagentes. Os parâmetros mais utilizados eram a dosagem de hemoglobina (Hb), o hematócrito (Ht) e a contagem dos leucócitos².

Com o avanço tecnológico introduziu-se os contadores hematológicos automatizados e semiautomatizados na realização do hemograma, que combinam diferentes metodologias para a avaliação dos índices hematológicos. Apesar do uso desses instrumentos, as técnicas manuais ainda continuam úteis como métodos de referência e na investigação de amostras com resultados anormais^{2,3}.

Os índices hematimétricos (IH) são usados para avaliar o formato, tamanho e volume dos eritrócitos, auxiliam na interpretação do hemograma e na classificação das anemias. São IH o VCM (volume corpuscular médio), o CHCM (concentração de hemoglobina corpuscular média), o HCM (hemoglobina corpuscular média) e o RDW (*red cell distribution width*), que é a amplitude da distribuição das hemácias^{4,5}.

Para o cálculo do VCM e do CHCM é necessário o valor do hematócrito, enquanto para o cálculo do HCM depende da contagem das hemácias. O hematócrito relaciona o volume ocupado pelas hemácias em um volume sanguíneo, sendo representado em porcentagem. Assim, as fórmulas para os cálculos dos IH são: VCM (hematócrito x 10 / hemácias); HCM (hemoglobina x 10 / hemácias); CHCM (hemoglobina x 100 / hematócrito)⁵.

Após o cálculo dos IH as anemias podem ser classificadas em microcítica, normocítica e macrocítica, considerando-se os valores referenciais do VCM. Em relação CHCM as anemias podem ser hipocrômica, normocrômica e hiperocrômica⁶.

Para o cálculo dos IH pode ser utilizada a linguagem de programação *Python*, criada em 1990, por Guido Van Rossum. A linguagem é aplicada no desenvolvimento de sistemas, pois é disponibilizada gratuitamente, além de ser compatível para qualquer sistema operacional⁷. Com ela podemos criar *sites*, organizar dados, executar *scripts*, testar comandos e entre outras funcionalidades. Além disso, utilizá-lo como calculadora, justificando seu uso nesse estudo⁸.

O cálculo dos IH contribui para o diagnóstico e prognóstico das anemias, uma condição bastante comum em pessoas e animais. Somado a isso a utilização da linguagem de programação *Python* para o cálculo desses índices mostra o avanço tecnológico na área de Análises Clínicas, além da integração da programação na rotina laboratorial. O conhecimento e aplicação dos IH são fundamentais para a formação de profissionais capacitados para atuar na área de Análises Clínicas.

Essa pesquisa teve como objetivo a criação de um *notebook* para calcular os IH e fazer as análises dos resultados obtidos, usando como ambiente de desenvolvimento integrado (IDE) o *Google Colab*, baseado na versão 3 do *Python*.

2. Métodos

O estudo foi realizado no laboratório de informática de uma instituição de Ensino Médio Técnico, responsável pelo curso técnico em Análises Clínicas. A calculadora foi desenvolvida no *Google Collaboratory* (*Google Colab* ou simplesmente *Colab*), utilizando a linguagem de programação *Python*, durante o ano de 2022. A linguagem *Python*, criada no final de 1989 e início de 1990 como projeto pessoal de Guido van Rossum, foi utilizada por ser acessível, simples, flexível, segura, portátil e disponível em diversas plataformas de código livre⁹.

O *Colab* é uma plataforma simples, gratuita, idealizada pelo *Google*. É de fácil acesso, já que é preciso conexão com internet estável e possuir uma conta do *Gmail* e bibliotecas da linguagem de *Python*¹⁰. Foi escolhido por ser um dispositivo virtual, com boa otimização, prático e que não precisa ser instalado no *desktop*: o seu uso é feito no navegador *web* de computadores e até mesmo de *smartphones*. Nele pode-se trabalhar em nuvem, tendo a possibilidade de interações e alterações pelas pessoas que estejam trabalhando simultaneamente. Qualquer alteração será salva na nuvem, garantindo que a execução não seja perdida¹¹.

Muitas são as utilidades pedagógicas e educacionais do *Colab*, mas uma delas é a facilidade de comunicação entre professor e aluno para o acompanhamento do aprendizado,

sendo útil para aulas online e presenciais. Nele é possível a criação dos chamados *notebooks*, estruturados em células que podem conter textos ou código executável (em *Python* ou *R*)¹².

