

UNIVERSIDADE FEDERAL DE SÃO CARLOS – UFSCAR – CAMPUS SÃO CARLOS

CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA – CCET

PROGRAMA DE PÓS-GRADUAÇÃO EM BIOTECNOLOGIA – PPGBIOTEC

DISSERTAÇÃO DE MESTRADO

**DESENVOLVIMENTO DE SISTEMA PARA CONTROLE, AQUISIÇÃO
E PRÉ-TRATAMENTO DE DADOS DE UM LIBS PORTÁTIL PARA
CARACTERIZAÇÃO ELEMENTAR EM SOLOS**

KLEYDSON STÊNIO GAIOSO DA SILVA

ORIENTADORA: DÉBORA MARCONDES BASTOS PEREIRA MILORI

São Carlos – SP

Fevereiro 2017

KLEYDSON STÊNIO GAIOSO DA SILVA

**DESENVOLVIMENTO DE SISTEMA PARA CONTROLE, AQUISIÇÃO
E PRÉ-TRATAMENTO DE DADOS DE UM LIBS PORTÁTIL PARA
CARACTERIZAÇÃO ELEMENTAR EM SOLOS**

Dissertação de mestrado, apresentada à banca no Centro de Ciências Exatas e de Tecnologia (CCET) da Universidade Federal de São Carlos (UFSCar) como parte dos requisitos para a obtenção do título de Mestre em Biotecnologia.

Linha de pesquisa:
Projeto de equipamentos e instrumentação para análise.

Orientadora: Profa. Dra. Débora Marcondes Bastos Pereira Milori.

São Carlos – SP

Fevereiro 2017

Dedico este trabalho à minha família, por todo apoio e suporte recebido ao longo dos anos; e ao meu País, com o desejo profundo e sincero de que este trabalho venha a ser útil para o crescimento tecnológico da agricultura brasileira.

AGRADECIMENTOS

Primeiramente a Deus, pela minha vida e saúde, por nunca me deixar desamparado, sempre me concedendo caminhos para seguir e força para superar desafios.

Aos meus pais, Arycimar e José, por sempre me incentivarem a buscar o sucesso onde a felicidade está.

Aos meus irmãos, Marcelo e Isabela, por serem minha força motriz, sempre me motivando a buscar o melhor.

À minha noiva, Graziene, por ser fonte inesgotável de amor, conselho, carinho, companheirismo e dedicação.

À minha orientadora, Dra. Débora Marcondes Bastos Pereira Milori, por toda paciência, suporte, incentivo, amizade e orientação na concretização de mais essa etapa de minha vida acadêmica.

Ao meu coorientador, Dr. Paulino Ribeiro Villas-Boas, pela amizade, conselhos, e auxílio constante no desenvolvimento de muitas das técnicas vitais a este trabalho.

Aos Drs. Gustavo Nicolodelli e Aida Bebechibuli Magalhães, por suas diversas sugestões e propostas para essa dissertação, bem como leitura e revisão desse trabalho.

Aos meus colegas de laboratório, Willy, Manuel, Marco, Alfredo, Anielle, Thiago, Ivan, Alex, Renata, Jonas, Camila, Amanda, Nayrê, Daniele e José, pela amizade, ajuda na concretização desse trabalho, e bons momentos do café.

À Secretária do Programa de Pós-Graduação em Biotecnologia, Cláudia Pastega, por toda disposição, agilidade e dedicação em resolver quaisquer problemas.

Aos professores do Programa de Pós-Graduação em Biotecnologia, pelo conhecimento adquirido nesses últimos dois anos.

À CAPES, pelo auxílio financeiro concedido.

À Empresa Brasileira de Pesquisa Agropecuária (Embrapa), particularmente à unidade de São Carlos – SP, Embrapa Instrumentação, por todo o suporte técnico e de pessoal essencial à concretização deste trabalho.

Aos professores integrantes da banca, pela disposição em contribuir para este trabalho.

Por fim, a todos que direta ou indiretamente participaram da realização deste trabalho ou simplesmente conviveram comigo ao longo desses últimos dois anos.

Muito obrigado a todos!

*“Com as lágrimas do tempo
E a cal do meu dia
Eu fiz o cimento
Da minha poesia.”*

Vinícius de Moraes

RESUMO

O agronegócio é uma atividade econômica de suma importância para o Brasil. Em 2015 contribuiu com 21% do PIB nacional. No entanto, a fertilidade do solo agrícola brasileiro é baixa e para manter a alta produtividade é importante fazer uso rotineiro de corretivos e fertilizantes. Associado ao tema econômico, à questão ambiental também gera grande preocupação, pois a ação antrópica vem alterando cada vez mais o clima no planeta em decorrência do aumento de gases do efeito estufa na atmosfera. Dessa forma, é necessário encontrar uma convergência sustentável entre a agricultura e o meio ambiente através do monitoramento contínuo da composição química do solo. Atualmente, a análise do solo envolve um conjunto de procedimentos físicos e químicos para avaliar diferentes características, como a quantidade de nutrientes disponíveis, o teor de matéria orgânica, a presença de elementos contaminantes, a textura do solo, entre outras. A demanda da agricultura de precisão de elaboração de mapas exige um grande número de avaliações. No entanto, ainda não há nenhuma técnica para avaliação em larga escala e a custo acessível para o produtor para a aplicação racional de corretivos e fertilizantes no solo. Com vistas nesse desafio, técnicas de caracterização rápidas, de baixo custo e que não demandam tratamento elaborado do solo são extremamente necessárias. A espectroscopia de emissão óptica com plasma induzido por laser (conhecida pelo acrônimo LIBS) é uma técnica espectro analítica que permite avaliar em tempo real a composição elementar de amostras sólidas, líquidas e gasosas. A rapidez e o baixo custo da análise, associado ao fato de ser uma técnica ambientalmente limpa, tornam-na uma técnica extremamente interessante e com grande potencial para aplicação em larga escala para levantamento de mapas de carbono e fertilidade do solo. Entretanto, atualmente os sistemas LIBS comerciais não são dedicados a análises de solo e sua interface não é amigável para usuários sem formação científica. Desse modo, este projeto de mestrado teve como objetivo desenvolver um sistema LIBS portátil de fácil utilização para análise de solos para ser implementado em laboratórios de prestação de serviços agronômicos. Além do desenvolvimento da plataforma LIBS foi necessário o desenvolvimento de softwares de controle, aquisição e tratamento dos espectros de solo. Devido sua importância no cenário ambiental, escolheu-se o carbono como elemento para calibrar e validar o LIBS portátil. Dois conjuntos de amostras de solo foram utilizados: um de teste, composto por amostras de KBr misturados com uma concentração conhecida de C, e outro de análise, formado por amostras de solos de diferentes texturas. Para as amostras de solo, obteve-se os valores de teor de carbono por meio da técnica de referência CNHS. Para o

desenvolvimento da metodologia de tratamento do sinal utilizou-se o equipamento de bancada Ocean Optics LIBS2500Plus. A análise de C para as amostras de teste obteve um coeficiente de correlação de 0,99 entre o LIBS e as concentrações de C. Para o conjunto de análise, a correlação foi 0,92 entre o LIBS e a técnica de referência. Ambos os resultados são bastante significativos.

Esta dissertação demonstrou a viabilidade de se utilizar um sistema LIBS portátil para caracterização elementar em solos. Espera-se que, após testes futuros, o equipamento proposto e projetado nessa dissertação venha a se tornar uma nova e valiosa instrumentação a ser aplicada na confecção de mapas de fertilidade e carbono do solo, como uma técnica analítica de rotina, fornecendo informações rápidas e confiáveis contribuindo cada vez mais para instrumentação dedicada à agricultura de precisão.

Palavras chave: LIBS, espectroscopia óptica, solos, caracterização elementar.

ABSTRACT

Agribusiness is an economic activity of great importance for Brazil. In 2015 it contributed with 21% of the national GDP (Gross Domestic Product). However, the fertility in Brazilian soils is low to maintain high productivity, therefore being important to routinely use correctives and fertilizers. Associated with the economic issue, the environmental one also generates great concern, as anthropic action has been increasingly changing the climate on the planet due to the increase of Greenhouse Gases in the atmosphere. In this way, a sustainable convergence between agriculture and the environment must be found through continuous monitoring of soil chemical composition. Currently, the soil analysis involves a set of physical and chemical procedures to evaluate different characteristics, such as the amount of available nutrients, soil organic matter, presence of pollutants, soil texture, among others. Precision agriculture demands a large number of assessments for creating nutrients maps. However, there is still no technique for large-scale valuation at affordable costs for the farmer in using rational application of soil improvers and fertilizers. With this challenge in mind, fast characterization techniques, at lower costs, and not requiring elaborate soil treatment are extremely necessary. Laser-induced breakdown spectroscopy (known by the acronym LIBS) is an analytical-spectrum technique that allows real-time evaluation of the elemental composition of solid, liquid and gaseous samples. The rapidity and low cost of the analysis, combined with the fact that it is an environmentally clean technique, makes it an extremely interesting with great potential for large-scale application for carbon and fertility soil mapping. However, commercial LIBS systems are currently not dedicated to soil analysis and its interface is not user-friendly for users without scientific training. Thus, this master's project aimed to develop an easy-to-use portable LIBS system for soil analysis to be implemented in agronomic services laboratories. In addition to the development of the LIBS platform, it was necessary to develop software for control, acquisition and treatment of soil spectra.

Due to its importance in the environmental scenario, carbon was chosen as an element to calibrate and validate portable LIBS. Two sets of soil samples were used: one, composed by KBr samples mixed with a well known concentration of C, and another one, composed by soil samples of different textures. For the soil samples, the carbon contents were obtained by means of the CNHS reference technique. For the development of the signal processing methodology, the Ocean Optics LIBS2500Plus equipment was used. The correlation coefficient for carbon analysis in C:KBr samples was around 0.99. For soil samples, the

correlation coefficient was 0.92 between the LIBS and the reference technique. Both results are quite significant.

This work demonstrated the feasibility of using a portable LIBS system for elementary characterization in soils. It is hoped that the proposed and designed equipment will become a valuable new instrumentation to be applied for soil mapping, as a routine analytical technique, providing fast and trustworthy measurements, contributing more and more for instrumentation in precision agriculture.

Keywords: LIBS, optical spectroscopy, soil, elementary characterization.

