

Camila Machado de Araújo

**Relatório de Estágio Supervisionado  
Laboratório Nacional de Luz Síncrotron**

Campina Grande, Paraíba

Dezembro de 2020

Camila Machado de Araújo

## **Relatório de Estágio Supervisionado** **Laboratório Nacional de Luz Síncrotron**

Relatório de Estágio Integrado submetido à Coordenação de Graduação em Engenharia Elétrica da Universidade Federal de Campina Grande, Campus Campina Grande, como parte dos requisitos necessários para a obtenção do grau de Bacharela em Ciências no Domínio da Engenharia Elétrica.

Universidade Federal de Campina Grande – UFCG

Centro de Engenharia Elétrica e Informática – CEEI

Departamento de Engenharia Elétrica – DEE

Orientador: Saulo Oliveira Dornellas Luiz, D. Sc.

Campina Grande, Paraíba

Dezembro de 2020

Camila Machado de Araújo

## **Relatório de Estágio Supervisionado Laboratório Nacional de Luz Síncrotron**

Relatório de Estágio Integrado submetido à Coordenação de Graduação em Engenharia Elétrica da Universidade Federal de Campina Grande, Campus Campina Grande, como parte dos requisitos necessários para a obtenção do grau de Bacharela em Ciências no Domínio da Engenharia Elétrica.

Trabalho aprovado. Campina Grande, Paraíba, 14 de dezembro de 2020:

---

**Saulo Oliveira Dornellas Luiz, D. Sc.**  
Orientador

---

**Danilo Freire de Souza Santos, D. Sc.**  
Convidado

Campina Grande, Paraíba  
Dezembro de 2020

*Este trabalho é dedicado às crianças adultas que,  
quando pequenas, sonharam em se tornar cientistas.*

# Agradecimentos

Agradeço aos meus pais, Adriane Machado de Araújo e Robson Sérgio Batista de Araújo, e ao meu irmão, Rodrigo Assis Machado de Araújo, aos quais eu devo tudo o que sou como ser humano.

Agradeço ao meu melhor amigo e namorado, André Igor Nóbrega da Silva, por tanto amor e por todo o apoio incondicional. Sem você eu não teria conseguido.

Agradeço ao meu orientador, Saulo Oliveira Dornellas Luiz, que desde o 2018 orientou-me no âmbito acadêmico, profissional e pessoal.

Agradeço ao meu supervisor, Narcizo Marques de Souza Neto, pela oportunidade de desenvolver este trabalho. Agradeço também aos colegas da linha EMA, Danusa, Gustavo, Joel, Judá, Leonardo, Lucas Andreotti, Marcos, Marina, Vanessa, Bárbara, Jairo, Ulisses e Ricardo, pelo acolhimento e por ajudarem em prontidão sempre que precisei. Em especial, a Lucas Henrique Francisco, pelas valorosas contribuições e direcionamentos.

Agradeço ao Instituto Serrapilheira pelo financiamento ao projeto "Um olhar com raios X em supercondutividade".

*‘— Senhoras e senhores — gritam eles —, estamos flutuando no espaço!  
Mas nenhuma das pessoas lá de baixo se interessa pela gritaria dos filósofos. .  
(O Mundo de Sofia, Jostein Gaarder)*

# Resumo

A descoberta de um material supercondutor em condições ambientes seria revolucionária para todas as aplicações elétricas. O cálculo teórico da temperatura crítica ( $T_c$ ) é essencial para a previsão de supercondutores, permitindo a análise de estruturas conhecidas e previstas teoricamente buscando alto  $T_c$ . O pacote Quantum ESPRESSO (QE) é um conjunto integrado de código aberto usado para esse propósito. Embora esta ferramenta esteja disponível, o processo de cálculo de  $T_c$  é bastante complexo. Por causa disso, o tempo de preparação da simulação de um composto é longo, podendo custar um dia inteiro de trabalho. O software ACESC foi desenvolvido em Python para automatizar cálculos de estrutura eletrônica com o pacote QE. O uso da nova ferramenta reduz o tempo de preparação das simulações para poucos segundos. Ainda, a arquitetura modular permite expandir as funcionalidades do programa e automatizar qualquer simulação suportada pelo pacote QE.

**Palavras-chave:** Temperatura Crítica de Supercondutividade, Quantum ESPRESSO, Python, Automação via Software.

# Abstract

The discovery of a superconducting material in ambient conditions would be revolutionary for all electrical applications. The theoretical calculation of the critical temperature ( $T_c$ ) is essential to the prediction of superconducting, allowing the analysis of both known and theoretically new structures seeking high  $T_c$ . The Quantum ESPRESSO (QE) package is an integrated open source suite used for this purpose. Although this tool is available, the  $T_c$  calculation process is quite complex. Because of this, the preparation time for the simulation of a compound is long, and can cost an entire day of work. The ACESC software was developed in Python to automate electronic structure calculations with the QE package. the use of the new tool reduces the preparation time of the simulations to a few seconds. Furthermore, the modular architecture allows to expand the program's functionalities and automate any simulation supported by the QE package.

**Keywords:** Critical Superconductivity Temperature, Quantum ESPRESSO, Python, Software Automation.



# Lista de ilustrações

Figura 1 – Processo antigo de cálculo de $T_c$ com o pacote QE do ponto de vista do usuário. . . . .	17
Figura 2 – Diretório de resultado do cálculo de $T_c$ com pacote QE. . . . .	18
Figura 3 – Novo processo de cálculo com software ACESC do ponto de vista do usuário. . . . .	21
Figura 4 – Diretório de resultado do cálculo de $T_c$ com software ACESC. . . . .	22
Figura 5 – Visão lógica da arquitetura do programa. . . . .	24
Figura 6 – Visão de implementação das principais interfaces, <i>IQuantumEspresso</i> e <i>ICalculation</i> . . . . .	24
Figura 7 – Visão de implementação - módulos do pacote <i>core</i> . . . . .	25
Figura 8 – Visão de implementação - módulos do pacote <i>util</i> . . . . .	26
Figura 9 – Visão de processo do software ACESC e bibliotecas utilizadas na implementação de cada etapa. . . . .	27

# Lista de abreviaturas e siglas

CNPEM	Centro Nacional de Pesquisa em Energia e Materiais
DFT	<i>Density Functional Theory</i>
EMA	<i>Extreme condition x-ray Methods of Analysis</i>
GPa	Giga Pascal
K	Kelvin
LNLS	Laboratório Nacional de Luz Síncrotron
QE	Quantum ESPRESSO
$T_c$	Temperatura crítica de supercondutividade

# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>11</b>
1.1	Objetivos	11
1.2	Estrutura do relatório	12
<b>2</b>	<b>AMBIENTE DE TRABALHO</b>	<b>13</b>
2.1	Laboratório Nacional de Luz Síncrotron	13
2.2	Projeto de pesquisa	13
<b>3</b>	<b>METODOLOGIA</b>	<b>15</b>
3.1	Cálculo da temperatura crítica de supercondutividade	15
3.2	Identificação de problemas	16
<b>4</b>	<b>ATIVIDADES DESENVOLVIDAS</b>	<b>19</b>
4.1	Estudo de ferramentas e técnicas para desenvolvimento de software	19
4.2	Software ACESC	20
4.2.1	Visão do usuário	20
4.2.2	Visão de implementação	22
<b>5</b>	<b>DISCUSSÃO</b>	<b>28</b>
<b>6</b>	<b>CONCLUSÃO</b>	<b>29</b>
	<b>REFERÊNCIAS</b>	<b>30</b>

# 1 Introdução

Neste documento são descritas as principais atividades desenvolvidas no estágio integrado pela discente Camila Machado de Araújo, do curso de Engenharia Elétrica da Universidade Federal de Campina grande – UFCG. O estágio foi realizado no Laboratório Nacional de Luz Síncrotron – LNLS, no projeto de pesquisa “Um olhar com raios x em supercondutividade”, sob supervisão do líder de pesquisa Narcizo Marques de Souza Neto e orientação do professor Saulo Oliveira Dornellas Luiz.