Na execução da linguagem *Python* são essenciais inserir comandos, variáveis (objetos) e identificadores (condicionais). Para a construção do trabalho, foi necessário o uso dos comandos "print" e "input", os objetos "str" (*string*), "int" e "float", e as condicionais "if", "elif" e "else"⁹.

O "print" é o comando de saída, utilizado quando se deseja a exibição de alguma informação na tela. Este código deve ser montado da seguinte maneira: coloca-se o comando 'print', logo após abre-se e fecha-se parênteses e aspas (podem ser simples ou duplas), e em seguida digita-se o texto entre as aspas. Exemplo: print('Olá, mundo!')¹³.

O "input" é o comando de entrada, utilizado quando se deseja inserir algum dado no programa e vem acompanhado das variáveis (objetos) que são elementos da linguagem que ocupam um ou mais *bytes* no armazenamento do computador, isto é, são locais usados para armazenar dados que serão utilizados posteriormente⁹.

O objeto "str" é utilizado para a inserção de caracteres quaisquer, como letras, números, símbolos etc. Usa-se o "int" para inserir números inteiros, positivos ou negativos; e o "float" é empregado para inserir números decimais, sendo o separador decimal o ponto "." e não a vírgula ",",¹³. A estrutura desse código é parecida com a do "print", porém difere-se no início, onde é necessário inserir o objeto desejado, abrir e fechar parênteses e digitar 'input', e em seguida repetir o processo realizado no 'print'. Exemplos: str(input('Qual o seu nome?')); int(input('Qual a sua idade?')); float(input('Qual o seu peso?'))¹³.

As condicionais comparam dados para a realização de algum comando. Ou seja, caso algo ocorra, o programa deve realizar determinada atividade. Primeiro o programa passa pelo "if", depois, se houver, pelo "elif" e, por fim, pelo "else". Dessa forma, se a situação do "if" for verdadeira, ele agirá, senão o programa passará, caso houver, para o "elif", e, se necessário, para o "else"¹⁴.

A estrutura condicional funciona da seguinte maneira: digita-se o 'if', dando espaço e coloca-se o primeiro item a ser comparado, depois digita-se o operador (por exemplo, o de "menor que" (<)), e depois coloca-se o segundo item a ser comparado, finalizando com ":". Ao teclar "enter", um novo código poderá ser escrito na linha de baixo, para que seja realizada a ação desejada, utilizando o comando "print" ou "input". Caso seja necessário o uso do "else", basta repetir o procedimento do "if". Apenas o else não precisa da presença dos itens a serem comparados¹⁴.

Após a consulta das referências foi possível a construção do *notebook*, com textos explicativos sobre os IH e os códigos em *Python* (descrito nos resultados) para registro dos dados do paciente, dos valores de referência e para os cálculos e análises dos índices hematimétricos. Após passou-se para a fase de testes, onde foram incluídos valores de 10 hemogramas no *notebook* e comparado com os resultados constantes nos exames, verificando-se o perfeito funcionamento da calculadora.

3. Resultados

A primeira entrada de dados do *notebook* refere-se à identificação dos pacientes. Nesse momento os alunos deveriam digitar informações como nome, idade, sexo entre outras. Abaixo há a descrição do código fonte:

```
nome_paciente = str(input('Nome completo do paciente: '))
idade_paciente = int(input('Idade do paciente (anos): '))
sexo_paciente = str(input('Sexo do paciente [M/F]: ')).upper()
if sexo_paciente == 'F':
    sexo_pac = 'feminino'
else:
    sexo_pac = 'masculino'
peso_paciente = float(input('Peso do paciente (Kg): '))
print(f'O(a) paciente {nome_paciente} tem {idade_paciente} anos, é do sexo {sexo_pac} e pesa {peso_paciente} Kg.')
```

Códigos semelhantes a identificação do paciente foram utilizados para receber os valores de referência máximo e mínimos para as hemácias, a hemoglobina, o hematócrito e os índices hematimétricos, que podem variar conforme a idade, sexo e outros fatores:

```
ref_he_min = float(input('Valor de referência mínimo para as hemácias (milhões/mm3)? '))
ref_he_max = float(input('Valor de referência máximo para as hemácias (milhões/mm3)? '))
ref_hb_min = float(input('Valor de referência mínimo para a hemoglobina (g/dL)? '))
ref_hb_max = float(input('Valor de referência máximo para a hemoglobina (g/dL)? '))
ref_ht_min = float(input('Valor de referência mínimo para o Ht (%)? '))
ref_ht_max = float(input('Valor de referência máximo para o Ht (%)? '))
ref_vcm_min = float(input('Valor de referência mínimo para o VCM (fL)? '))
ref_vcm_max = float(input('Valor de referência máximo para o VCM (fL)? '))
```

```

ref_hcm_min = float(input('Valor de referência mínimo para o HCM (pg)? '))
ref_hcm_max = float(input('Valor de referência máximo para o HCM (pg)? '))
ref_chcm_min = float(input('Valor de referência mínimo para o CHCM (g/dL)? '))
ref_chcm_max = float(input('Valor de referência máximo para o CHCM (g/dL)? '))
ref_rdw_min = float(input('Valor de referência mínimo para o RDW (%)? '))
ref_rdw_max = float(input('Valor de referência máximo para o RDW (%)? '))

