**ARTHUR ADABO DE CAMARGO**

**PEDRO ALVARES EGGERS**

**COMPUTER AIDED MOLECULAR DESIGN (CAMD) APLICADO À INDÚSTRIA DE COSMÉTICOS**

Trabalho de Conclusão de Curso em PROJETO DE ENGENHARIA apresentada à Escola Politécnica da Universidade de São Paulo para Graduação no Curso de Engenharia Química

(Undergraduate Engineering Project and Design)

**Departamento de Engenharia Química**

**São Paulo, 2019**

ARTHUR ADABO DE CAMARGO

PEDRO ALVARES EGGERS

**COMPUTER AIDED MOLECULAR DESIGN (CAMD) APLICADO À INDÚSTRIA DE COSMÉTICOS**

Trabalho de Conclusão de Curso em PROJETO DE ENGENHARIA apresentada à Escola Politécnica da Universidade de São Paulo como requisito parcial para Graduação no Curso de Engenharia Química

(Undergraduate Engineering Project and Design)

Orientadores:

Prof. Dr. Moisés Teles

São Paulo

2019

|  |
| --- |
| Autorizo a entrega dessa versão da Monografia sob responsabilidade única dos autores e de seus orientadores.  São Paulo, xx de xx de 2019  Assinatura do orientador: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (Prof. Dr. Moisés Teles)  Assinatura do autor: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (Arthur Adabo de Camargo)  Assinatura do autor: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (Pedro Alvares Eggers) |

FICHA CATALOGRÁFICA

|  |
| --- |
| Cruces Cerro, Jorge  Microfibrilas de celulose nas fibras marrons para melhoria das propriedades de papéis kraftliner / J. Cruces-Cerro. São Paulo, 2016.  230 p.  Trabalho de Conclusão de Curso (PROJETO de ENGENHARIA de Graduação) - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia Química.  1. FIbras celulósicas 2. Kraftliner 3. Papel Cartão 4.Microfibrilas de celulose 5. Máquina de papel I.Universidade de São Paulo.Escola Politécnica.Departamento de Engenharia Química II.t. |

AGRADECIMENTOS

Ao ... por ..

A ....por...

RESUMO

A industria de cosméticos, assim como muitas outras, depende de odores artificiais para que seus produtos sejam mais atrativos ao cliente. Para obtenção desses odores, é necessário conhecimento de propriedades fisico-químicas de moléculas – como a pressão de vapor – para que elas se encaixem nos requerimentos dos produtos. Assim, pode-se utilizar o método de CAMD, que pretende obter uma molécula dadas certas propriedades, como, no caso, o odor.

Com o CAMD, o uso de um banco de dados que permite fácil acesso à propriedades das partículas é essencial para que especialistas ou até programas automatizados possam formular novos compostos com odores agradáveis e que sigam as necessidades do produto, como baixa toxicidade, certo ponto de ebulição ou solubilidade específica.

Então, neste projeto, foram utilizadas as linguagens *SQLite* e *Python* para montar uma interface que seja capaz de acessar um banco de dados que contenha informações sobre compostos, tentando identificar características e associá-las a odores, assim como calcular valores julgados relevantes e que possam ser úteis para o desenvolvimento de determinados bens produzidos sobretudo pela indústria de cosméticos.

COMPUTER AIDED MOLECULAR DESIGN APPLIED TO THE COSMETIC INDUSTRY

ABSTRACT

The cosmetics industry, as well as many others, relies on artificial odors to make their products more attractive to their customers. To obtain these odors, it is necessary to know the physico-chemical properties of molecules - such as vapor pressure - to fit the requirements of the products. Thus, the CAMD method, which seeks to obtain a molecule given certain properties, such as odor, may be used.

With CAMD, the use of a database that allows easy access to molecule properties is essential for specialists or even automated programs to formulate new compounds with pleasant odors that follow product needs, such as low toxicity, a certain boiling point or a specific solubility.