São descritas as atividade realizadas entre 17 de fevereiro de 2020 e 30 de novembro de 2020, totalizando 1384h. Cumpre-se, assim, a carga horária obrigatória de 660h necessárias para aprovação na disciplina de estágio integrado.

A realização do estágio, cumprimento de sua carga horária e apresentação do relatório de atividades são obrigatórios para a obtenção do diploma de bacharel em Engenharia Elétrica. Sua finalidade é proporcionar uma experiência profissional na qual os alunos coloquem em prática os conhecimentos aprendidos, além de complementar sua formação com habilidades essenciais para o mercado de trabalho.

Dentre as atividades desenvolvidas pela estudante, destacam-se:

1. Identificação de problemas;
2. levantamento de requisitos de projeto;
3. interação com usuário;
4. desenvolvimento de software em Python.

## 1.1 Objetivos

Inserido no projeto: "Um olhar com raios X em supercondutividade", o objetivo do estágio é realizar simulações teóricas de propriedades de materiais supercondutores. Foram feitos cálculos de primeiros princípios para previsão de estruturas cristalinas estáveis e estimativa de temperaturas críticas de supercondutividade.

Pode-se destacar os objetivos específicos:

- Aprender sobre as ferramentas existentes e fundamentação teórica do projeto;
- otimizar ou desenvolver novas ferramentas para os cálculos de propriedades de materiais;

- desenvolver ou utilizar códigos existentes para determinar estruturas e temperaturas de materiais supercondutores.

## 1.2 Estrutura do relatório

Este documento está dividido em 6 capítulos. O primeiro dedicado à introdução e apresentação dos objetivos do trabalho. No segundo capítulo é apresentado o ambiente de trabalho. Segue-se a metodologia de trabalho. No quarto capítulo são apresentadas as atividades desenvolvidas, no quinto capítulo, uma discussão sobre o trabalho apresentado e, por fim, o sexto capítulo apresenta a conclusão deste trabalho.

## 2 Ambiente de Trabalho

Neste capítulo será apresentado o Laboratório Nacional de Luz Síncrotron e o projeto de pesquisa no qual a estagiária atuou, desenvolvido nas instalações do Sirius.

### 2.1 Laboratório Nacional de Luz Síncrotron

O LNLS é um dos quatro grandes laboratórios que compõem o Centro Nacional de Pesquisa em Energia e Materiais (CNPEM). O LNLS é responsável pela única fonte de luz síncrotron da América Latina, o Sirius, um dos mais avançados aceleradores de partículas do mundo em sua categoria, considerado de quarta geração.

Inaugurado em 2018, o Sirius é a maior e mais complexa infraestrutura de pesquisa já construída no país. Sua estrutura possui diversas linhas de luz para realização de experimentos variados com radiação síncrotron produzida no acelerador. Assim, é possível executar diversas técnicas de análise microscópica da matéria e de suas propriedades fundamentais.

As estações experimentais do LNLS estão disponíveis para pesquisadores brasileiros, estrangeiros e usuários industriais realizarem suas investigações. Com instalações abertas, o laboratório recebe anualmente cerca de 1200 pesquisadores envolvidos em mais de 400 estudos que originam aproximadamente 200 artigos publicados em periódicos científicos.

A estagiária atuou em um dos projetos de pesquisa desenvolvidos no LNLS, cujos experimentos serão realizados na linha EMA (*Extreme condition x-ray Methods of Analysis*). Esta linha de luz destaca-se no estudo de materiais sob condições extremas, como em altíssimas pressões.

### 2.2 Projeto de pesquisa

O Projeto de pesquisa: "Um olhar com Raios X em Supercondutividade" é coordenado pelo líder de pesquisa Narcizo Marques de Souza Neto, supervisor do estágio. O projeto tem por objetivo desenvolver novas técnicas de espectroscopia com raios X de alto brilho para estudar o fenômeno de supercondutividade. Embora o fenômeno tenha sido descoberto em 1911, ainda não há um modelo geral que explique seu comportamento.

Há ainda a possibilidade de descobrir um novo material supercondutor à temperatura ambiente. Uma descoberta como essa poderia revolucionar a tecnologia moderna. Isso se deve à característica de resistência nula de um supercondutor. Um material sem

perdas por resistência elétrica ocasionaria uma grande redução no custo das aplicações que envolvem eletricidade.

Além de ser desenvolvido nas instalações do LNLS, o projeto é apoiado pelo Instituto Serrapilheira, uma instituição privada, sem fins lucrativos, de fomento à ciência no Brasil. O Serrapilheira foi criado em 2017 e possui duas frentes de atuação: a ciência e a divulgação científica. Dentre suas atividades, fornece apoio financeiro a projetos de pesquisa fundamental. Desde sua criação, o instituto já apoiou mais de 123 projetos de pesquisa e 36 projetos de divulgação científica.

## 3 Metodologia

Durante este estágio foram empregados o método de desenvolvimento ágil e a abordagem de *Design Thinking*. Segundo a metodologia de desenvolvimento ágil, projetos de software podem ter um objetivo claro e ao mesmo tempo permitir a adaptação e mudança de requisitos durante seu desenvolvimento. Isso ocorre a medida que os desenvolvedores e clientes interagem e exploram as possibilidades de solução de um problema durante o processo de desenvolvimento (HIGHSMITH; COCKBURN, 2002).

Processos ágeis levam em consideração imprevistos e possibilidades de mudança constantemente. O processo de desenvolvimento adapta-se a medida que surgem problemas e novas situações. Para isso, é preciso um ambiente de colaboração e interação contínuo e flexibilidade de desenvolvimento. Aplicou-se uma adaptação do modelo Kanban (CORONA; PANI, 2013) para um único desenvolvedor. A estagiária realizou reuniões quinzenais com o supervisor de estágio e pesquisador relacionado ao projeto nas quais apresentava as atualizações no desenvolvimento e as atividades seguintes. Quando necessário, marcava-se reuniões mais frequentes.

O *Design Thinking* é uma abordagem de resolução de problemas que coloca o usuário no centro do processo de desenvolvimento até chegar a uma solução. Isso ocorre a partir das etapas de: empatia, definição do problema, ideação, prototipação e teste. O processo de empatia envolve conversar com usuários e se inserir no contexto do problema para compreender suas necessidades. Colocando-se sob a perspectiva do usuário é possível definir melhor o problema a ser resolvido. As ideias de solução são colocadas em prática, testando-as com os usuários (STANFORD, 2010).

O processo de empatia com o usuário envolve compreender qual será o contexto de utilização do sistema, bem como as motivações e necessidades dos usuários. Com essa finalidade, nesse capítulo introduz-se o contexto das pesquisas em supercondutividade, bem como os problemas identificados no processo antigo de cálculo de  $T_c$ .

### 3.1 Cálculo da temperatura crítica de supercondutividade

Os materiais supercondutores conduzem eletricidade sem qualquer resistência (perda de energia) abaixo de uma determinada temperatura crítica ( $T_c$ ). Um dos principais objetivos dessa área de estudo é a descoberta de um material supercondutor a temperatura ambiente, que possa ser utilizado em várias aplicações com eletricidade.

Em 2015, foi descoberto o sulfeto de hidrogênio ( $H_3S$ ) sob supercondutividade de alta pressão ( $> 150$  GPa), com o valor de  $T_c$  de 203K (DROZDOV et al., 2015).