```

Após inserirmos os valores de referência temos de inserir os valores obtidos no exame do paciente, em relação a contagem de hemácias (he) e contagem de hemoglobina (hb) e hematócrito (ht):

```

he_paciente = float(input('Valor das hemácias do paciente (milhões/mm3): '))
hb_paciente = float(input('Valor da hemoglobina do paciente (g/dL): '))
ht_paciente = float(input('Valor do hematócrito do paciente (%): '))

```

Os códigos para os cálculos dos IH (VCM, HCM e CHCM) e inserção do valor do RDW do paciente são:

```

vcm_paciente = (ht_paciente / he_paciente) * 10
print(f'O VCM do paciente é {vcm_paciente:.2f} fL')
hcm_paciente = (hb_paciente / he_paciente) * 10
print(f'O HCM do paciente é {hcm_paciente:.2f} pg')
chcm_paciente = (hb_paciente / ht_paciente) * 100
print(f'O CHCM do paciente é {chcm_paciente:.2f} g/dL')
rdw_paciente = float(input('Valor do RDW do paciente (%)? '))

```

A partir dos valores de referência e dos dados fornecidos pelo hemograma do paciente foram calculados os IH e pode-se fazer a análise da sua condição hematológica.

A primeira análise refere-se ao número de hemácias, ou seja, com ela é possível verificar a condição de normalidade, policitemia ou anemia. Nesse momento as estruturas condicionais em Python são imprescindíveis:

```

if he_paciente > ref_he_max:
    print('Paciente apresenta policitemia.')
elif he_paciente < ref_he_min:
    print('Paciente apresenta anemia.')
else:
    print('Paciente sem alterações no número de hemácias.')

```

A partir da análise do VCM é possível definir se as hemácias estão com seu diâmetro aumentado (macrocitose) ou diminuído (microcitose). No caso do HCM saberemos apenas se

está aumentado ou diminuído, enquanto o CHCM indica se o paciente apresenta hipocromia ou hiperchromia:

```

if vcm_paciente > ref_vcm_max:
    print ('O paciente apresenta macrocitose.')
elif vcm_paciente < ref_vcm_min:
    print ('O paciente apresenta microcitose.')
else:
    print ('Paciente sem alterações no VCM.')

if hcm_paciente > ref_hcm_max:
    print ('O HCM do paciente está aumentado.')
elif hcm_paciente < ref_hcm_min:
    print ('O HCM do paciente está diminuído.')
else:
    print ('Paciente sem alterações no HCM.')

if chcm_paciente > ref_chcm_max:
    print ('Paciente apresenta hiperchromia.')
elif chcm_paciente < ref_chcm_min:
    print ('Paciente apresenta hipocromia.')
else:
    print ('Paciente sem alterações no CHCM.')

```

O valor do RDW indica anisocitose, se o paciente estiver com valores acima dos valores de referência:

```

if rdw_paciente > ref_rdw_max:
    print ('Paciente apresenta anisocitose.')
else:
    print ('Paciente sem alterações no RDW.')

```

Foram realizados testes com 10 hemogramas e em todos eles os resultados da calculadora em Python coincidiram com os dados dos exames. Em um paciente com contagem de eritrócitos igual a 4,79 milhões/mm³, hemoglobina de 15,2 g/dL e hematócrito de 47,9 % foram obtidos os seguintes IH (figura 1).