So, in this project, the SQLite and Python languages ​​were used to build a database that contains compound information and an interface that can access it, trying to identify characteristics and associate them with odors, as well as calculate relevant values which may be useful for the development of certain goods produced mainly by the cosmetics industry.

LISTA DE FIGURAS

[Figura 1 - Exemplo da estrutura do banco de dados 9](#_Toc26552010)

[Figura 2 - Exemplo da execução do programa com GUI 10](#_Toc26552011)

NOMENCLATURA E SÍMBOLOS

|  |  |  |
| --- | --- | --- |
| **Síglas** | | |
| CAMD | *Computer Aided Molecular Design* |  |
| GUI | *Graphic User Interface* |  |
| SMILES | *Simplified Molecular Input Line Entry Specification* |  |
| IUPAC | *International Union of Pure and Applied Chemistry* |  |
| *csv* | *Comma Separated Values* (formato de arquivo) |  |

SUMÁRIO

[1 INTRODUÇÃO 2](#_Toc26614964)

[1.1 Definição de Computed Aided Molecular Design 2](#_Toc26614965)

[1.2 Importância do CAMD para a indústria de cosméticos 2](#_Toc26614966)

[1.3 Objetivos do trabalho 3](#_Toc26614967)

[1.4 Estrutura da apresentação do Projeto 3](#_Toc26614968)

[2 DESCRIÇÃO DOS OBJETOS DO ESTUDO 5](#_Toc26614969)

[2.1 Banco de dados de compostos 5](#_Toc26614970)

[2.2 Interface de Usuário 5](#_Toc26614971)

[3 REVISÃO DAS TECNOLOGIAS 6](#_Toc26614972)

[3.1 Banco de dados de compostos 6](#_Toc26614973)

[4 DESCRIÇÃO DE PROCESSO 8](#_Toc26614974)

[4.1 Tabulação de dados: SMILES e odor 8](#_Toc26614975)

[4.2 Importação de dados do *PubChem* via API 8](#_Toc26614976)

[4.3 Criação do banco de dados 9](#_Toc26614977)

[4.4 Criação da interface gráfica do usuário (GUI) 10](#_Toc26614978)

[5 CONCLUSÃO E FUTUROS TRABALHOS 13](#_Toc26614979)

[6 REFERÊNCIAS BIBLIOGRÁFICAS 14](#_Toc26614980)

[7 APÊNDICE 15](#_Toc26614981)

[*7.1* Programa em Python para a GUI *(gui.py)* 15](#_Toc26614982)

[*7.2* Programa desenvolvido em Python para fazer as queries no banco de dados *(query.py)* 16](#_Toc26614983)

[*7.3* Programa para obter os dados de compostos químicos do *PubChem (import\_pubchem.py)* 17](#_Toc26614984)

**SUMÁRIO EXECUTIVO**

(MÁXIMO DUAS PAGINAS)

# INTRODUÇÃO

## Definição de Computed Aided Molecular Design

Na indústria química, de maneira geral, os compostos utilizados em determinado processo são conhecidos, sendo exploradas as suas propriedades em dadas condições. Por exemplo, em um processo de separação de uma mistura composta por tolueno e benzeno, é importante saber quais propriedades – como densidade, calor específico etc. – estas substâncias possuem nas temperaturas e pressões utilizadas.

O *Computer Aided Molecular Design* é o processo inverso: sabendo as propriedades desejadas em certas condições, busca-se um composto químico que as apresente. Este processo envolve tanto buscas simples em bancos de dados como análise de modelos moleculares para prever características relevantes da molécula, como seus grupos funcionais.

## Importância do CAMD para a indústria de cosméticos

Na formulação de um perfume, muitas vezes sabe-se qual será o seu aroma (floral, amadeirado, doce) antes de ser definida qual será sua composição. No passo seguinte, especialistas em aroma, cujo conhecimento do ramo é essencialmente empírico, selecionam substâncias que atendam à necessidade do produtor.