Desde essa descoberta, a ideia de que pressões extremas podem estabilizar fases de supercondutores com  $T_c$  mais alto tem sido foco de diversos estudos (BI et al., 2018). A busca por novos materiais de alto  $T_c$  se beneficia fortemente da previsão teórica de novas composições e estruturas promissoras por meio de simulações. Vários super-hidretos tiveram sua estabilidade em pressões extremas e valores de  $T_c$  assim previstos. O composto conhecido com o maior  $T_c$  observado (260K a 180-200 GPa) é o  $\text{LaH}_{10}$  (GEBALLE et al., 2018), (LIU et al., 2017), estudado do ponto de vista teórico antes de ser relatado experimentalmente (SOMAYAZULU et al., 2019).

Uma das principais ferramentas computacionais disponíveis para o cálculo de  $T_c$  é o pacote Quantum ESPRESSO <sup>1</sup> (QE), o qual é um conjunto integrado de código aberto para cálculos de estrutura eletrônica e modelagem de materiais em nanoescala. Sua implementação se baseia na Teoria do Funcional da Densidade (DFT - *Density Functional Theory*), em ondas planas e pseudopotenciais (GIANNOZZI et al., 2017). Essa é a principal ferramenta utilizada pelo grupo de pesquisa em supercondutividade da linha EMA para calcular  $T_c$ . Embora esta poderosa ferramenta esteja disponível, foram identificados vários problemas que dificultam a utilização da mesma.

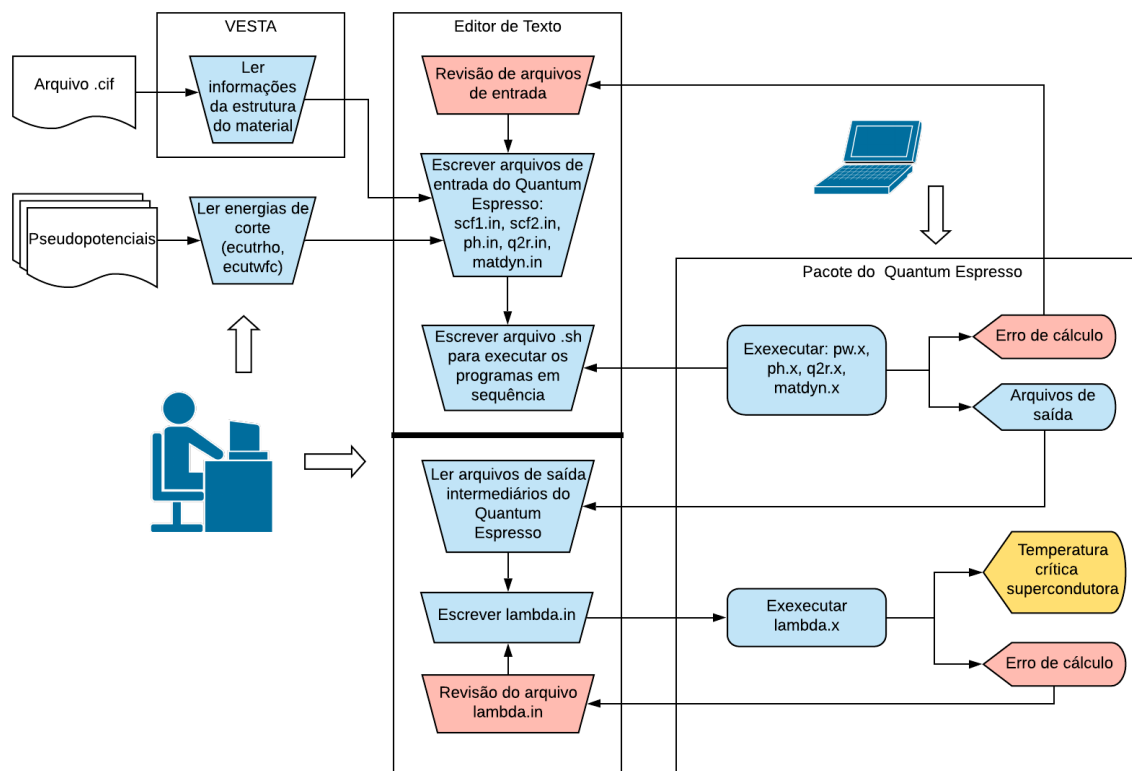
O cálculo de  $T_c$  com o pacote QE requer seis etapas diferentes e cinco programas: *pw.x*, *ph.x*, *q2r.x*, *matdyn.x*, *lambda.x*, de dois pacotes do QE: PWscf e PHonon. O pacote PWscf permite calcular propriedades da estrutura eletrônica aplicando a DFT. O pacote PHonon permite o cálculo dos fônons de uma estrutura cristalina, ou seja, os modos, intensidade e frequência de vibrações do retículo cristalino. O programa *pw.x* realiza a solução da DFT, o programa *ph.x* calcula os fônons, os programas *q2r.x*, *matdyn.x* e *lambda.x* reúnem as informações da estrutura eletrônica e dos fônons para calcular  $T_c$ . A Figura 1 representa esquematicamente as principais atividades do processo antigo de cálculo de  $T_c$ , do ponto de vista do usuário.

## 3.2 Identificação de problemas

A primeira atividade do estágio foi a familiarização com todas as etapas do processo de cálculo de  $T_c$  e compreensão dos principais conceitos físicos envolvidos. A imersão no processo e contato direto com os usuários permitiu elencar vários problemas e dificuldades operacionais, listados a seguir:

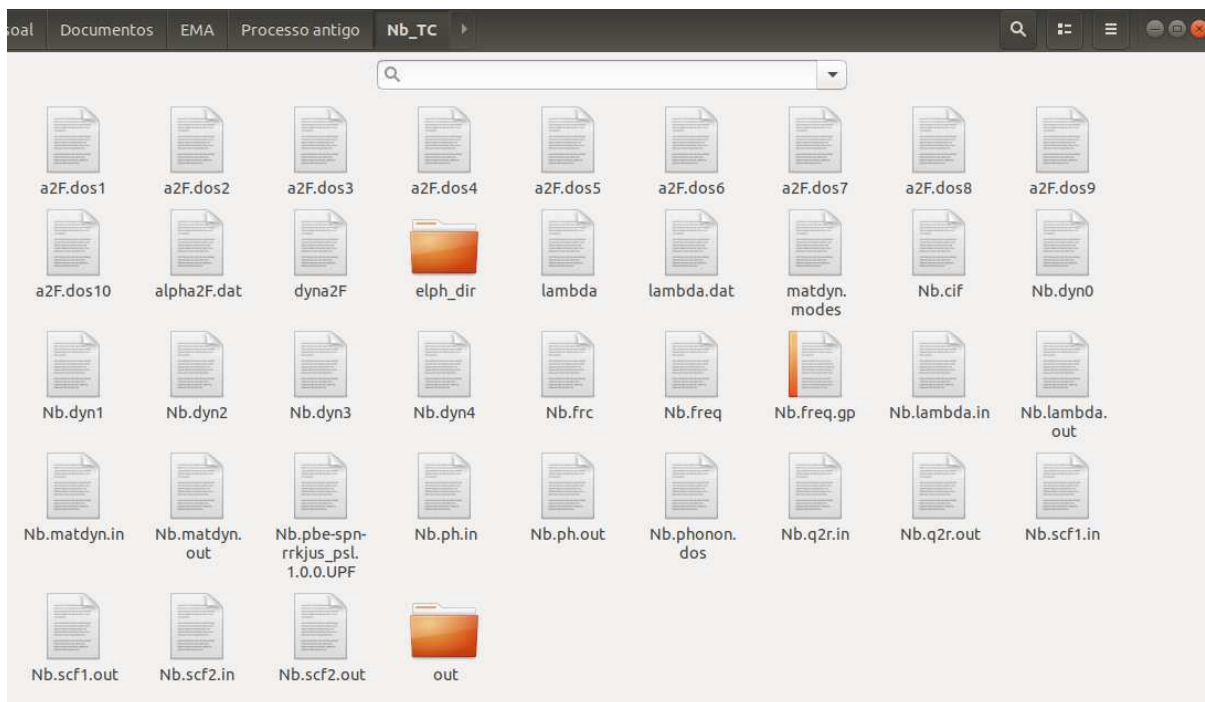
- **Preocupação com indentação** - cada arquivo de entrada é escrito no formato *.txt* e requer uma indentação específica para a definição de parâmetros. Erros simples como um espaço em branco no final do arquivo ou uma contra-barras ausente podem comprometer o funcionamento do programa e são difíceis de identificar, visto que não há uma ferramenta de verificação desses arquivos.