LISTA DE FIGURAS

FIGURA 1 – ILUSTRAÇÃO DO MAPA DE NUTRIENTES EM UMA PROPRIEDADE AGRÍCOLA UTILIZANDO AMOSTRA COMPOSTA.....	10
FIGURA 2 – PLUMA LUMINOSA DE VAPOR DE CARBONO OBTIDA POR BOYLE .	11
FIGURA 3 – PRIMEIRO MODELO DO INSTRUMENTO FABRICADO PELA JARRELL-ASH	12
FIGURA 4 – DIAGRAMA DOS PRINCIPAIS EVENTOS QUE ENVOLVEM UMA MEDIDA LIBS QUANDO PLASMA É GERADO POR UM LASER PULSADO DE NANOSEGUNDO.	13
FIGURA 5 – ESQUEMA SIMPLIFICADO DE UM EQUIPAMENTO LIBS.....	14
FIGURA 6 – LINHA DO TEMPO DE EVOLUÇÃO DO PLASMA DURANTE UMA MEDIDA LIBS	15
FIGURA 7 – COMPARAÇÃO DO SINAL OBTIDO EM UMA MEDIÇÃO LIBS ALTERANDO-SE O TEMPO DE ATRASO	17
FIGURA 8 – ESPECTRO LIBS MOSTRANDO A INTERFERÊNCIA NAS LINHAS DE C PARA UMA AMOSTRA DE ESPODOSOLO (A) E OXISOLO (B)	20
FIGURA 9 – MODELO DE CALIBRAÇÃO PARA QUANTIFICAÇÃO DE C UTILIZANDO LIBS OBTIDO POR NICOLODELLI PARA SOLOS BRASILEIROS	21
FIGURA 10 – EQUIPAMENTO LIBSCAN25+ DA APPLIED PHOTONICS (CABEÇA E CONSOLE)	22
FIGURA 11 – CABEÇA DO LIBSCAN25+ (A) E MÓDULOS ADICIONAIS DE CÂMARAS DE ABLAÇÃO (B-C)	23
FIGURA 12 – QUANTOM DA BERTIN TECHNOLOGIES EFETUANDO UMA MEDIÇÃO EM CARÇAÇA DE TV CRT.....	23
FIGURA 13 – EASYLIBS DA IVEA SOLUTION, COM UM CONJUNTO DE 3 ESPECTRÔMETROS	24
FIGURA 14 – STELLARCASE-LIBS DA STELLARNET	25
FIGURA 15 – LIBS Z-200 DA SCIAPS	26
FIGURA 16 – EQUIPAMENTO CHEMLITE DA TSI	26
FIGURA 17 – SISTEMA LIBS DE BANCADA MODELO LIBS2500PLUS: (1) FONTE DE ENERGIA DO LASER; (2) LASER; (3) CÂMARA DE ABLAÇÃO; (4) CONJUNTO DE ESPECTRÔMETROS	29
FIGURA 18 – ARQUIVO RESULTANTE DA MEDIÇÃO DE UMA AMOSTRA NO LIBS2500PLUS.....	30
FIGURA 19 – ESPECTRO CARACTERÍSTICO OBTIDO COM O LIBS2500PLUS SEPARADO POR REGIÕES ESPECTRAIS	31
FIGURA 20 – PÁGINA INICIAL DO NIST ATOMIC SPECTRA DATABASE LINES FORM.....	32
FIGURA 21 – RESULTADO DE UMA BUSCA POR EMISSÕES DE C EM UM INTERVALO DE 190 A 194 NM UTILIZANDO O NIST	33
FIGURA 22 – KIT COMPLETO DO LASER KIGRE MK-367.....	34
FIGURA 23 – ENERGIA DO LASER EM FUNÇÃO DO NÚMERO DE PULSOS E TOTAL DE DISPAROS	35
FIGURA 24 – ESPECTRÔMETRO STELLARNET EPP2000-LR.....	35
FIGURA 25 – ESPECTRO CARACTERÍSTICO OBTIDO COM O LIBS PORTÁTIL STELLARNET	36

FIGURA 26 – PORTA LIBS STELLARNET MONTADO EM BANCADA.....	37
FIGURA 27 - O RASPBERRY PI 2	39
FIGURA 28 – MONITOR PLAYTIX SPIDER ESSENCE 10,1”.....	42
FIGURA 29 – FONTE DE COMPUTADOR CORSAIR CX430	43
FIGURA 30 – CIRCUITO ELETRÔNICO DE DISPARO DO LASER, FINAL (ESQUERDA) E PROJETO (DIREITA)	44
FIGURA 31 – ESPECTRO TOTAL (4 TIROS) DE UMA AMOSTRA DE SOLO OBTIDO COM O LIBS PORTÁTIL PROPOSTO NO TRABALHO MONTADO EM BANCADA ...	46
FIGURA 32 – MODELO 3D DO PROTÓTIPO: VISTA SUPERIOR COM ABERTURA...	47
FIGURA 33 - MODELO 3D DO PROTÓTIPO: VISTA FRONTAL.....	47
FIGURA 34 - MODELO 3D DO PROTÓTIPO: VISTA TRAZEIRA	48
FIGURA 35 - MODELO 3D DO PROTÓTIPO: VISTA DIREITA	48
FIGURA 36 - MODELO 3D DO PROTÓTIPO: VISTA ESQUERDA.....	49
FIGURA 37 - MODELO 3D DO PROTÓTIPO: VISTA INTERNA DA PAREDE ESQUERDA.....	49
FIGURA 38 - MODELO 3D DO PROTÓTIPO: VISTA ISO.....	50
FIGURA 39 – ESPECTRO LIBS CARACTERÍSTICO OBTIDO COM O EQUIPAMENTO LIBS2500 PLUS.....	53
FIGURA 40 – ESQUEMA DA ESTRUTURA DE PASTAS NO PROGRAMA DESENVOLVIDO	54
FIGURA 41 – ESPECTRO LIBS TOTAL (TODOS OS TIROS) DE UMA DETERMINADA AMOSTRA OBTIDO COM O LIBS2500PLUS	55
FIGURA 42 – ESPECTRO TOTAL DE UMA AMOSTRA APÓS A REMOÇÃO DE OUTLIERS COM O SAM	56
FIGURA 43 – ESPECTRO FINAL CORRIGIDO, PRONTO PARA USO EM OUTRAS ETAPAS DA ANÁLISE	57
FIGURA 44 – PICO DE UM ELEMENTO SEM NENHUM TRATAMENTO.....	58
FIGURA 45 – PICO DE UM ELEMENTO COM CORREÇÃO DE INCLINAÇÃO.....	58
FIGURA 46 – PICO MÉDIO DE UM ELEMENTO APÓS OS TRATAMENTOS.....	59
FIGURA 47 – FIT DO PICO UTILIZANDO UMA CURVA LORENTZIANA	60
FIGURA 48 – RELATÓRIO COM OS VALORES DAS ÁREAS CALCULADAS PARA CADA AMOSTRA ANALISADA COM O MÓDULO DE FIT	60
FIGURA 49 – MODELO LINEAR PARA O ELEMENTO DE ANÁLISE UTILIZANDO LIBS	61
FIGURA 50 – ESTRUTURA DE PASTAS DO PYTHON LIBS ANALYSER	63
FIGURA 51 – UI DO PROGRAMA APÓS A EXECUÇÃO DE TODOS OS MÓDULOS ..	65
FIGURA 52 – GUI DO PYTHON LIBS ANALYSER	66
FIGURA 53 – FABRICAÇÃO DO PROTÓTIPO: BASE, LATERAIS E ABERTURA PARA A FONTE DE COMPUTADOR	67
FIGURA 54 – FABRICAÇÃO DO PROTÓTIPO: EFETUADO CORTE PARA FIXAÇÃO DO MONITOR TOUCH.....	68
FIGURA 55 – FABRICAÇÃO DO PROTÓTIPO: LASER FIXADO E BASE DO PORTA AMOSTRA COM EIXO PARA ROTAÇÃO.....	68
FIGURA 56 – FABRICAÇÃO DO PROTÓTIPO: ABERTURA LATERAL PARA INSERÇÃO DO PORTA AMOSTRA.....	69
FIGURA 57 – FABRICAÇÃO DO PROTÓTIPO: FURO LATERAL PARA RESPIRO DO COOLER DA FONTE DO COMPUTADOR E TAMPA FIXADA	69

FIGURA 58 – FABRICAÇÃO DO PROTÓTIPO: TAMPA SUPERIOR E MONITOR TOUCH FIXADO COM INCLINAÇÃO DE 50°	70
FIGURA 59 – FABRICAÇÃO DO PROTÓTIPO: EQUIPAMENTO APÓS ANODIZAÇÃO NA COR PRETA, SEM TAMPA SUPERIOR	70
FIGURA 60 – ROBÔ MIRÃ DURANTE O II WORKSHOP SOBRE LASER EM QUÍMICA ANALÍTICA	71
FIGURA 61 – APRESENTAÇÃO DOS ESPECTROS MEDIDOS DURANTE UM DOS EXPERIMENTOS COM O ROBÔ MIRÃ	72
FIGURA 62 – MODELO LINEAR PARA A QUANTIFICAÇÃO DE C NAS AMOSTRAS CONTROLADAS PARA O PICO EM 192,03 NM	74
FIGURA 63 – MODELO LINEAR PARA A QUANTIFICAÇÃO DE C NAS AMOSTRAS CONTROLADAS PARA O PICO EM 247,8 NM	75
FIGURA 64 – PICO DE K EM 693,87 NM DETECTADO NAS AMOSTRAS DE TESTE	76
FIGURA 65 – FIT UTILIZANDO UMA CURVA LORENTZIANA PARA OS PICOS DE C E AL	79
FIGURA 66 – MODELO LINEAR PARA A CONCENTRAÇÃO DE C EM FUNÇÃO DA ÁREA DO PICO PARA AS AMOSTRAS DE SOLO	80