Nesse sentido, o uso de CAMD poderia, com base em informações do aroma, da pressão de vapor e de outras propriedades relevantes para a fragrância, fornecer combinações possíveis de compostos de maneira objetiva, eficiente e rápida.

É importante ressaltar que o CAMD não tem como objetivo substituir o trabalho dos especialistas em aroma, que adicionam um componente pessoal e subjetivo aos perfumes, mas sim em auxiliá-los a obter mais combinações de substâncias que gerariam as características desejadas no produto em um tempo menor do que atualmente.

O perfumista David Apel, há 39 anos no ramo e tendo desenvolvido perfumes para marcas como Avon, Nina Ricci e Gap, afirma que aplicações de CAMD utilizando inteligência artificial – como a *Philyra*, da IBM – "dão a chance de encontrar fórmulas de perfumes que eu nunca teria pensado sozinho" (OSTERATH, 2019).

## Objetivos do trabalho

O presente trabalho tem como objetivo criar um programa básico para auxiliar na escolha de compostos com base em suas propriedades. Inicialmente, será elaborado um banco de dados das substâncias relevantes à indústria de cosméticos, escrito em SQLite, que será acessível via uma interface de usuário escrita em *Python*.

O resultado do trabalho em questão poderá ser ampliado futuramente para de fato usar modelos preditivos e *Machine Learning* e escolher moléculas com propriedades escolhidas pelo usuário.

## Estrutura da apresentação do Projeto

A Tese foi desenvolvida de acordo com a estrutura indicada a seguir:

**Capítulo 2**: descreve os objetos do projeto, como o programa, sua aplicação e suas estruturas.

**Capítulo 3**: descreve as ferramentas e *softwares* utilizados.

**Capítulo 4**: apresenta o passo-a-passo da realização do projeto, bem como o algoritmo do programa criado.

**Capítulo 5**: apresenta as conclusões obtidas com o projeto e planejamento para próximos passos de melhorias

**Apêndice**: reúne o código dos três arquivos criados em *Python*.

# DESCRIÇÃO DOS OBJETOS DO ESTUDO

Nesta seção, serão descritos os objetos de estudo que compõem o programa realizado neste trabalho.

## Banco de dados de compostos

De forma a armazenar informações relevantes sobre compostos com aromas, bem como permitir a pesquisa de maneira fácil e rápida, foi elaborado um banco de dados para tais substâncias.

## Interface de Usuário

# REVISÃO DAS TECNOLOGIAS

Nesta seção, serão descritas as tecnologias utilizadas para elaborar o projeto, focando em suas características técnicas.

## Banco de dados de compostos

A linguagem escolhida para o banco de dados em questão foi o *SQLite* devido ao fato desta ser simples, altamente funcional, rápida, gratuita e no domínio público, compatível com várias plataformas e com capacidade de armazenamento suficientemente grande para a aplicação neste trabalho. Como os bancos de dados em *SQLite* suportam até 140 TB de armazenamento, não haveria qualquer dificuldade técnica em armazenar um número grande de compostos em passos futuros do trabalho.

No momento da conclusão deste trabalho, haviam 234 compostos no banco de dados, ocupando apenas 36 kB de espaço. Nota-se, portanto, que o projeto é completamente escalável, não havendo uma limitação real à ampliação do banco de dados. Teoricamente, supondo um crescimento linear do espaço em disco ocupado pelo banco de dados com o número de linhas, poderiam ser armazenados 910 bilhões de compostos, tornando-o essencialmente ilimitado.

Além disso, o *SQLite* é a linguagem de banco de dados mais utilizada no mundo, estando presente em todos os aparelhos celulares com *Android* e *iOS*, todos os computadores com *Windows 10*, todos os *Macs*, nos navegadores *Chrome, Firefox e Safari* e em outras aplicações.

## Interface gráfica, interação com o banco de dados e tratamento de dados

Para realizar as partes do programa relativas à *GUI*; interações com o banco de dados – criação, atualização de tabelas, bem como consultas às mesmas – e tratamento de dados, a linguagem escolhida foi *Python*.