Figura 1 – Resultado dos índices hematimétricos (IH) utilizando a calculadora.

```
vcm_paciente = (ht_paciente / he_paciente) * 10
print(f'O VCM do paciente é {vcm_paciente:.2f} fL')

O VCM do paciente é 100.00 fL

hcm_paciente = (hb_paciente / he_paciente) * 10
print(f'O HCM do paciente é {hcm_paciente:.2f} pg')

O HCM do paciente é 31.73 pg

chcm_paciente = (hb_paciente / ht_paciente) * 100
print(f'O CHCM do paciente é {chcm_paciente:.2f} g/dL')

O CHCM do paciente é 31.73 g/dL
```

Fonte: autores (2022)

4. Discussão

A avaliação dos IH é executada no eritrograma (análise da série vermelha-eritrócitos), designando características importantes dos eritrócitos para a classificação de doenças hemolíticas, principalmente as anemias conforme os quatro índices hematimétricos¹⁵.

Em geral, a importância dos cálculos dos IH relaciona-se com o diagnóstico dos diferentes tipos de anemias, pois com eles pode-se identificar alterações no formato e no número de hemácias¹⁵. Além disso, as alterações dos índices podem ser indicadores de problemas nas amostras, nos equipamentos ou, até mesmo, de características do paciente, por isso é de extrema importância que a análise e interpretação dos laudos sejam feitas com cautela e por um profissional experiente¹⁶.

O *Python* é uma linguagem de programação acessível e relevante na área da saúde, que se caracteriza por usar tecnologias que produzem resultados e diagnósticos mais eficazes¹⁷. Na pandemia da COVID-19 o *Python* foi eficiente para analisar resultados dos testes do SARS-CoV-2 pela detecção de alterações de cores em microtubos¹⁸. É usado também para realizar previsões de óbito, como determinar os principais fatores de risco e características clínicas em idosos com COVID-19 e até mesmo relacionar a cura ou óbito do paciente pelas informações obtidas¹⁹. Existem outros trabalhos na área da saúde que utilizaram essa linguagem, por exemplo, quando foi usada para o controle da obesidade pela inteligência artificial e no levantamento e análise de dados da dengue em Recife²⁰.

Dessa forma, percebe-se o potencial de criação de projetos que podem se tornar programas de análises, sendo capazes de contribuir no tratamento de enfermidades, ao trazerem

resultados precisos, ou então, na coleta de dados sobre elas, conseguindo até fazer previsões epidemiológicas²⁰. Ademais, o *Python* aplicado na educação pode ser uma maneira de preparar os alunos para o futuro no mercado de trabalho e crescimento da tecnologia no mundo, por ser uma forma de aprendizagem em múltiplas áreas²¹.

5. Conclusão

Por ser versátil, o *Python* pode ser utilizado na saúde e na educação, como em matemática, física e química, por exemplo. Além disso possibilita aos educados o contato com linguagem de programação e conhecimentos na área de informática.

O uso do *Google Colab* torna o *notebook* acessível e disponível a outros alunos, pois qualquer indivíduo com o *link* do *notebook* pode acessá-lo e utilizá-lo. Ou seja, é factível a realização de testes com a comunidade acadêmica e receber dela sugestões de melhoria nas seções conceituais ou nos códigos utilizados.

Os códigos utilizados nesse estudo podem ser reproduzidos em outros ambientes de desenvolvimento integrado (IDE) e podem ser utilizados na construção de aplicativos com uma interface gráfica mais elaborada.

Assim, nota-se que é possível utilizar o *Python* para criar diversas aplicações. Para isso, os estudantes devem receber incentivos de suas instituições de ensino, a fim de fomentar sua criatividade, com o propósito de que sejam idealizados projetos de programas inovadores. Uma perspectiva futura é o desenvolvimento de um aplicativo que pode ser distribuído para uso em celulares, ou seja, a geração de um produto educacional.

Referências

1. Marty E, Marty RM. Hematologia Laboratorial. São Paulo: Érica, 2015. 120 p.
2. Bain B. Células Sanguíneas: Um Guia Prático. 5. ed. São Paulo: Artemed Editora S. A., 2016. 490 p.
3. Hoffmann LP, Polletti C, Roehrig KS et al. Avaliação dos índices hematimétricos emitidos pelos contadores hematológicos Pentra 120 Range e Sysmex XT-2000i. Revista Brasileira de Análises Clínicas. 2007; 39 (1): 25-28.
4. Souza DS, Santos ES, Alves EA, et al. Conceitos e métodos para formação de profissionais em laboratórios de saúde. 3. ed. Rio de Janeiro: Fiocruz; 2013. 306 p.
5. Santos, SLF, Costa VA, Tavares HST, et al. Utilização dos índices hematimétricos no diagnóstico diferencial de anemias microcíticas. Mostra Científica da Farmácia [Internet]. 2016; 3(1): 1-5. Disponível em: <http://reservas.fcrcs.edu.br/index.php/mostracientificafarmacia/article/view/1275>
6. Ribeiro-Alves MA, Gordan PA. Diagnóstico de anemia em pacientes portadores de doença renal crônica. Jornal Brasileiro de Nefrologia. 2007; 29(4): 4-6.
7. Borges LE. Python para desenvolvedores. São Paulo: Novatec Editora; 2014. 320 p.