LISTA DE TABELAS

TABELA 1 - PARTICIPAÇÃO DO AGRONEGÓCIO NO PIB DO BRASIL (EM %)	1
TABELA 2 - PIB DO AGRONEGÓCIO BRASILEIRO, 2011 A 2015, EM MILHÕES DE REAIS	2
TABELA 3 - PRODUÇÃO E IMPORTAÇÃO DE FERTILIZANTES NO BRASIL (2013 A 2016)	3
TABELA 4 – INTERVALO ESPECTRAL REFERENTE A CADA UM DOS 7 MINI-ESPECTRÔMETROS DO EQUIPAMENTO LIBS2500PLUS	31
TABELA 5 – TEORES DE C PARA AS AMOSTRAS DO GRUPO DE CONTROLE	73
TABELA 6 – CARACTERÍSTICAS DAS ÁREAS EXPERIMENTAIS UTILIZADAS PARA A COLETA DO CONJUNTO DE AMOSTRAS DE SOLO CARACTERIZADAS	77
TABELA 7 – VALORES DE CONCENTRAÇÃO DE C E TIPO DE TEXTURA DE SOLO PARA O CONJUNTO DE AMOSTRAS DE SOLOS UTILIZADOS	78

LISTA DE ABREVIATURAS E SIGLAS

LIBS	Laser Induced Breakdown Spectroscopy
PIB	Produto Interno Bruto
KBr	Brometo de potássio
CHNS	Análise Elementar (Carbono; Hidrogênio; Nitrogênio e Enxofre)
ICP-OES	Inductively Coupled Plasma Optical Emission Spectrometry
ASS	Atomic Absorption Spectroscopy
NASA	National Aeronautics and Space Administration
ANDA	Associação Nacional Para Difusão de Adubos
PC	Personal Computer (Computador Pessoal)
FAAS	Flame Atomic Absorbtion Spectroscopy
GFAAS	Graphite Furnace Atomic Absorption
ICP-MS	Inductively Coupled Plasma – Mass Spectroscopy
LA	Laser Ablation
ETV	Electrothermal Vaporisation
YAG	Yttrium Aluminium Garnet
Nd:YAG	Neodymium-doped Yttrium Aluminium Garnet
CCD	Charge Coupled Device
NIST	National Institute of Standards and Technology
EDS	Energy-dispersive X-ray Spectroscopy
RPi	Raspberry Pi
SAM	Spectral Angle Mapper
GPU	Graphics Processing Unit
CPU	Central Processing Unit
UI	User Interface
GUI	Graphical User Interface
USB	Universal Serial Bus
RAM	Random-access Memory
SoC	System on Chip
GPIO	General Purpose Input/Output

SUMÁRIO

CAPÍTULO 1 – INTRODUÇÃO	1
1.1. CONTEXTUALIZAÇÃO DO ESTADO ATUAL DO AGRONEGÓCIO NO BRASIL	1
1.2. QUESTÕES AMBIENTAIS	3
1.3. AGRICULTURA DE PRECISÃO	4
1.4. FOTÔNICA APLICADA À AGRICULTURA	5
1.5. OBJETIVOS	7
1.5.1. OBJETIVOS ESPECÍFICOS	7
CAPÍTULO 2 – REVISÃO BIBLIOGRÁFICA	9
2.1. ANÁLISE DE NUTRIENTES EM SOLOS	9
2.2. LIBS (LASER INDUCED BREAKDOWN SPECTROSCOPY)	11
2.2.1. LIBS APLICADO NA ANÁLISE DE C EM SOLOS	19
CAPÍTULO 3 – ESTADO DA ARTE DE EQUIPAMENTOS LIBS PORTÁTEIS	22
3.1. LIBSCAN 25+ (APPLIED PHOTONICS)	22
3.2. QUANTOM (BERTIN TECHNOLOGIES)	23
3.3. EASYLIBS (IVEA SOLUTION)	24
3.4. STELLARCASE-LIBS (STELLARNET)	24
3.5. Z-200 (SCIAPS)	25
3.6. CHEMLITE (TSI)	26
CAPÍTULO 4 – DESENVOLVIMENTO DO PROJETO DE LIBS PORTÁTIL	28
4.1. EQUIPAMENTOS LIBS DE REFERÊNCIA	28
4.1.1. OCEAN OPTICS LIBS2500PLUS	28
4.1.2. PORTA LIBS STELLARNET	33
4.2. DESENVOLVIMENTO DO PROJETO DE LIBS PORTÁTIL	36
4.2.1. SISTEMA LIBS	37
4.2.2. MINICOMPUTADOR RASPBERRY PI	38
4.2.3. LINUX	39
4.2.4. PYTHON	40
4.2.5. EQUIPAMENTOS ADICIONAIS: MONITOR E FONTE	42
4.3. SISTEMA PARA ACIONAMENTO DO DISPARO DO LASER	43
4.4. DRIVER LINUX PARA A AQUISIÇÃO DE SINAL	44

4.5.	MODELOS 3D PARA FABRICAÇÃO DO PROTÓTIPO	46
CAPÍTULO 5 – DESENVOLVIMENTO DE SOFTWARE DE ANÁLISE DE SINAIS 51		
5.1.	MÓDULOS DO SOFTWARE DE ANÁLISE DE ESPECTROS LIBS.....	52
5.1.1.	MÓDULO DE LEITURA DE ESPECTROS.....	53
5.1.2.	MÓDULO DE ISOLAMENTO DE ELEMENTOS (PICOS)	57
5.1.3.	MÓDULO DE AJUSTE DAS CURVAS DOS PICOS	59
5.1.4.	MÓDULO DE CRIAÇÃO DE MODELO	61
5.2.	INTERFACES DE USUÁRIO	61
5.2.1.	INTERFACE MODO TEXTO	64
5.2.2.	INTERFACE GRÁFICA.....	65
CAPÍTULO 6 – RESULTADOS E DISCUSSÕES		67
6.1.	DESENVOLVIMENTO DO PROTÓTIPO	67
6.2.	TESTES COM O SISTEMA DE DISPARO DO LASER E AQUISIÇÃO DE ESPECTROS	71
6.3.	VALIDAÇÃO DO SOFTWARE DE ANÁLISE DE ESPECTROS	72
6.3.1.	AMOSTRAS – GRUPO DE CONTROLE	73
6.3.1.1.	RESULTADOS DAS AMOSTRAS DO GRUPO DE CONTROLE.....	73
6.3.2.	AMOSTRAS – GRUPO DE ANÁLISE	76
6.3.2.1.	TÉCNICA DE REFERÊNCIA PARA ANÁLISE DE C – CHNS.....	78
6.3.2.2.	RESULTADOS DAS AMOSTRAS DO GRUPO DE ANÁLISE.....	79
CAPÍTULO 7 – CONCLUSÃO		81
CAPÍTULO 8 – REFERÊNCIAS BIBLIOGRÁFICAS		83
APÊNDICE A – CÓDIGO FONTE DA INTERFACE MODO TEXTO.....		87
	ARQUIVO PYLIBSANALYSER-1.18.PY	87
APÊNDICE B – CÓDIGO FONTE DA INTERFACE GRÁFICA		91
	ARQUIVO PYLIBSANALYSER-1.18-GUI.PY	91
	ARQUIVO INTERFACE.PY	97
APÊNDICE C – CÓDIGO FONTE DO DRIVER PYTHON/LINUX PARA OS ESPECTRÔMETROS STELLARNET EPPUSB2000-LR		103
	ARQUIVO ESTRELLA_PYTHON_CONTROLLER.PY	103
	ARQUIVO ENUMERATIONS.PY	104
	ARQUIVO FUNCTIONS.PY	106
	ARQUIVO STRUCTS.PY	109

ARQUIVO_INIT.PY 111

CAPÍTULO 1 – INTRODUÇÃO

1.1.Contextualização do estado atual do agronegócio no Brasil

O agronegócio é uma atividade extremamente importante para a economia brasileira. Entre os anos de 2011 a 2015, o Agronegócio Total (Insumos, Agropecuária, Indústria e Serviços) possuiu uma média de participação no PIB nacional de 21%, com arrecadação R\$1,28 trilhão de reais.

No ano de 2016, somente entre janeiro a outubro de 2016, houve um aumento de 4,28% em relação a 2015 (PIB CPEA, 2016). Isso significa que R\$1,34 trilhão das riquezas produzidas no País veio da agricultura e da pecuária, dos quais 68% (R\$909 milhões) vieram apenas do ramo agrícola. Tais valores encontram-se melhor detalhados nas Tabelas 1 e 2 abaixo.

Tabela 1 - Participação do Agronegócio no PIB do Brasil (em %).

	2011	2012	2013	2014	2015
PIB total BR (em R\$ milhões)	4,376,382	4,814,760	5,331,619	5,778,953	6,000,570
Agronegócio Total (A+B+C+D)	21.94	20.52	20.68	20.44	21.35
A) Insumos	2.51	2.41	2.41	2.40	2.54
B) Agropecuária	6.13	5.73	5.98	6.06	6.38
C) Indústria	6.38	5.91	5.85	5.65	5.86
D) Serviços	6.92	6.47	6.44	6.32	6.57
Ramo Agrícola (A+B+C+D)	15.51	14.71	14.37	13.87	14.53
A) Insumos	1.53	1.48	1.43	1.38	1.48
B) Agropecuária	3.56	3.42	3.37	3.28	3.47
C) Indústria	5.53	5.15	5.08	4.89	5.09
D) Serviços	4.88	4.65	4.50	4.32	4.49
Ramo Pecuário (A+B+C+D)	6.43	5.81	6.31	6.57	6.82
A) Insumos	0.99	0.93	0.98	1.01	1.06
B) Agropecuária	2.57	2.31	2.62	2.78	2.91
C) Indústria	0.85	0.75	0.77	0.77	0.77
D) Serviços	2.03	1.82	1.94	2.01	2.07

Fonte: PIB CPEA (adaptado).

Tabela 2 - PIB do Agronegócio Brasileiro, 2011 a 2015, em Milhões de Reais.

	2011	2012	2013	2014	2015
Agronegócio Total (A+B+C+D)	1,215,615	1,180,519	1,241,738	1,262,452	1,280,827
A) Insumos	139,349	138,902	144,469	147,974	152,254
B) Agropecuária	339,487	329,845	359,355	374,662	382,721
C) Indústria	353,508	339,788	351,071	349,244	351,859
D) Serviços	383,271	371,984	386,844	390,573	393,992
Ramo Agrícola (A+B+C+D)	859,379	846,041	862,950	856,689	871,779
A) Insumos	84,759	85,421	85,866	85,326	88,566
B) Agropecuária	197,302	196,936	202,101	202,744	208,291
C) Indústria	306,670	296,460	304,859	301,940	305,359
D) Serviços	270,648	267,225	270,124	266,679	269,563
Ramo Pecuário (A+B+C+D)	356,236	334,477	378,788	405,763	409,048
A) Insumos	54,591	53,481	58,603	62,647	63,689
B) Agropecuária	142,185	132,909	157,254	171,918	174,430
C) Indústria	46,838	43,328	46,211	47,304	46,500
D) Serviços	112,623	104,759	116,719	123,894	124,429

Fonte: PIB CPEA (adaptado).