<sup>1</sup> <<https://www.quantum-espresso.org/>>

Figura 1 – Processo antigo de cálculo de  $T_c$  com o pacote QE do ponto de vista do usuário.

Fonte: Autoria Própria, 2020.

- **Compatibilização de variáveis** - os diversos programas compartilham dados por arquivos intermediários, ou seja, os arquivos de saída de um dado programa servirão como entradas para o programa subsequente. Para que isso ocorra, há variáveis nos arquivos de entrada que indicam os nomes dos arquivos intermediários. Caso algum dos nomes esteja errado, os dados não serão encontrados pelo programa subsequente, comprometendo completamente o cálculo.
- **Grande número de variáveis de entrada** - o usuário precisa determinar quase 70 parâmetros, considerando os 6 arquivos de entrada. Algumas delas são difíceis de obter, como a posição exata dos átomos do material, sua massa atômica e o arquivo de pseudopotencial usado para cada elemento. Esses são fatores objetivos que podem ser automatizados.
- **Diversos processos individuais** - cada programa possui uma sintaxe Própria, 2020 para o comando de execução pelo terminal. Paliativamente, utilizava-se um arquivo `.sh` para encadear as 5 etapas do cálculo e salvar os dados de saída sem precisar esperar os resultados de cada etapa para iniciar a próxima. A escrita do arquivo de entrada do sexto programa dependia dos resultados dos anteriores, portanto, não podia ser automatizada com as demais.

Figura 2 – Diretório de resultado do cálculo de  $T_c$  com pacote QE.

Fonte: Autoria Própria, 2020.

- **Desorganização dos arquivos de saída** - cada programa do pacote QE possui um arquivo de saída principal com informações importantes em formato amigavelmente legível. Juntamente a ele, há inúmeros arquivos intermediários que não são de interesse imediato do usuário e nem próprios para leitura humana. No processo antigo, ao final do cálculo, todos os arquivos de entrada, de saída e intermediários encontravam-se misturados no mesmo diretório, Figura 2.

O processo de cálculo era manual em muitas etapas objetivas que não dependiam de julgamento do pesquisador. Além disso, estava sujeito a inúmeras falhas humanas, difíceis de detectar na ausência de qualquer ferramenta de verificação dos arquivos. Devido aos problemas citados, o tempo de preparação de uma simulação era grande, podendo custar um dia inteiro de trabalho para preparar os arquivos de entrada para o cálculo de  $T_c$  de apenas uma estrutura cristalina.

As simulações de  $T_c$  são ferramentas poderosas na pesquisa em supercondutividade, como apresentado na seção 3.1. Indicações teóricas de propriedades supercondutoras são importantes para orientar os escopos dos experimentos, aumentando suas chances de sucesso. Isso envolve a realização de diversas simulações com compostos diferentes, além de diversas simulações de um mesmo composto, variando a pressão. Dessa forma, automatizar o processo de cálculo da temperatura de  $T_c$  representa ganhos significativos de tempo e produtividade para o andamento da pesquisa.

## 4 Atividades Desenvolvidas

Neste capítulo será descrita a arquitetura do software ACESC, desenvolvido com o objetivo de sanar os problemas elencados na seção 3.2. Também relata-se as atividades de estudo paralelas que permitiram a obtenção dos conhecimentos necessários para desenvolver a solução apresentada.

### 4.1 Estudo de ferramentas e técnicas para desenvolvimento de software

O processo de desenvolvimento de software pode ser comparado a escrita de um texto. Embora seja um processo criativo, no qual cada programador desenvolverá uma solução única e original, ele precisa ser bem estruturado para ser compreendido por outras pessoas e, no caso do desenvolvimento de software, expandido e reutilizado. Há diversas estruturas e conceitos que orientam um bom desenvolvimento, desde a nomenclatura de variáveis até as escolhas arquiteturais. Assim, o desenvolvimento do sistema proposto foi acompanhado do estudo contínuo de novas técnicas para desenvolvimento de software e importantes etapas de refatoração.

Em um primeiro momento, o estudo da linguagem Python foi essencial para o início do desenvolvimento do programa. Essa linguagem foi escolhida por ser amplamente utilizada no laboratório e na área de estudo. O pacote QE é implementado em Python, assim como a biblioteca ASE<sup>1</sup> (*Atomistic Simulation Environment*) (LARSEN et al., 2017), que permite a manipulação de estruturas cristalinas e possui funções compatíveis com o pacote QE. Os principais materiais utilizados foram a documentação oficial da linguagem e o PEP8 (ROSSUM; WARSAW; COGHLAN, 2001), ambos disponíveis em <https://www.python.org/>, e o livro "Learning Python" (LUTZ, 2013).

Em um segundo momento, foi fundamental o estudo de padrões de design para estruturar a arquitetura. A determinação de uma boa arquitetura para resolver um problema de software com orientação a objeto é uma tarefa extremamente difícil, inclusive para programadores experientes. O projeto precisa ser específico o suficiente para resolver determinado problema e, ao mesmo tempo, manter um grau de generalidade para ser aplicado a problemas futuros. Flexibilidade e reusabilidade são as principais características de uma boa solução. Nesse contexto, o estudo de padrões de design bem estabelecidos torna mais fácil a construção de bons projetos. O livro "Elements of reusable object-oriented software" (GAMMA et al., 1995) foi fonte importante de conhecimento nesse campo.

<sup>1</sup> <<https://wiki.fysik.dtu.dk/ase/about.html>>

Outra faceta fundamental no desenvolvimento de qualquer projeto de software é o versionamento. Escolheu-se a ferramenta de código aberto git (TORVALDS; HAMANO et al., 2005) para o controle de versão na máquina de desenvolvimento local, e o GitHub para hospedar remotamente o projeto e distribuí-lo no futuro. No contexto do desenvolvimento de código aberto, uma boa estruturação de repositório e documentação são primordiais. Para isso, o estudo dos repositórios de grandes projetos como o pacote QE e a biblioteca ASE foi muito importante.

## 4.2 Software ACESC

Inicialmente, desenvolveu-se uma solução de software para automatizar o cálculo  $T_c$ , com o objetivo de eliminar operações manuais e otimizar o tempo dos pesquisadores. No entanto, o pacote QE permite a obtenção de diversas outras propriedades da matéria. No decorrer do projeto, identificou-se a possibilidade de alterar a arquitetura existente para permitir a inclusão de qualquer outra rotina de cálculo do QE.

O software ACESC (*Automated Crystla-based Electronic Structure Calcualtor*) automatiza a obtenção de propriedades da matéria a partir do pacote QE. O software encapsula as especificidades dos diversos programas do QE e suas inter-relações em um nível de abstração inferior ao do usuário, por meio da implementação de rotinas de cálculo. Desse modo, o pesquisador precisa lidar apenas com os parâmetros de interesse do ponto de vista físico para a tarefa desejada, de forma simples e rápida.