8. Coelho FC. Computação científica com Python. Rio de Janeiro: Edição do autor; 2007. 263 p.
9. Banin, SL. Python 3: Conceitos e aplicações: Uma abordagem didática. São Paulo: Editora Érica; 2018. 264 p.
10. Silva, MD. Aplicação da Ferramenta Google Colaboratory para o Ensino da Linguagem Python. Anais da Escola Regional de Engenharia de Software (ERES) [internet]. 2020; 67-76. Disponível em: <https://sol.sbc.org.br/index.php/eres/article/view/13717>
11. Almeida F, Hayashi V. Experimento de Processamento de Músicas com Python Notebook e Aplicativo IoT. XXXIX Simpósio Brasileiro de Telecomunicações e Processamento de Sinais [Internet]. 2021; 1-5. Disponível em: <https://biblioteca.sbrt.org.br/articlefile/2884.pdf>
12. Baptista L. Usando Python e Google Colab para Ensinar Química Física Durante a Pandemia. 2021. 11 f. Documento de Trabalho- Universidade do Estado do Rio de Janeiro, Departamento de Química e Ambiental, Rio de Janeiro, 2021. Disponível em: <https://chemrxiv.org/engage/apigateway/chemrxiv/assets/orp/resource/item/60c7548e0f50dba5b9397d43/original/using-python-and-google-colab-to-teach-physical-chemistry-during-pandemic.pdf>
13. Menezes MNC. Introdução à Programação com Python. 3 ed. São Paulo: Novatec; 2019. 328 p.
14. Perkovic, L. Introdução à Computação Usando Python - um Foco no Desenvolvimento de Aplicações. São Paulo: LTC; 2016. 516 p.
15. Grotto HZW. O hemograma: importância para a interpretação da biópsia. Revista Brasileira de Hematologia e Hemoterapia. 2009; 31(3): 178-182.
16. Brum MF. Eritrograma: Novas Perspectivas de Análise. 2013. 12 f. Trabalho de Conclusão de Curso (Pós-Graduação em Hematologia Laboratorial) – Curso de Pós-Graduação, Universidade Regional do Noroeste do Estado do Rio Grande do Sul, Rio Grande do Sul, 2013. Disponível em: https://bibliodigital.unijui.edu.br:8443/xmlui/bitstream/handle/123456789/1657/Artigo_Proto%20MABEL_BRUM_pdf.pdf?sequence=1&isAllowed=y
17. Filho EAG, Silva MA, Campelo LC, et al. Uso de inteligência artificial para auxiliar o combate à obesidade: estudo de caso em Python. Ciência da Computação: Tecnologias Emergentes em Computação. 2021; 2(1): 274 – 282.
18. Feltrin P, Kulman C. Desenvolvimento de métodos portáteis para o diagnóstico de COVID-19. Mostra Integrada da Região NORTE – IFRS 2021. Disponível em: <https://eventos.ifrs.edu.br/index.php/mostraintegradaregiaoNorte/MoepexIbiruba/paper/view/11175>
19. Lima TPF, Sena GR, Neves, CS, et al. Previsão de óbito e importância de características clínicas em idosos com COVID-19 utilizando o Algoritmo Random Forest. Revista Brasileira de Saúde Materno Infantil. 2021; 21: 453-460.
20. Castro AF, Oliveira AG, Vieira GFF, et al. Utilização da linguagem Python na análise de dados de dengue em Recife, capital do estado de Pernambuco, Nordeste do Brasil e definição de linha de tendência usando Machine Learning. Disponível em: https://www.researchgate.net/publication/361989988_Utilizacao_da_linguagem_Python_na_analise_de_dados_de_dengue_em_Recife_capital_do_estado_de_Pernambuco_Nordeste_do_Brasil_e_definicao_de_linha_de_tendencia_usando_Machine_Learning_Use_of_Python_language
21. Fria BL, Alves JN, Costa CM. Uma experiência de inserção de Linguagem de Programação no Ensino Médio Integrado à Educação Profissional Técnica a partir da flexibilização curricular. Revista Labor. 2021; 1(25): 242-260.