Entretanto, o ramo Agrícola, mesmo tendo uma participação da economia em torno de 15% do PIB em 2016, cerca de 50% do custo de todos os produtos agrícolas provém de gastos com fertilizantes. Em grande parte, esse número se dá devido à aplicação ineficiente dos fertilizantes, que por sua vez, é fruto de uma defasagem de instrumentação para a caracterização da grande quantidade de solo agrícola do país.

Segundo dados da Associação Nacional para Difusão de Adubos (ANDA, 2017), no ano de 2016, o Brasil produziu 8.999 mil toneladas de fertilizantes, em face de 24.485 mil toneladas importadas; em outras palavras, aproximadamente 73% de todos os fertilizantes utilizados no Brasil são importados, sendo em maioria extraídos de fontes não renováveis. A Tabela 3 mostra os dados de produção e importação de fertilizantes no Brasil dos últimos quatro anos (2013 a 2016). No comparativo entre os anos de 2015 e 2016, é possível notar que houve leve queda na produção nacional de fertilizantes (-1,3%), mas a demanda por importados cresceu consideravelmente (16,1%), que é um indicativo do pleno crescimento da agricultura brasileira.

Tabela 3 - Produção e importação de fertilizantes no Brasil (2013 a 2016)

Produção Nacional de Fertilizantes (em toneladas de produto)					
	2013	2014	2015	2016	2016/2015
Dezembro	703.464	770.411	748.911	775.171	3,5 %
Janeiro a Dezembro	9.304.713	8.817.680	9.115.260	8.999.817	-1,3%
Total do Ano	9.304.713	8.817.680	9.115.260	8.999.817	
Importação de Fertilizantes (em toneladas de produto)					
	2013	2014	2015	2016	2016/2015
Dezembro	1.473.022	1.455.134	1.232.042	2.106.418	71,0 %
Janeiro a Dezembro	21.618.822	24.035.600	21.087.299	24.485.493	16,1%
Total do Ano	21.618.822	24.035.600	21.087.299	24.485.493	

Fonte: ANDA (adaptado).

1.2. Questões Ambientais

Apesar deste bom desempenho, é um consenso no setor que muito ainda pode ser melhorado através do aporte de tecnologias que ajudem o aumento de produtividade e redução de impactos ambientais. Em estudo recente, foi constatado que formas adequadas de manejo em pastagens podem contribuir muito mais para o sequestro de carbono atmosférico do que matas nativas (XAVIER, 2014), dando um indicativo de que a agropecuária também pode ser uma grande aliada em questões ambientais.

Um grande indicativo às questões ambientais se dá com a medição de carbono do solo. Atualmente, a medição de carbono é foco de atenção de convenções e acordos internacionais relacionados às mudanças climáticas globais (CCST, 2007; CLIMATE CHANGE, 2013). Contudo, é necessário que os métodos de quantificação de carbono sejam eficientes para oferecer melhores estimativas dos inventários de carbono. Mais que isso, tais métodos instrumentais precisam conferir precisão e exatidão nas determinações e serem ambientalmente limpos em termos de geração de resíduos (SEGNINI, 2008). O conhecimento mais exato e preciso sobre os teores de carbono no solo é fundamental em diversas áreas da ciência do solo, entretanto mapas de concentração precisos demandam um grande número de medidas, o que resulta em dois problemas: alto custo por amostra em técnicas de referência, e a grande quantidade de resíduos resultante de métodos químicos.

Há ainda hoje carência de informações consistentes quanto à definição e recomendação do método mais adequado para a obtenção de resultados confiáveis e satisfatórios, principalmente para uso em metodologias não convencionais como métodos espectroscópicos (SEGNINI, 2008). Além disso, quando se faz necessário grande quantidade de amostras e análises, o uso de métodos caros, laboriosos, demorados e que geram grande quantidade de resíduos químicos não são mais desejáveis.

1.3.Agricultura de Precisão

A Agricultura de Precisão busca avançar no desenvolvimento agropecuário e acompanhar a demanda por alimentos de forma sustentável, fazendo uso de tecnologias e conhecimentos científicos. Essa área da ciência preconiza que a gestão da produção agrícola considere a heterogeneidade das condições do campo e, conseqüentemente, a variabilidade espacial e temporal do sistema produtivo. Entretanto, para que seja possível mapear condições de solo, produtividade e infestação de doenças, além de sistemas georreferenciados, é imprescindível desenvolver instrumentação específica para cada um desses fatores. Neste tema a demanda por instrumentação de laboratório e de campo também é intensa.

A pesquisa de métodos, sensores e equipamentos aplicados à caracterização de solos, fertilizantes, plantas e avaliação da qualidade da água é uma área com grandes oportunidades a serem exploradas, e que podem gerar inovação na ciência e no campo com forte impacto sócio-econômico-ambiental. O monitoramento ambiental neste contexto também é extremamente importante para que se garanta a sustentabilidade dos processos de produção. Nesse contexto temos a análise da composição do solo, sendo de fundamental importância para o estudo de diversos problemas atuais, sejam eles relacionados à agricultura, como a análise de nutrientes e contaminantes em solos e fertilizantes; ou relacionados ao meio-ambiente, por exemplo, os níveis de carbono nos solos da floresta Amazônica e a avaliação de seu comportamento dinâmico sob diferentes usos do solo, sendo uma área de pesquisa em plena expansão (SCHMIDT, 2012).

A forma padrão de análise do solo é realizada através de coleta de amostras em campo e, posteriormente, o envio das mesmas ao laboratório, a fim de ser realizada a etapa de caracterização. A coleta no campo é em sua maioria (DUARTE, 1996) realizada em blocos casualizados (técnica também conhecida como “delineamento em blocos ao acaso”), para diversas profundidades, geralmente de 0 a 100 cm de profundidade. Na etapa de

caracterização, muitos dos equipamentos comumente utilizados nas técnicas de referência para análise elementar do solo, como por exemplo, Espectrometria de Emissão Atômica com Plasma Acoplado Indutivamente (ICP-OES), Espectroscopia de Absorção atômica (AAS) e Análise Elementar (Carbono; Hidrogênio; Nitrogênio; Enxofre - CHNS), são grandes, feitos para uso em laboratório e de difícil portabilização. Além disso, tais equipamentos geram resíduos químicos (resultantes de digestão em ácida), necessitam de reagentes e neutralizantes, e demandam um longo tempo de análise.

Assim, existe uma grande oportunidade para o desenvolvimento de soluções e equipamentos que ajudem no levantamento de dados de uma dada propriedade agrícola, viabilizando medidas em larga escala de modo prático e rápido, para confecção de mapas de acúmulo de carbono mais precisos, análise de macro e micronutrientes (como nitrogênio, fósforo e potássio) e identificação de contaminantes, permitindo a aplicação racional de insumos, gerando economia para o produtor, e sendo ambientalmente mais sustentável.

1.4.Fotônica aplicada à agricultura

Fotônica é o nome dado à ciência que estuda as aplicações técnicas da luz. Técnicas fotônicas são largamente aplicadas em várias áreas da ciência por sua sensibilidade, precisão e alto potencial de portabilidade. Além disso, o custo de análise e preparo da amostra é muito reduzido para estas técnicas. A medicina tem explorado muito a utilização destas técnicas, tanto para diagnóstico de doenças, como para o tratamento e realização de cirurgias.

Na agricultura, entretanto, sua utilização é muito insipiente, com poucos grupos de pesquisa desenvolvendo nova instrumentação. O potencial destas técnicas para medidas em campo é enorme, podendo se transformar em ferramentas interessantes para análises agro-ambientais. Além disso, a ausência ou o pouco preparo da amostra, fazem destas técnicas também um atrativo para aplicação em larga escala seguindo os preceitos da Química Verde, ou seja, métodos analíticos onde são reduzidos ou eliminados o uso e a geração de substâncias nocivas à saúde humana e ao ambiente (CORREA&ZUIN, 2009).

A técnica LIBS (Espectroscopia de emissão óptica com plasma induzido por laser) é uma técnica promissora, pois permite avaliar a composição elementar de amostras sólidas, líquidas e gasosas levando para isso apenas alguns segundos. Este método vem se tornando cada vez mais conhecido devido ao fato de possibilitar uma análise elementar de

forma rápida e versátil, com pouca ou nenhuma preparação das amostras (HERNÁNDEZ, 2011). Suscintamente, o LIBS utiliza um plasma gerado por pulsos de laser de alta energia para preparar a amostra e excitar termicamente os analitos que compõe a amostra, em um único passo. Após um curto intervalo de tempo, esse plasma esfria, permitindo que seja possível detectar a luz emitida quando os analitos, átomos e íons que compõe o plasma, decaem para o estado fundamental. Essa luz é coletada por um espectrômetro, convertendo o sinal luminoso em eletrônico, resultando em um espectro característico. O espectro LIBS é composto por linhas espectrais bem estreitas de todos os elementos contidos na amostra, que é uma impressão digital da amostra analisada, e a intensidade dessas linhas de emissão é proporcional à concentração do elemento na amostra analisada. Quando consideradas aplicações agroambientais, a técnica LIBS aplicada para avaliar a composição elementar de amostras como água, plantas, fertilizantes, solos e sedimentos compõe a maioria dos artigos encontrados na literatura. No solo, particularmente, a técnica tem sido utilizada para determinar teores de carbono e matéria orgânica, nutrientes, contaminantes e determinação de textura e pH do solo (FERREIRA, 2015).

No Laboratório de Óptica e Fotônica da Embrapa Instrumentação já foram desenvolvidos diversos estudos comprovando a aplicabilidade do LIBS em agricultura. Em fertilizantes, a técnica foi utilizada para quantificação de fósforo (P) e potássio (K) em um conjunto bem heterogêneo, contendo fertilizantes comerciais, rochas fosfáticas e organominerais (MARANGONI, 2015; CERVANTES, 2016). Em solos, desenvolveu-se uma metodologia eficiente para a caracterização de carbono (C) em solos Amazônicos, chegando a resultados bem expressivos (NICOLODELLI, 2014), bem como métodos para rápida análise de texturas (VILLAS-BOAS, 2016).