### 4.2.1 Visão do usuário

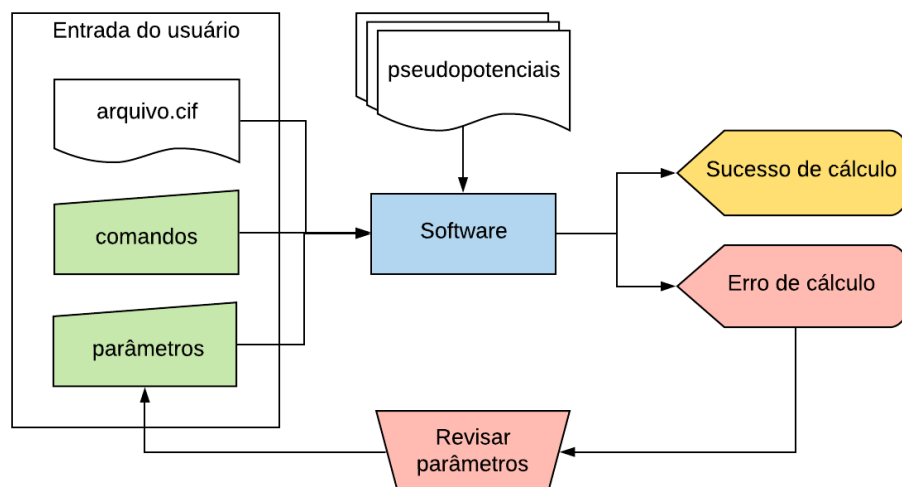
O novo processo, do ponto de vista do usuário é apresentado na Figura 3. Destaca-se a grande redução no número de tarefas do usuário em relação ao processo antigo (Figura 1) e a eliminação de vários estágios propícios a erros. O tempo de preparo de uma simulação, que podia ser de mais de um dia de trabalho, foi reduzido a poucos segundos (no cenário mais simples). O processo de cálculo foi drasticamente simplificado para o usuário. Ao mesmo tempo, manteve-se toda a flexibilidade de ajustes da simulação.

O software ACESC é operado pelo terminal. O comando para execução de uma simulação possui o seguinte formato padrão:

```
$ acesc.py <arquivo.cif> <opmode> <clmode> <parametros.txt>
```

As entradas *arquivo.cif* e *parâmtros.txt* são diretórios de arquivos e *opmode* e *clmode* são comandos opcionais. O arquivo *cif* (*Crystallographic Input Information*) é a descrição da estrutura cristalina e única entrada obrigatória. Este arquivo possui formato padrão amplamente utilizado pela comunidade científica e contém todas as informações do

Figura 3 – Novo processo de cálculo com software ACESC do ponto de vista do usuário.



Fonte: Autoria Própria, 2020.

material para os cálculos de estrutura eletrônica. Também é de fácil acesso, visto que há bancos de dados de domínio público que disponibilizam arquivos *cif* de diversas materias. Um exemplo é o COD (Crystallography Open Database)<sup>2</sup>.

O arquivo de parâmetros reúne em um único arquivo de texto os parâmetros do cálculo de um tipo de simulação. Mantém-se a nível do usuário apenas os parâmetros de significado físico e ajuste de convergência. O arquivo é opcional pois, na sua ausência, cada tipo de simulação possui parâmetros padrões que garantem um cálculo consistente. O formato padrão do arquivo é gerado pelo próprio programa para facilitar a alteração das variáveis e evitar erros de digitação.

Os comandos são opcionais pois também possuem valores padrão pré-definidos. O *clmode* determina o tipo de de simulação, enquanto o *opmode* determina o modo de operação. Os comandos suportados são apresentados a seguir.

### opmode

- **'n' (new)** - cria um novo diretório para os resultados;
- **'w' (overwrite)** - caso já exista um diretório com o mesmo nome, seus dados serão sobrescritos com os resultados do novo cálculo;
- **'c' (continue)** - continua um cálculo não terminado.

<sup>2</sup> <<http://www.crystallography.net/cod/>>

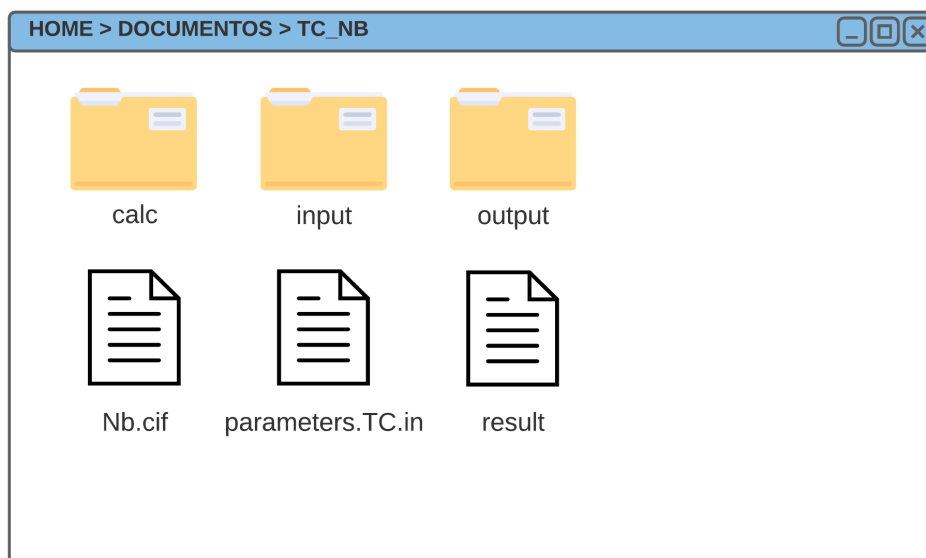
## clmode

- **'tc'** - cálculo da temperatura crítica de supercondutividade;
- **'ph'** - cálculo de fônons;
- **'eos'** - cálculo da equação de estado.

A saída do software ACESC é uma pasta cujo nome é composto pelo nome o arquivo *cif* e pelo tipo de simulação. Os arquivos de interação com o pacote QE são organizados em três subdiretórios: **'calc'**, para arquivos intermediários do QE; **'input'**, para os arquivos de entrada do QE e **'output'**, para os arquivos de saída principais QE.

A reprodutibilidade de uma simulação é garantida pela manutenção de dois arquivos no diretório de saída: o arquivo *cif* e o arquivo de parâmetros. Utilizando esses dois arquivos como entradas para uma nova simulação, obtém-se exatamente os mesmo resultados. Há ainda um terceiro arquivo, denominado "result", que contém os principais resultados de uma simulação. Na Figura 4 observa-se como são organizados os arquivos de saída do cálculo de  $T_c$  do nióbio (Nb).

Figura 4 – Diretório de resultado do cálculo de  $T_c$  com software ACESC.



Fonte: Autoria Própria, 2020.

### 4.2.2 Visão de implementação

O software ACESC foi desenvolvido em Python e é adequado para um sistema operacional Linux. Adotou-se a estratégia de construir modelos baseados nos programas do QE e nas rotinas cálculo de uma propriedade da matéria que deseja-se obter. Tanto

os modelos dos programas do QE quanto os modelos de cálculos foram implementados a partir de heranças das interfaces padrão.

Cada programa do QE possui um modelo dentro do software ASESC que implementa a interface *IQuantumEspresso()*. Assim, as especificidades como escrita de arquivo de entrada, determinação de parâmetros internos, comando de execução, são encapsuladas pelo modelo. A interface padrão garante fácil manipulação dos programas dentro do software para construir diferentes rotinas de simulação.

Do mesmo modo, cada tipo de simulação possui um modelo que implementa a interface *ICalculation()*. Cada modelo de cálculo automatiza todas as etapas de execução dos programas do QE para obter a propriedade desejada. A interface unificada permite que, independente do tipo de simulação, o programa possua um controlador único do processo. A arquitetura modular desenvolvida permite a inclusão de qualquer outra funcionalidade do pacote QE a partir da implementação de novos modelos de rotinas de cálculo, segundo a interface *ICalculation()*.