Os motivos dessa escolha estão no fato do *Python* ser uma linguagem poderosa e versátil, contendo diversos módulos nativos ou não para as mais variadas aplicações, bem como pela familiaridade dos integrantes com a mesma. Além disso, o número de aplicações baseadas em *Python* e de pessoas com conhecimento na linguagem têm crescido fortemente – 456% desde 2018 –, tornando mais provável que algum usuário do programa conheça a forma com que ele foi escrito.

Os módulos do *Python* utilizados na elaboração do programa estão apresentados com suas respectivas funções na tabela 1 a seguir:

Tabela - Módulos de *Python* utilizados

|  |  |
| --- | --- |
| Módulo | Função |
| **Interface Gráfica** | |
| *Tkinter* | Elementos visuais, como a janela do programa |
| *PIL (Python Imaging Library)* | Leitura de imagens de compostos no formato .png |
| **Interação com banco de dados** | |
| *sqlite3* | Interpretação e interação com *SQLite3* |
| *csv* | Leitura de tabelas em *.csv* |
| **Tratamento de dados** | |
| *pubchempy* | *API* para obter dados de moléculas do *PubChem* |
| *csv* | Leitura de tabelas em *.csv* |
| *xlsxwriter* | Escrita em arquivos .*xlsx* do Excel |

# DESCRIÇÃO DO PROCESSO

Neste capítulo, serão explicados os programas utilizados para pré-processamento de dados e para consultas ao banco de dados de substâncias criado. Foram criados três programas, cada um para realizar um objetivo relacionado ao trabalho. As etapas realizadas foram as seguintes:

## Tabulação de dados: SMILES e odor

Com base em dados conhecidos sobre o odor de compostos (KORICHI, 2008) e suas respectivas notações em SMILES, inicialmente organizou-se uma tabela com estes dados no *Excel*, sendo ela convertida ao formato *.csv*. Esta parte foi feita manualmente devido à dificuldade de coletar de maneira automática os dados do artigo citado.

O formato *.csv* foi escolhido devido à facilidade de se trabalhar com o mesmo em *Python*, utilizando o módulo homônimo nesta linguagem para fazê-lo. Seria possível armazenar os dados em formatos *.xls* ou *.xlsx*, mas utilizando outro módulo no *Python* para trata-los, como o *Pandas*, por exemplo.

## Importação de dados do *PubChem* via API

Com a tabela correlacionando a notação SMILES das moléculas com seus odores, o próximo passo foi obter mais dados – como nome IUPAC e fórmula química – sobre elas em bancos de dados confiáveis. No caso, foi escolhido o *PubChem* devido à sua extensão e confiabilidade (ele é mantido pela *U.S. National Library of Medicine).*

Levando em conta a quantidade de compostos cujas informações deveriam ser buscadas, o preenchimento manual destes valores foi tido como inviável pelo grupo. Para automatizar esse processo, foi escrito um pequeno programa (*import\_pubchem.py*) em *Python* utilizando o módulo *pubchempy*, disponibilizado pelo próprio *PubChem* (SWAIN, 2017).

O passo-a-passo do programa é o seguinte:

1. Leitura e conversão ao objeto *list* do arquivo contendo a informação sobre a notação SMILES extraído de Korichi et. al.
2. Para cada composto no arquivo:
   1. Pesquisa do seu nome e fórmula molecular no *PubChem*
   2. Escrita desses dados em um novo arquivo *.xlsx* utilizando o módulo *xlsxwriter* do *Python*

Essa etapa consiste no pré-processamento dos dados que, embora tenha levado cerca de 10 minutos para ser concluída, é extremamente mais rápida do que o processo feito manualmente. Além disso, vale ressaltar que a API também permite o *download* de imagens das estruturas do compostos, que serão mostradas ao usuário.