Entretanto, apesar do LIBS possuir vantagens por ser mais rápido e precisar de pouco ou nenhum preparo de amostra, um LIBS convencional continua sendo inviável para funcionamento fora de um laboratório, dado seu tamanho e alta demanda de energia elétrica (fonte do laser, espectrômetros, computador, além de refrigeração externa). Assim, existe a necessidade da criação de um equipamento que una todas as vantagens do LIBS com a possibilidade de usar o sistema nos mais variados ambientes, de tamanho reduzido e de fácil transporte para efetuar medidas mais próximas do local de coleta, em um posto avançado ou laboratório móvel. Na literatura alguns trabalhos mostram que é possível o desenvolvimento de tal equipamento (OSORIO, 2011; RAKOVSKÝ, 2014).

Por todas as vantagens que a técnica apresenta, o LIBS vem ganhando bastante atenção mundial e está em franco desenvolvimento. Os principais interesses estão focados na miniaturização dos componentes e no desenvolvimento de equipamentos portáteis. A potencialidade do LIBS despertou o interesse de importantes agências, como a NASA (National Aeronautics e Space Administration), interessadas na criação de sistemas portáteis e de operação remota. Um caso recente com grande repercussão foi a missão espacial enviada a Marte com o robô “Curiosity”, que foi equipado com um equipamento LIBS para caracterização rápida do solo marciano. Agências militares também estão investindo no LIBS, onde o intuito é a construção de equipamentos de uso remoto ou à distância para identificar explosivos e materiais radioativos ou tóxicos.

No entanto, a criação de um LIBS, portátil ou não, tem diversos desafios:

- 1) Os eventos associados à medição ocorrem em uma escala de tempo muito curta (da ordem de nano segundos), o que faz com que a eletrônica envolvida tenha que ser muito precisa;
- 2) As empresas que fornecem os equipamentos comerciais investem pesadamente em desenvolvimento de software, garantindo que seus produtos saiam para o mercado com a certeza de que os operadores consigam efetuar medidas de modo simples e eficiente;
- 3) Por mais simplificado que seja o software de aquisição do espectro LIBS, a análise do sinal obtido não é trivial, e demanda um alto grau experiência com a técnica por parte do operador/analizador, o que poderia dificultar a disseminação da técnica.

1.5.Objetivos

Esse trabalho de mestrado teve como objetivo o desenvolvimento de um protótipo de um LIBS portátil. Para tal, foram desenvolvidos sistemas para controle de disparo do laser e acionamento de espectrômetros, aquisição de sinal e pré-tratamento de dados, mas ainda assim garantindo um uso simples e intuitivo.

1.5.1. Objetivos específicos

- Fazer testes com espectros de um LIBS comercial, utilizando amostras controladas e amostras de solo, visando obter uma metodologia confiável para a análise de carbono, devido à sua importância no contexto ambiental;
- Definir metodologia de análise, criar softwares que possam ser utilizados tanto em computadores convencionais (PCs) quanto em sistemas embarcados;
- Desenvolver um sistema (programa e circuito eletrônico) que permita o acionamento via software do disparo do laser;
- Desenvolver um driver Linux para o espectrômetro utilizado no sistema LIBS portátil, que seja passível de uso em múltiplas arquiteturas de computadores: 32 bits (i386), 64 bits (amd64) e ARM, garantindo sua fácil aplicação em diversas áreas;
- Criar uma interface de uso amigável e intuitiva para o usuário final/operador do sistema seja em modo texto (*User Interface - UI*), ou em modo gráfico (*Graphical User Interface - GUI*);
- Projetar um LIBS portátil, definindo equipamentos adicionais a serem adquiridos e utilizando todos os softwares desenvolvidos, a fim de se obter um espectro de uma amostra de solo.

CAPÍTULO 2 – REVISÃO BIBLIOGRÁFICA

2.1. Análise de nutrientes em solos

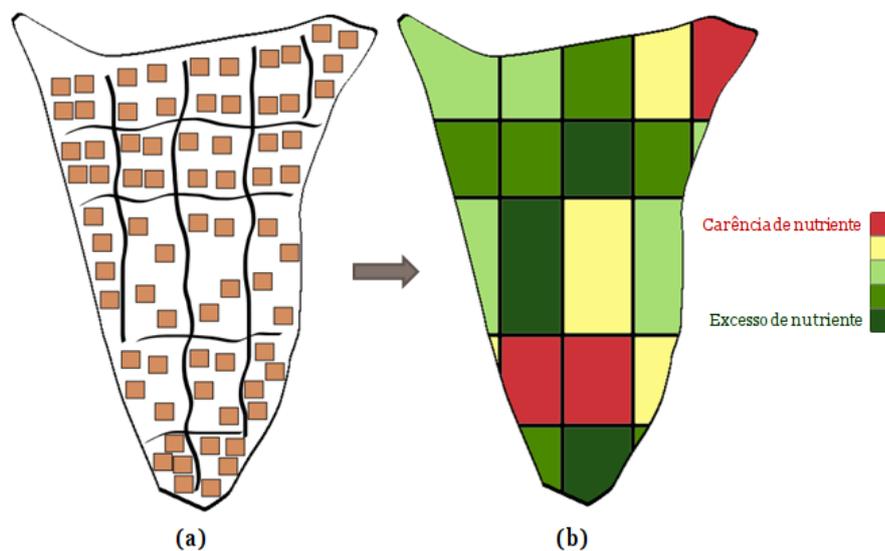
As principais causas que podem limitar os rendimentos das culturas e a qualidade dos produtos agrícolas são: desperdícios de insumos, déficits nutricionais das plantas, pragas e doenças, e a forte dependência nacional de importação de fertilizantes. Inclusive, a sustentabilidade ambiental dos sistemas de produção cada vez mais tem sido cobrada no mercado nacional e internacional. Dessa forma, para um sistema agrícola, o monitoramento de elementos minerais é prática comum, realizada com o objetivo de mapear o solo, evidenciando suas reais necessidades. Com esse mapeamento é possível efetuar uma correção com relação à falta de nutrientes essenciais ou à remediação, além da possibilidade de detectar altas concentrações de metais pesados. A prática comum da análise de solo permite a tomada de decisão por parte do agricultor em tempo hábil e impede perdas no plantio ou até mesmo a comercialização de alimentos contaminados garantindo produtividade e lucratividade para a sua lavoura.

O método mais comum para a análise elementar dos teores disponíveis no solo é a amostragem de determinada quantidade de solo e o envio da amostra a um laboratório especializado. Em grande maioria, usa-se apenas a camada arável (de 0-20 cm de profundidade) para a análise, mas em alguns casos, as amostras de solo podem também representar camadas mais profundas como 20-40 ou 40-60 cm (em pesquisas mais elaboradas, visando detectar a percolação de elementos em camadas mais profundas, pode-se utilizar até 100 cm de profundidade). Essas amostras irão representar todo o universo de solo que se quer avaliar. Uma amostra de solo da camada arável, por exemplo, é composta pela mistura de 15 a 20 amostras simples retiradas dessa camada de uma área da propriedade rural, em zigue-zague, de modo a compor uma amostra representativa da área que se quer avaliar. Essa área deve representar, no máximo, 20 ha de terra (padrão para o Estado de São Paulo ou em torno 40 Mg de solo (CAMARGOS, 2005).

No laboratório, a amostra é submetida a um processo de preparo, o qual tem como objetivo disponibilizar os analitos em solução, uma vez que a maioria das técnicas analíticas instrumentais exige a introdução da amostra no estado líquido. Inicia-se o procedimento analítico com um preparo de secagem, moagem e peneiragem do solo e a retirada de um pequeno volume para cada tipo de determinação que deverá ser feita, ou seja,

cada elemento (ou grupo de elementos) no solo é feito através de um procedimento diferente. Dentre as análises de rotina a serem efetuados cita-se: fósforo e potássio, pH do solo, cálcio, magnésio e enxofre, micronutrientes, matéria orgânica e argila. Nessa etapa são utilizados diferentes reagentes e extratores capazes de discriminar os diferentes conteúdos que compõe o solo. Eles são agitados de 10 minutos a 16 h, dependendo da determinação que se deseja fazer. Nessa fase, também é necessário cuidado para que não haja contaminação da amostra que está sendo manipulada durante as análises. Finalmente, o último passo é a interpretação e recomendação dos resultados. A partir dos dados obtidos das condições do solo é possível fazer a recomendação correta para o uso de corretivos e fertilizantes. Para tanto, são utilizadas classes de teores obtidas a partir de experimentos de campo com diversas culturas. Em geral, um bom resultado da fertilidade do solo depende da qualidade das partes de cada um de seus componentes, como a amostragem do solo que deve ser representativa e cuidados com o preparo da amostra. A Figura 1 abaixo ilustra uma propriedade agrícola utilizando essa metodologia: em (a) temos a separação do terreno em blocos, sendo coletadas quatro amostras por região, em (b) o mapa gerado após a análise.

Figura 1 – Ilustração do mapa de nutrientes em uma propriedade agrícola utilizando amostra composta



Fonte: https://commons.wikimedia.org/wiki/File%3ACaf%C3%A9_G%C3%A1lia.jpg
(Adaptado).

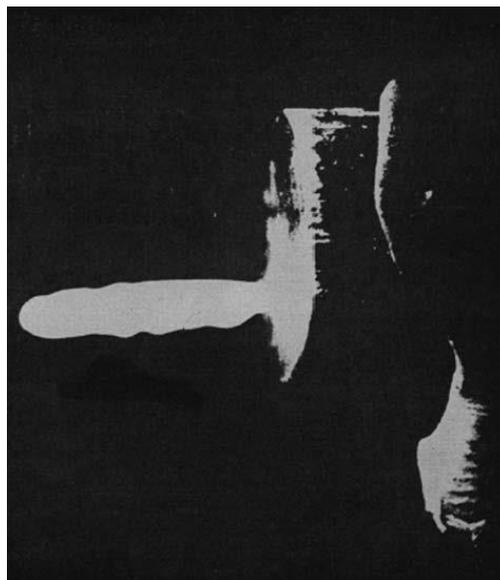
Nos últimos anos, muitas pesquisas foram realizadas enfocando como tema o preparo da amostra. Esses estudos tiveram como objetivo a obtenção de procedimentos simples e eficientes para o preparo da amostra, contemplando a redução dos erros sistemáticos e a redução nas quantidades de reagentes utilizados nessa etapa. A maioria dos métodos de

preparo, gerados por esses estudos previam a determinação dos analitos utilizando técnicas analíticas consagradas, tais como espectrometria de absorção atômica (FAAS e GFAAS), espectrometria de emissão ótica com plasma de argônio indutivamente acoplado (ICP-OES) e espectrometria de massa acoplada com plasma indutivo (ICP-MS). No caso da análise direta de sólidos, sistemas de ablação por laser (LA) e vaporização eletrotérmica (ETV) foram utilizados, acoplados a ICP-OES e ICP-MS. Apesar da utilização de técnicas instrumentalmente precisas e capazes de detectar níveis de concentração da ordem de $\mu\text{g l}^{-1}$ e pg l^{-1} , a etapa de preparo continua a sendo o “calcanhar de Aquiles” da sequência analítica. Nesse contexto, a análise direta da amostra, sem pré-preparo, fica consagrada como situação analítica ideal.