Na Figura 5 observa-se uma visão lógica de implementação do programa, contendo os principais elementos da arquitetura e a interação entre eles. Uma simulação é composta por três principais elementos: um **banco de dados**, uma **rotina de cálculo** e uma interface para o **diretório de resultados**. Cada um desses elementos são objetos dentro do programa.

A rotina de cálculo é uma composição dos objetos que representam o banco de dados, os diretórios de trabalho e os programas do QE. Ela coordena a definição dos parâmetros dos programas a partir do banco de dados, controla a transição entre os diretórios para escrita e execução dos programas e salva os dados de saída.

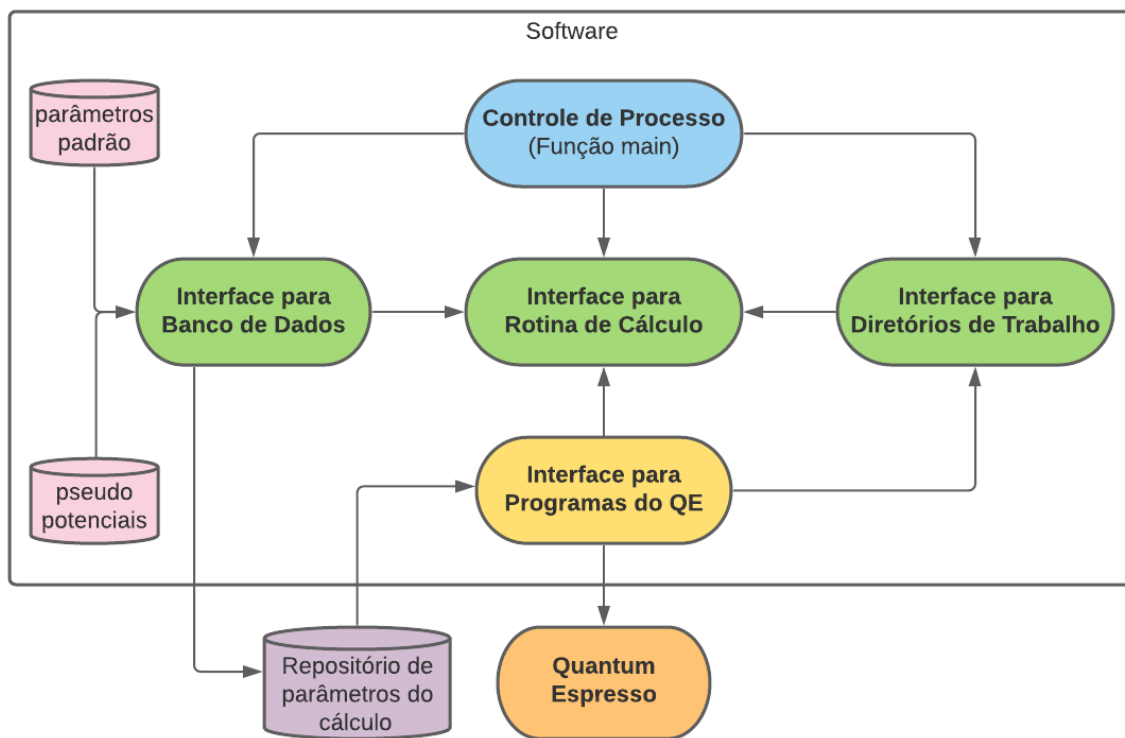
Na camada **Controle de Processo** ocorre a obtenção dos dados de entrada do usuário, a partir dos quais são criados os três objetos que definem a simulação. A fonte de dados para a criação de cada objeto varia segundo as entradas do programa. Assim, a lógica de criação de cada objeto é encapsulada em uma função 'fábrica' que recebe como entrada os comandos do usuário.

O banco de dados de 'parâmetros padrão' é implementado nas classes que derivam da interface *IMakeDatabase*, a partir de comandos da biblioteca *pickledb*. O banco de dados de pseudopotenciais é formado por arquivos de texto padrão de representação de cada pseudopotencial reunidos em um diretório específico do pacote ASESC. O 'repositório de variáveis' é um arquivo de texto gerado e manipulado pela biblioteca *pickledb*.

Nas Figuras 7 e 8 é apresentada a visão de implementação dos dois principais pacotes do programas, *core* e *util*, respectivamente. E a Figura 6 representa os diagramas UML simplificados das principais interfaces, *IQuantumEspresso* e *ICalculation*, descritas anteriormente.

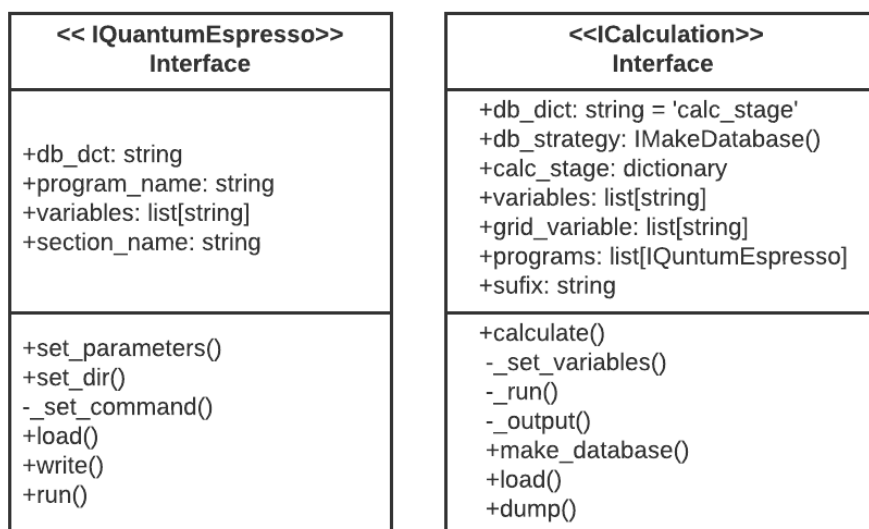


Figura 5 – Visão lógica da arquitetura do programa.



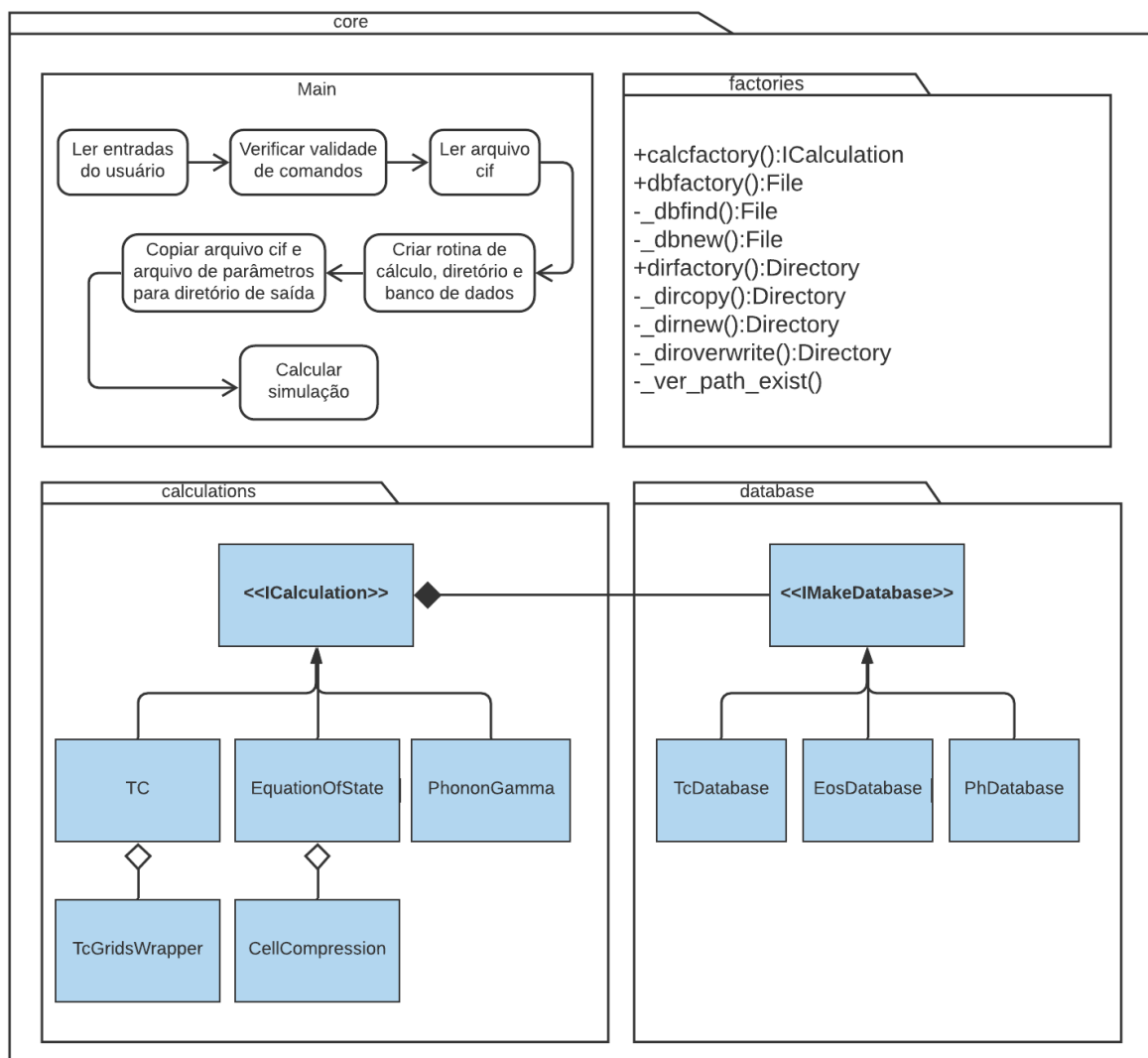
Fonte: Autoria Própria, 2020.