Todavia, a obtenção de mais dados sobre as moléculas, como a densidade, pKa e solubilidade, levou mais tempo para ser feita devido ao fato de estas informações não estarem acessíveis via API, sendo necessário obtê-las manualmente. Ademais, nem todos os compostos no *PubChem* continham dados sobre estas propriedades, tornando o banco de dados incompleto em alguns pontos. Todo esse processo levou algumas horas para ser feito.

## Criação do banco de dados

Em seguida, foi criado o banco de dados utilizando a linguagem *SQLite3*, feito também automaticamente através de um programa (*query.py*) em *Python*, utilizando o módulo *sqlite3* desta linguagem. No banco de dados, foram inseridas treze colunas, relativas respectivamente ao SMILES, odor, nome IUPAC, à fórmula química, massa molar, aos pontos de ebulição e fusão, à densidade, solubilidade, pressão de vapor, densidade de vapor, ao ponto de flash e pKa da molécula em questão.

Todavia, embora alguns destes valores tenham sido encontrados facilmente, outros não foram, deixando o banco de dados incompleto em alguns pontos. Estes valores ainda estão sendo pesquisados em outras fontes.

Após a criação do banco de dados, feita através da função *create\_db()* no arquivo *query.py*, ele foi populado com o arquivo criado no ponto 4.2 deste relatório utilizando a função *update\_db()* do arquivo. Foi necessário, devido a particularidades da linguagem, converter o arquivo com os dados em uma lista de tuplas do *Python*.

Na figura 1 a seguir, está exposto como o banco de dados está estruturado.



Figura 1 - Exemplo da estrutura do banco de dados

## Criação da interface gráfica do usuário (GUI)

De forma a tornar a busca ao banco de dados mais amigável a eventuais usuários, foi criada uma *GUI (Graphic User Interface)* também em *Python*, utilizando o módulo *tkinter* (SHIPMAN, 2013), cujo programa tem nome *gui.py*.

O funcionamento deste programa depende da criação de uma classe – chamada de *Window* por representar a janela aberta ao usuário quando o arquivo é executado – com diversos elementos:

1. Três barras de pesquisa: a primeira pelo nome do composto, a segunda pela sua fórmula e a última pelo seu odor;
2. Três botões para realizar as pesquisar citadas acima;
3. Três *Frames* (como se fossem “quadros” da janela, facilitando sua organização): um para comportar as pesquisas acima e dois para expor os seus resultados.

Quando o usuário clica algum botão, é executada a função *search\_compounds*, que acessa o arquivo *query.py*, realiza a pesquisa com base no *input* do usuário, e expõe os seus resultados nos *Frames* inferiores.

Na figura 2 abaixo, está exposto um exemplo de como a *GUI* está na data de envio deste relatório. É importante ressaltar que nem todas as colunas do banco de dados estão sendo expostas devido ao fato de se tratar de uma versão preliminar da *GUI*.