2.2.LIBS (Laser Induced Breakdown Spectroscopy)

O surgimento do LIBS data da década de 1960, concomitante com a descoberta do Laser (BAUDELET&SMITH, 2013). Em janeiro de 1962, G. C. Dacey, então diretor do Laboratório de pesquisas em eletrônica e estado sólido do *Bell Telephone Laboratories* reportou em um periódico na *Science* um trabalho não publicado de W.S Boyle, o qual havia formado um plasma luminoso utilizando um laser de rubi com 10 J e 1 ms de pulso (Figura 2).

Figura 2 – Pluma luminosa de vapor de carbono obtida por Boyle

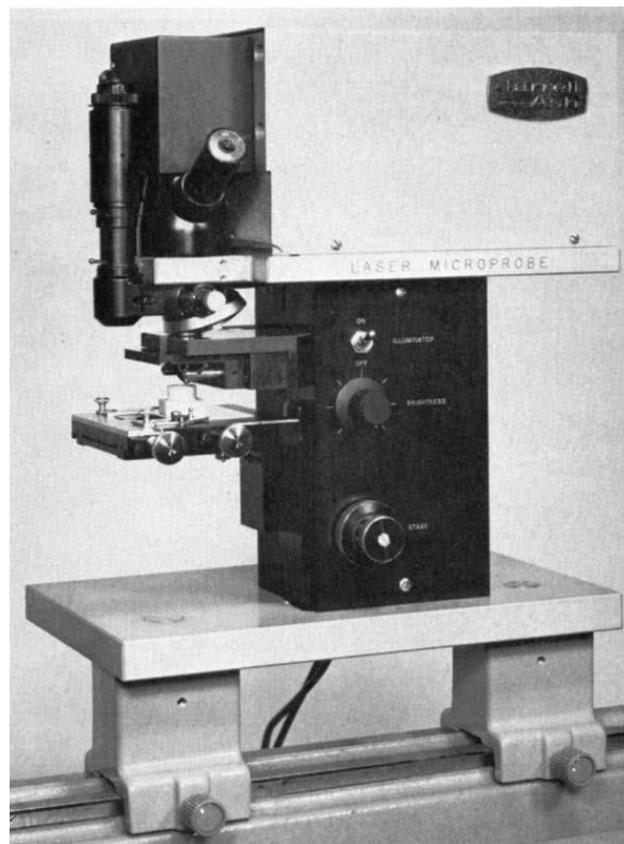


Fonte: BAUDELET&SMITH.

Inspirado pelos resultados obtidos no Laboratórios Bell, Peter Franken, da Universidade de Michigan começou experimentos similares, utilizando um laser de ruby

emprestado da companhia *Trion Instruments*. A Trion formou uma parceria com Fred Brech, vice-presidente de pesquisas na *Jarrel-Ash* e, em questão de meses, Brech e Lee Cross, da Trion, obtiveram resultados interessantes como o espectro de ferro obtido após focalizar um laser de Ruby de 5 J em uma lâmina montada entre dois eletrodos de cobre conectados a um capacitor carregado. Tais resultados foram divulgados no *X Colloquium Spectroscopicum Internationale*, que ocorreu na Universidade de Maryland em junho de 1962. O artigo não foi publicado, mas o resumo foi aceito na *Applied Spectroscopy*, obtendo certa notoriedade. Rapidamente o equipamento de Brech foi comercializado pela Jarrel-Ash (Figura 3) e anunciado na coluna “*What’s new*” (“O que há de novo”) na *Applied Spectroscopy* em outubro de 1962. A técnica foi chamada de “Espectroscopia de Micro-emissão”, e o equipamento foi comercializado na época por U\$12.000,00 (aproximadamente U\$100.000,00 nos dias atuais).

Figura 3 – Primeiro modelo do instrumento fabricado pela Jarrel-Ash

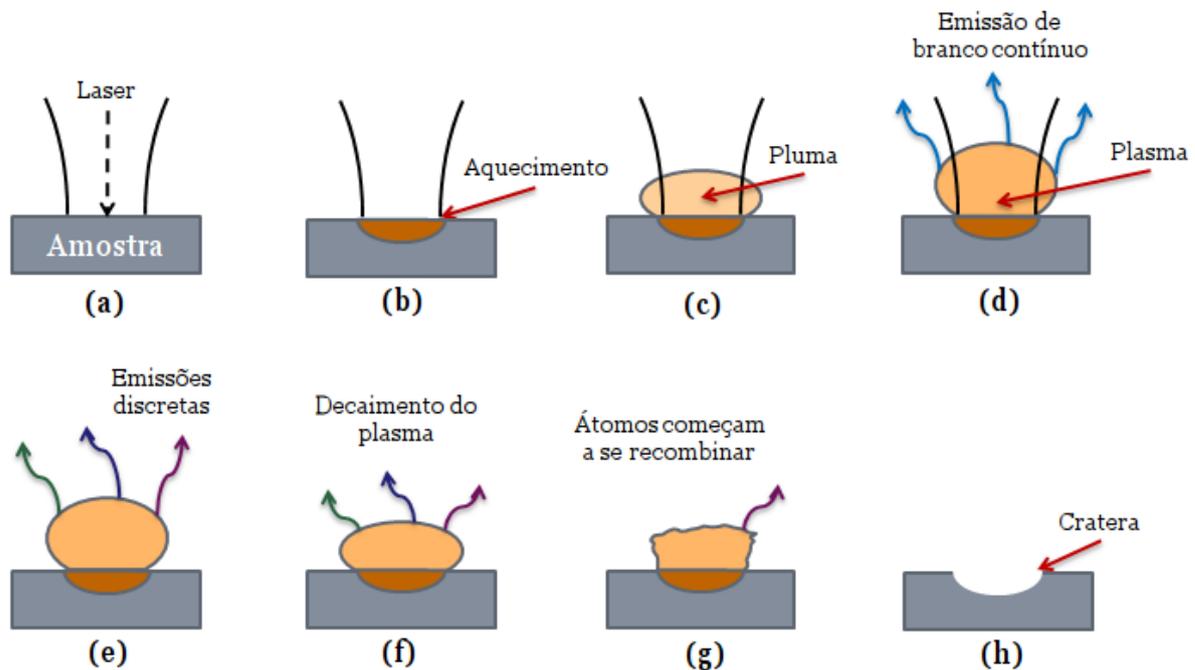


Fonte: BAUDELET&SMITH.

A técnica evoluiu paulatinamente desde então, mas teve um grande salto somente a partir da década de 1980, com o advento de lasers mais potentes e a menor custo, bem como detectores e espectrômetros mais robustos e de maior resolução. Hoje podemos

definir a técnica LIBS (*Laser Induced Breakdown Spectroscopy*) como um tipo de espectrometria de emissão atômica que utiliza a geração de plasma através de pulsos de alta potência (SACCHI, 1991) para obter informações elementares de uma amostra. Podemos verificar os principais eventos que envolvem uma medida de um LIBS com laser de nano segundo na Figura 4 abaixo.

Figura 4 – Diagrama dos principais eventos que envolvem uma medida LIBS quando plasma é gerado por um laser pulsado de nanosegundo.



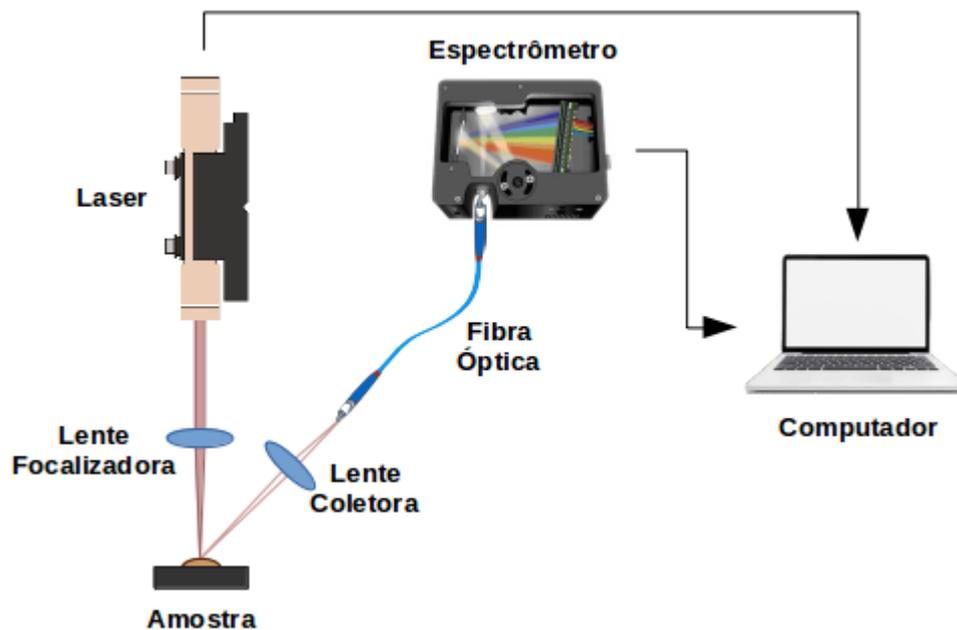
Fonte: NOLL (Adaptado).