Figura 6 – Visão de implementação das principais interfaces, *IQuantumEspresso* e *ICalculation*



Fonte: Autoria Própria, 2020.

Figura 7 – Visão de implementação - módulos do pacote *core*.

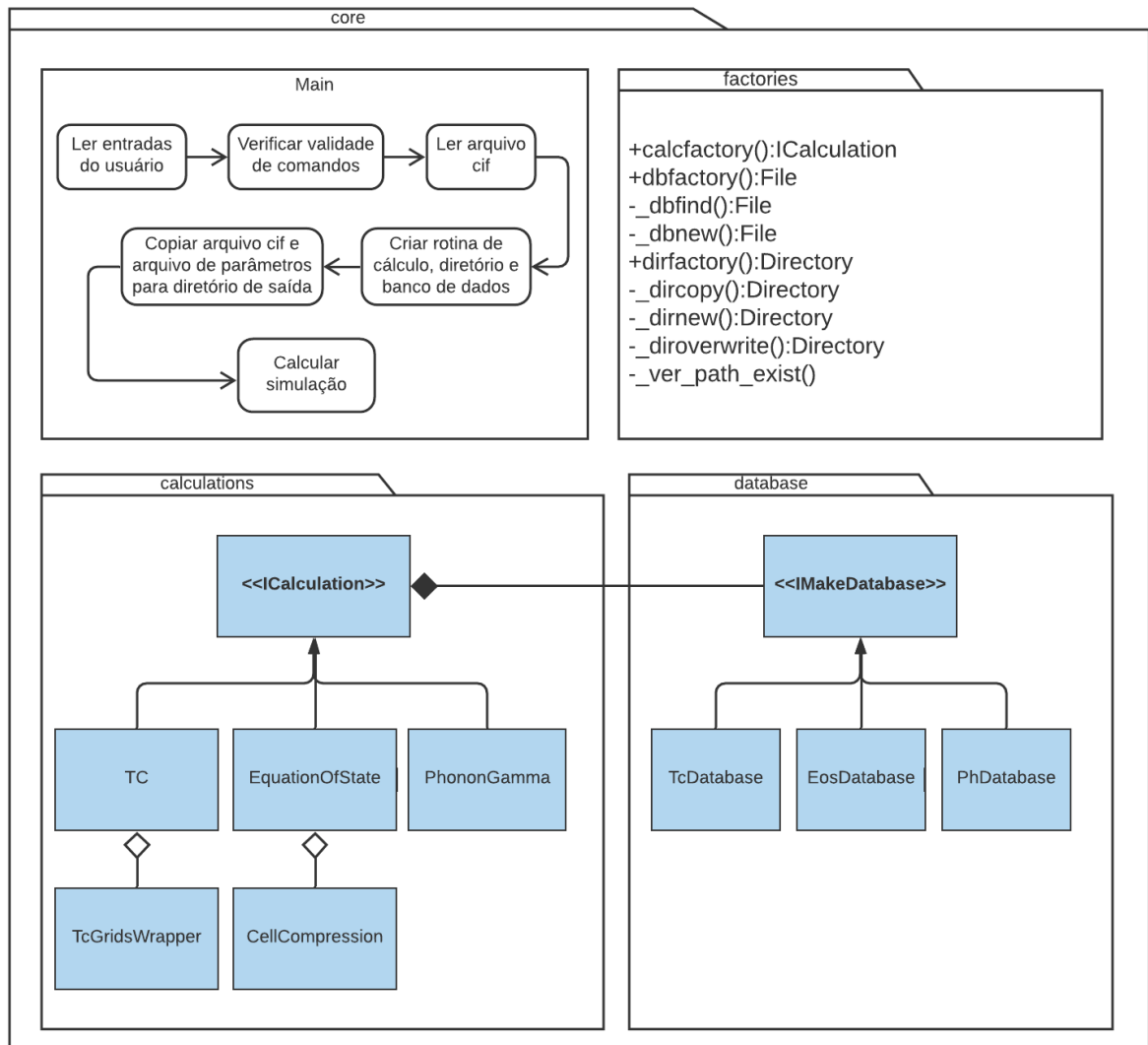


Fonte: Autoria Própria, 2020.

Observa-se na Figura 9 as principais atividades que o programa realiza em ordem de execução. A cada etapa estão associadas as bibliotecas Python utilizadas em sua implementação <sup>3 4 5 6 7 8</sup>.

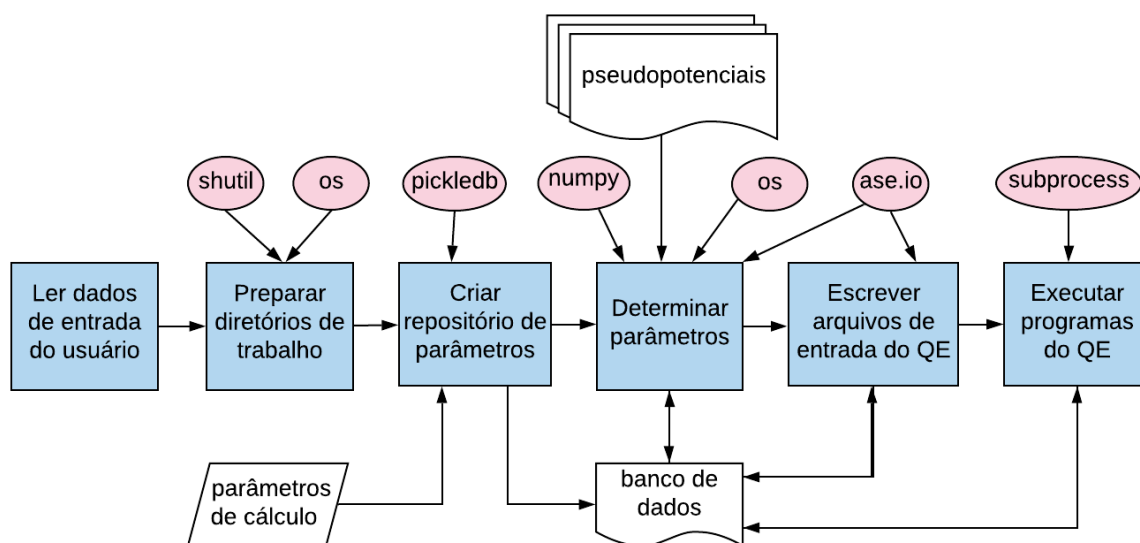
<sup>3</sup> biblioteca *shutil* <<https://docs.python.org/3/library/shutil.html>>  
<sup>4</sup> biblioteca *os* <<https://docs.python.org/3/library/os.html?highlight=os#module-os>>  
<sup>5</sup> biblioteca *pickledb* <<https://patx.github.io/pickledb/>>  
<sup>6</sup> biblioteca *numpy* <<https://numpy.org/>>  
<sup>7</sup> biblioteca *ase.io* <<https://wiki.fysik.dtu.dk/ase/index.html>>  
<sup>8</sup> biblioteca *subprocess* <<https://docs.python.org/3/library/subprocess.html?highlight=subprocess#module-subprocess>>