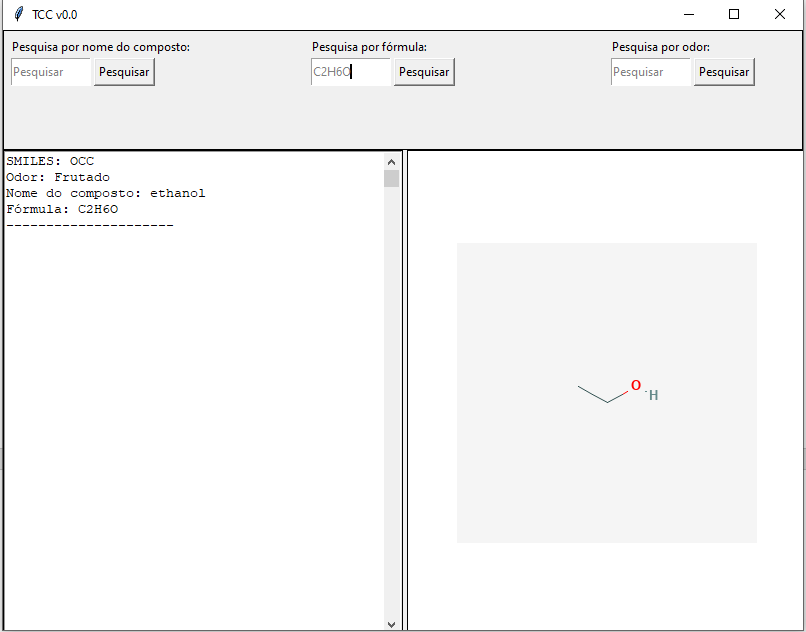


Figura - Exemplo da execução do programa com GUI

# CONCLUSÃO E FUTUROS TRABALHOS

O programa criado, embora atualmente não forneça *insights* completamente úteis à indústria cosmética, pode servir como uma base consistente para tal. Além disso, notou-se que a criação e modificação do banco de dados, bem como as pesquisas a ele, são feitas de maneira rápida e objetiva. A interface de usuário, embora simples, mostrou-se suficiente para atender as demandas iniciais por buscas a um banco de dados.

Futuramente, será necessário ampliar o banco de dados para conter mais moléculas e mais colunas de informações. Neste banco de dados, idealmente, poderiam ser inclusas tabelas com coeficientes de modelos de propriedades, de forma que estas poderiam ser calculadas conforme o *input* do usuário. Ademais, há a necessidade de se melhorar a *GUI* do programa, de forma a exibir de maneira mais organizada as informações coletadas e/ou calculadas a partir do banco de dados.

Considerando a eficiência do projeto, o ponto em que poderiam ser aplicadas mais melhorias seria na obtenção das propriedades das moléculas, o qual levou um tempo considerável para ser concluído, além de ser feito manualmente, tornado erros de preenchimento mais suscetíveis. Como estes dados não estão acessíveis via API, seria necessário desenvolver tal funcionalidade, fato que necessitaria permissão do órgão responsável pelo *PubChem*, ou busca-los em outra plataforma.

Além disso, fortificando a importância da inclusão de modelos termodinâmicos, estes poderiam ser utilizados para estimar as propriedades ausentes no *PubChem* para alguns compostos.

# REFERÊNCIAS BIBLIOGRÁFICAS

OSTERATH, Brigitte. Artificial intelligence creates perfumes without being able to smell them. **Deutsche Welle**, 31 mai. 2019. Disponível em: https://www.dw.com/en/artificial-intelligence-creates-perfumes-without-being-able-to-smell-them/a-48989202. Acesso em: 8 out. 2019.

KORICHI, Mourad. et. al. Computer-aided aroma design. II. Quantitative structure–odour relationship. **Chemical Engineering and Processing: Process Intensification**, volume 47, ed. 11, 1912-1925, outubro de 2008.

SWAIN, Matt. **PubChemPy Documentation**: Release 1.0.4. [*S. l.*], 11 abr. 2017. Disponível em: https://pubchempy.readthedocs.io/en/latest/. Acesso em: 2 dez. 2019

SHIPMAN, John. **Tkinter 8.5 reference**: A GUI for Python. [*S. l.*], 31 dez. 2013. Disponível em: https://anzeljg.github.io/rin2/book2/2405/docs/tkinter/index.html. Acesso em: 29 nov. 2019.

SQLITE CONSORTIUM. **What Is SQLite?**. [*S. l.*]. Disponível em: https://www.sqlite.org/index.html. Acesso em: 7 dez. 2019.

Bhawani. The Incredible Growth of Python Language. **Zibtek**. [*S. l.*], 22 out. 2019. Disponível em: https://www.zibtek.com/blog/the-incredible-growth-of-python/. Acesso em: 7 dez. 2019.

# APÊNDICE

## Programa em Python para a GUI *(gui.py)*

## Programa desenvolvido em Python para fazer as queries no banco de dados *(query.py)*

## Programa para obter os dados de compostos químicos do *PubChem (import\_pubchem.py)*