A focalização (a) do feixe laser de alta potência na amostra (da ordem de GW/cm^2) provoca um aquecimento em sua superfície (b) e posteriormente a ejeção/ablação de uma pequena porção da matéria, da ordem de nanogramas, formando uma pluma de plasma (c). Como o pulso ainda está excitando a amostra, a temperatura continua aumentando até ordens superiores a 50.000 K, resultando na ignição do plasma (d). Nessa temperatura, o material se dissocia em íons e átomos excitados, emitindo um contínuo de radiação (esse contínuo não corresponde a uma informação útil para a caracterização de materiais). Os microsegundos iniciais de ignição do plasma são seguidos por um período mais longo quando as linhas de emissão dos elementos começam a aparecer e a emissão mais alargada começa a decair. Nessa fase, a temperatura do plasma é da ordem de 10.000K. Ao longo dessa fase é possível medir linhas de emissão atômicas e iônicas dos elementos presentes, e identificá-los

(e-g). Após a extinção do plasma, que dura em torno de 15 μ s, sobra apenas uma pequena cratera na amostra (h).

Um equipamento convencional que utiliza LIBS é formado por: laser, espectrômetro, lentes de focalização e coletas apropriadas, fibras óticas, sistema de aquisição de dados, controle e sincronização (como esquematizado na Figura 5). O laser mais comumente usado é um laser de estado sólido de cristal YAG (yttrium aluminium garnet; $Y_3Al_5O_{12}$) dopado com neodímio (Nd:YAG) emite radiação em um comprimento de onda característico de 1064 nm com pulso de 10 ns de duração. Também são utilizados outros tipos de laser, como o Excimer, que emite radiação na região do visível e ultravioleta. O espectrômetro mais amplamente utilizado é do Czerny-Tuner, que possui boa resolução e robustez para equipamentos portáteis. Outro tipo de espectrômetro é com arranjo Echelle, utilizado em sistemas LIBS devido sua alta resolução e por permitir uma aquisição com um range espectral bem maior do que o Czerny-Tuner. Em nosso equipamento de referência, é utilizado um arranjo Czerny-Tuner para dispersar a radiação ao longo de uma CCD (charge coupled device), realizando uma aquisição dentro do intervalo espectral de 188 até 966 nm, podendo detectar qualquer elemento da tabela periódica que emita nessa região.

Figura 5 – Esquema simplificado de um equipamento LIBS

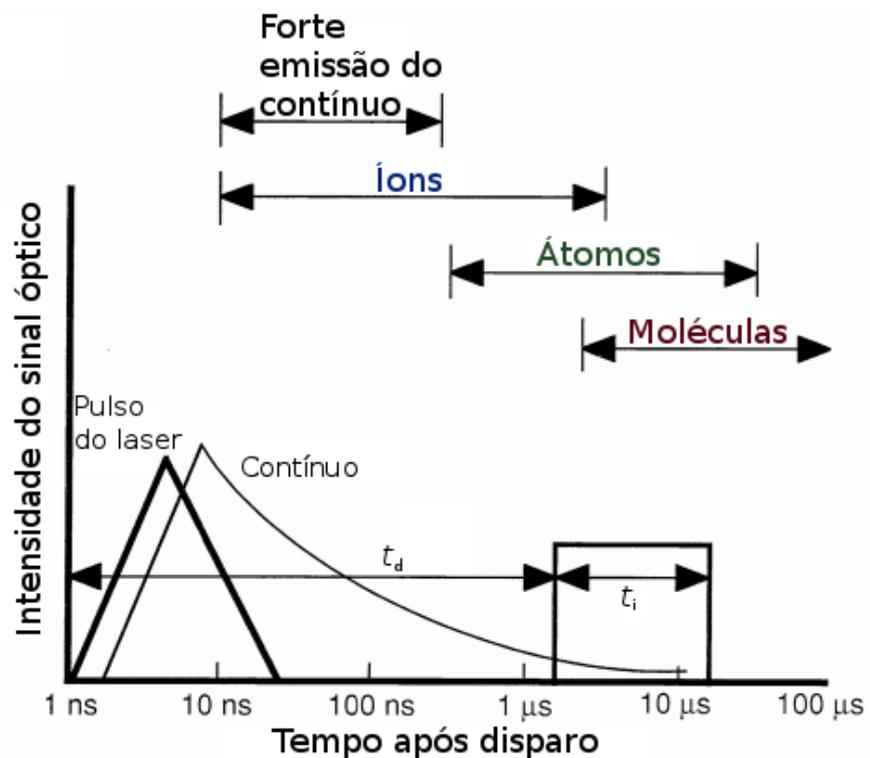


Fonte: Autoria própria

Com relação à aquisição do sinal obtido, existem dois parâmetros de suma importância envolvidos: o tempo de integração (*integration time* – t_i), que é o quanto o

obturador do espectrômetro ficará aberto coletando a luz originada do plasma, em geral o intervalo de tempo é da ordem de $10 \mu\text{s}$; e o tempo de atraso (*delay time* – t_d), que é o tempo de atraso necessário entre o disparo do pulso de ablação e o início da aquisição da radiação pelo espectrômetro. Esse intervalo de tempo em geral varia entre 1 a $10 \mu\text{s}$. A Figura 6 apresenta uma linha do tempo da medição, mostrando evolução temporal desde a sua formação. Uma vez que o laser focalizado atinge a amostra, durante um curto intervalo de tempo da ordem de 4 a 10 nano segundos, ela irá absorver a energia dessa radiação e iniciará um processo de fragmentação onde a fração de um micrograma de material será ejetado da superfície do material. Esse material, 1-2 ns depois do início do disparo do laser entra em ignição, formando o plasma. Nesses instantes iniciais, observa-se apenas uma luz branca, ou a emissão de um contínuo. Terminado o disparo do laser, entre 8-10 ns, os microsegundos iniciais de ignição do plasma são seguidos por um período mais longo, quando as linhas de emissão dos elementos começam a aparecer conforme a emissão mais alargada começa a decair. Nessa fase, já em processo de resfriamento, é possível detectar as emissões discretas de íons e átomos. Por isso é importante fazer o melhor ajuste do t_d e t_i , determinar a melhor janela temporal para se ligar o espectrômetro e se evitar a aquisição da emissão do contínuo, que mascara a informação realmente importante.

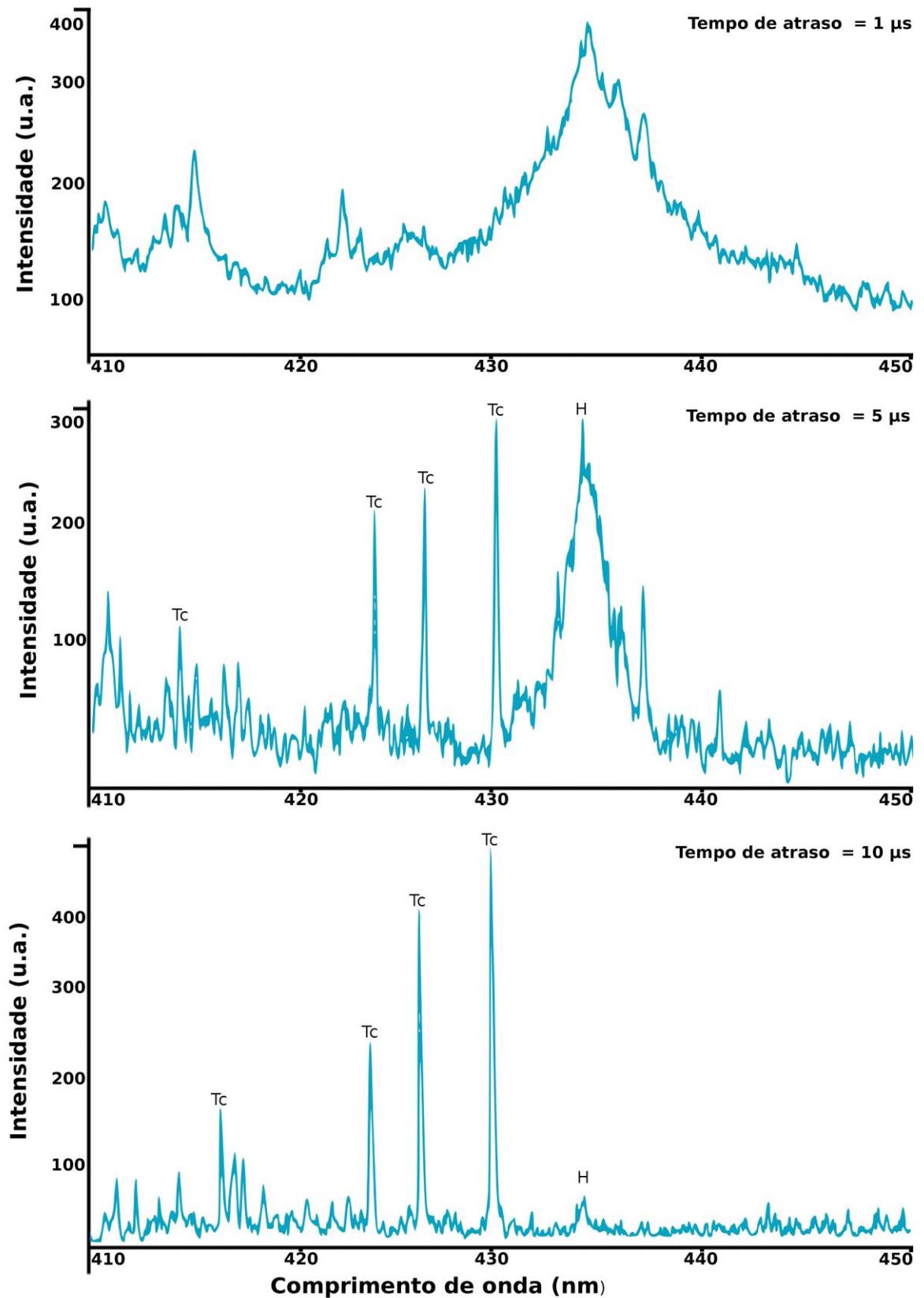
Figura 6 – Linha do tempo de evolução do plasma durante uma medida LIBS



Fonte: MIZIOLEK; PALLESCHI; SCHECHTER (Adaptado).

O tempo de delay está intimamente relacionado à quais elementos poderão ser detectados com a técnica. Elementos mais leves tendem a ser melhor detectados com um atraso menor que elementos mais pesados. No exemplo da Figura 7, o tecnécio (Tc) e o hidrogênio não podem ser detectados com um tempo de delay de 1 μs , havendo forte interferência do branco, mas já é possível detectar suas linhas com um t_d de 5 μs ; já aos 10 μs de t_d , as linhas de Tc ficam mais evidentes, mas a linha do H fica quase indetectável.