Figura 8 – Visão de implementação - módulos do pacote *util*.



Fonte: Autoria Própria, 2020.

Figura 9 – Visão de processo do software ACESC e bibliotecas utilizadas na implementação de cada etapa.



Fonte: Autoria Própria, 2020.

## 5 Discussão

Neste relatório foram descritas as principais atividades de estágio desenvolvidas no Laboratório Nacional de Luz Síncrotron como parte da disciplina de Estágio Integrado. As atividades culminaram no desenvolvimento do software ACESC com o objetivo de automatizar simulações de estrutura eletrônica realizadas pelo grupo de pesquisa em supercondutividade que atua na linha de luz EMA do Sirius.

O software desenvolvido terá um papel importante nas pesquisas do projeto "Um olhar com raios X em supercondutividade", desenvolvido no Sirius. A automação de cálculos de estrutura eletrônica reduz um dia inteiro de trabalho a segundos. No entanto, a ferramenta desenvolvida é operada pelo terminal em um sistema Linux. Melhorias importantes para o projeto seriam: a implementação de uma interface gráfica e a adaptação para outros sistemas operacionais.

Ainda, uma das linhas de pesquisa do projeto "Um olhar com raios X em supercondutividade" é desenvolver uma ferramenta de previsão de novos materiais supercondutores. O software ACESC permite a obtenção de propriedades de materiais com apenas sua estrutura cristalina como entrada. Isso representa um passo em direção a uma ferramenta de busca computadorizada para descobrir novos materiais com base na otimização das propriedades desejadas.

## 6 Conclusão

Neste relatório de estágio, reportou-se todo o processo de desenvolvimento de um software desde a concepção do problema de ineficiência até arquitetura final da solução, o software ACESC. Pretende-se disponibilizar o projeto desenvolvido como uma ferramenta de código aberto para toda a comunidade de pesquisa em supercondutividade. O trabalho foi executado segundo métodos de desenvolvimento ágil e de *Design Thinking* centrada no usuário.

Destaca-se a importância da flexibilidade dos métodos ágeis, pois cada desafio do desenvolvimento foi acompanhada do estudo paralelo das ferramentas para solucioná-lo. Assim, o processo foi constantemente adaptado às dificuldades e necessidades de cada etapa. Essa flexibilidade permitiu uma grande transformação de requisitos durante o desenvolvimento: a reestruturação da arquitetura para permitir a inclusão de novos tipos de simulações, além do cálculo de  $T_c$ . Dessa forma, os benefícios da automação de simulações de propriedades dos materiais podem ser expandidos para diversas áreas de pesquisa além da supercondutividade.

O *Design Thinking* também foi fundamental no desenvolvimento da solução. Manteve-se contato constante com os futuros usuários do sistema, que participaram ativamente desde a concepção inicial, testando e comentando cada etapa de desenvolvimento. Destaca-se ainda o papel central da engenharia na pesquisa científica, pois grandes descobertas partem da utilização de ferramentas computacionais. Elas permitem orientar e validar experimentos, assim como calcular propriedades de materiais conhecidos ou previstos teoricamente.

O estágio foi uma experiência extremamente enriquecedora. A aquisição dos conhecimentos necessários na área de desenvolvimento de software representou o maior desafio deste trabalho, desde o aprendizado de uma nova linguagem de programação até o estudo de paradigmas de arquitetura. Os conhecimentos das disciplinas de Introdução à Ciência da Computação, Técnicas de Programação e Informática Industrial destacam-se como base importante nesse processo.

Os conhecimentos construídos na graduação tornaram a aluna apta a buscar e compreender ferramentas para solucionar os desafios encontrados. Também constituíram uma base sólida para dialogar com outras áreas do conhecimento, como a física de matéria condensada. Essa experiência permitiu ainda a ampliação do conhecimento, por parte da aluna, das áreas de atuação de um engenheiro eletricista.

# Referências

- BI, T. et al. The search for superconductivity in high pressure hydrides. *arXiv preprint arXiv:1806.00163*, 2018. Citado na página 16.
- CORONA, E.; PANI, F. E. A review of lean-kanban approaches in the software development. *WSEAS Transactions on Information Science and Applications*, v. 10, n. 1, p. 1–13, 2013. Citado na página 15.
- DROZDOV, A. et al. Conventional superconductivity at 203 kelvin at high pressures in the sulfur hydride system. *Nature*, Nature Publishing Group, v. 525, n. 7567, p. 73–76, 2015. Citado na página 15.
- GAMMA, E. et al. Elements of reusable object-oriented software. *Reading: Addison-Wesley*, 1995. Citado na página 19.
- GEBALLE, Z. M. et al. Synthesis and stability of lanthanum superhydrides. *Angewandte Chemie*, Wiley Online Library, v. 130, n. 3, p. 696–700, 2018. Citado na página 16.
- GIANNOZZI, P. et al. Advanced capabilities for materials modelling with quantum espresso. *Journal of Physics: Condensed Matter*, v. 29, n. 46, p. 465901, 2017. Disponível em: <<http://stacks.iop.org/0953-8984/29/i=46/a=465901>>. Citado na página 16.
- HIGHSMITH, J.; COCKBURN, A. What is agile software development? *The Journal of Defense Software Engineering*, v. 15, n. 10, p. 4–9, 2002. Citado na página 15.
- LARSEN, A. H. et al. The atomic simulation environment—a python library for working with atoms. *Journal of Physics: Condensed Matter*, IOP Publishing, v. 29, n. 27, p. 273002, 2017. Citado na página 19.
- LIU, H. et al. Potential high-*t<sub>c</sub>* superconducting lanthanum and yttrium hydrides at high pressure. *Proceedings of the National Academy of Sciences*, National Acad Sciences, v. 114, n. 27, p. 6990–6995, 2017. Citado na página 16.
- LUTZ, M. *Learning python: Powerful object-oriented programming*. [S.l.]: "O'Reilly Media, Inc.", 2013. Citado na página 19.
- ROSSUM, G. V.; WARSAW, B.; COGHLAN, N. Pep 8: style guide for python code. *Python. org*, v. 1565, 2001. Citado na página 19.
- SOMAYAZULU, M. et al. Evidence for superconductivity above 260 k in lanthanum superhydride at megabar pressures. *Physical review letters*, APS, v. 122, n. 2, p. 027001, 2019. Citado na página 16.
- STANFORD, I. of Design at. An introduction to design thinking: Process guide. Hasso Platner Institute of Design Palo Alto, CA, 2010. Citado na página 15.
- TORVALDS, L.; HAMANO, J. et al. Git. *Software Freedom Conservancy*, 2005. Citado na página 20.