Figura 7 – Comparação do sinal obtido em uma medição LIBS alterando-se o tempo de atraso



Fonte: APPLIED PHOTONICS (Adaptado).

O LIBS apresenta algumas vantagens quando comparado com outras técnicas espectroscópicas (ICP-OES e EDS, CHNS e AAS), como: ser um método de análise semi-destrutivo, pois utiliza uma ínfima quantidade de material (da ordem de μg), sendo ideal para análise de materiais geológicos e obras de arte; não aquece as redondezas do material, dispensando sistemas de refrigeração; capacidade de detecção da heterogeneidade da amostra, devido à pequena área de focalização do laser; uso relativamente simples, pois a amostra usada é feita com pouco ou nenhum preparo (não necessita de digestão em ácido, por exemplo); possibilidade de operação remota e mapeamento de superfície e profundidade de solos.

A desvantagem que a técnica apresenta é a variabilidade no sinal obtido, pois como o LIBS não possui etapa de preparação de amostras, os efeitos de matriz são importantes, fazendo com que os resultados dependam da característica do material estudado. Assim, não podemos utilizar a mesma curva de calibração para analisar solos e rochas, necessitando sempre de uma calibração para cada tipo de material. Como bons candidatos a técnicas de referencia, podemos destacar os já citados ICP-OES e AAS; mas uma vez calibrado, o tempo de análise e quantificação das amostras com o LIBS se dá em apenas alguns segundos.

Também, outro fator a se considerar é que, como o laser é focalizado em uma região muito pequena da amostra, apenas um disparo pode não ser suficiente para garantir homogeneidade na medição. Associando essa condição ao fato de que também pode haver leves variações nas condições de formação do plasma, levando a intensidade relativa obtida a uma variabilidade intrínseca, chegamos a um ponto crucial da medição de uma amostra com LIBS: fazer diversos tiros por amostra, de modo que possa ser extraído um espectro médio representativo. Outra desvantagem é o limite de detecção, que pode ser de até duas ordens de grandeza em comparação com as técnicas químicas. Contudo, esse cenário vem cada vez mais sendo superado por novas metodologias de análise.

Por todas as vantagens que a técnica apresenta, o LIBS vem ganhando bastante atenção mundial e está em franco desenvolvimento. Os principais interesses estão focados na miniaturização dos componentes e no desenvolvimento de equipamentos portáteis. A potencialidade do LIBS despertou o interesse de importantes agências, como a NASA (National Aeronautics e Space Administration), interessadas na criação de sistemas portáteis e

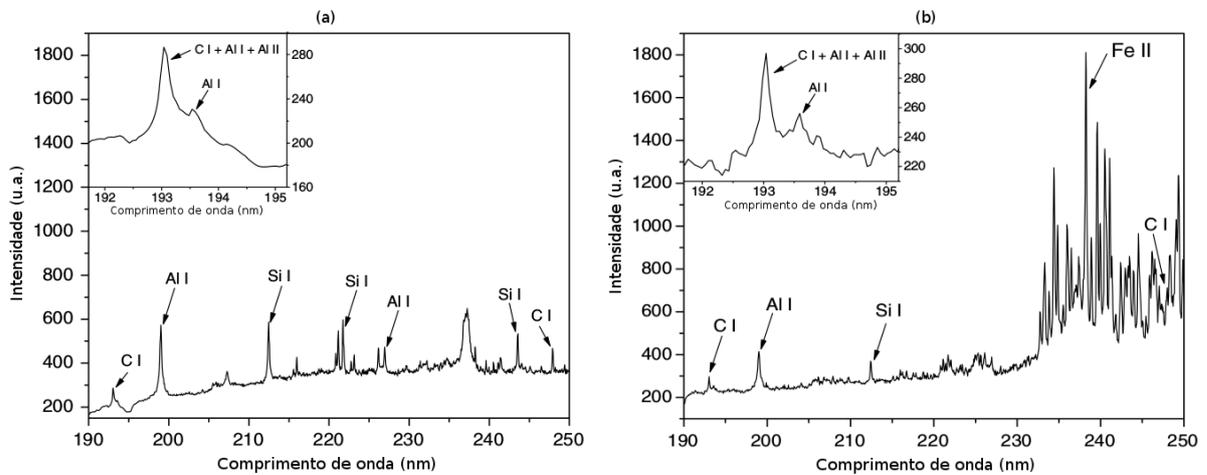
de operação remota. Um caso recente com grande repercussão foi a missão espacial enviada a Marte com o robô “Curiosity”, que foi equipado com um equipamento LIBS para caracterização rápida do solo marciano. Agências militares também estão investindo no LIBS, onde o intuito é a construção de equipamentos de uso remoto ou à distância para identificar explosivos e materiais radioativos ou tóxicos.

2.2.1. LIBS aplicado na análise de C em solos

Para análises de solos, a principal vantagem do LIBS é o potencial de se fazer uma análise multielementar em única medida, com mínima preparação da amostra, rápida velocidade de processamento de amostras e o baixo custo de análise, quando comparado com os métodos de referência. Nesse contexto, o LIBS vem sendo considerado como um método alternativo para quantificar contaminantes, macro/micronutrientes e o Carbono (C) em diferentes solos (FERREIRA, 2011; NICOLODELLI, 2015). Outra característica que torna a técnica LIBS atrativa para experimentos em campo é que devido à sua configuração totalmente ótica, isto é, a amostra é acessada óticamente, o sistema permite ser compactado e portabilizado para medidas *in situ*.

O laboratório de ótica e laser da Embrapa Instrumentação já vem trabalhando há mais de 15 anos com aplicação da técnica LIBS em amostras de interesse da agricultura e meio ambiente. Dentre os diferentes estudos conduzidos pelo grupo, a proposta de análise de quantificação de C tem grande importância uma vez que ele pode oferecer contribuições relevantes nas análises ambientais e de estoque de Carbono. Após uma pesquisa na literatura atual, estudos de importantes grupos de pesquisas no Brasil e no mundo utilizam duas linhas de C, em 193,03 e em 247,8 nm, para a quantificação de Carbono por meio da técnica LIBS (CREMERS, 2001; EBINGER, 2003; DA SILVA, 2009; BELKOV, 2009; LEWIS, 2011; DIAZ, 2012; AYYALASOMAYAJULA; 2012). Entretanto, para solos brasileiros que são ricos em Ferro (Fe) e Alumínio (Al), tais linhas são fortemente interferidas pelas linhas de emissões desses dois elementos. A Figura 8 mostra espectros LIBS de dois tipos de solos brasileiros, espodosolo (a) e oxisolo (b), onde é possível verificar a interferência nos picos.

Figura 8 – Espectro LIBS mostrando a interferência nas linhas de C para uma amostra de espodosolo (a) e oxisolo (b)



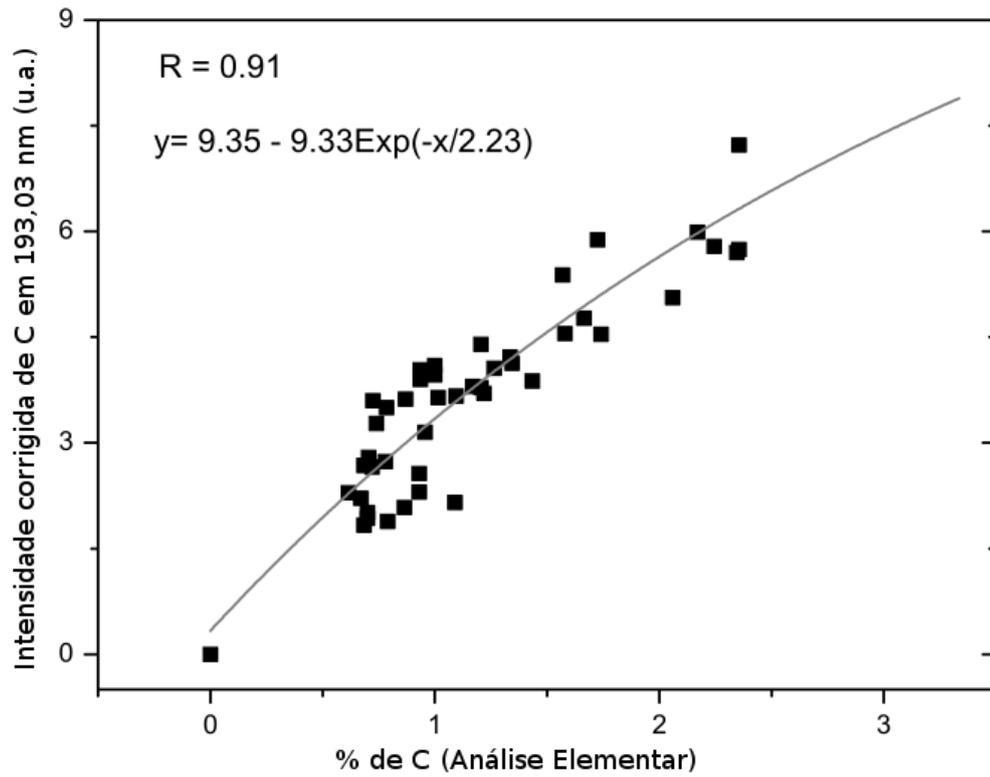
Fonte: NICOLODELI (Adaptado).

Em um trabalho recente publicado pelo grupo (NICOLODELI, 2014), foi desenvolvido uma metodologia para correção da interferência de alumínio na linha de 193,03 nm e que mostrou resultados extremamente promissores. A Equação 1 mostra uma relação que nos permite inferir a área do pico de C em 247,8 nm através das áreas dos picos do C em 193,03 e do Alumínio.

$$A_{C+Al} - \alpha_1 * A_{Al} = A_{C247} \quad (1)$$

onde A_{C+Al} corresponde as transições em 193,03 (C I, Al I e Al II) e A_{Al} corresponde transição de Al em 193,58 nm. O fator α_1 pode ser obtido através de um ajuste linear que maximize o coeficiente de Pearson obtido na criação do modelo de calibração. A Figura 9 mostra um gráfico de correlação entre o C total obtido com o CHNS e com o LIBS após determinação do C pelo método analítico proposto por Nicolodelli, e assim obteve-se um valor corrigido para o pico de C em 193,03 nm. A correlação foi de 91% ($R=0,91$), corroborando com o fato de essa ser uma metodologia interessante para a análise de C em solos brasileiros.

Figura 9 – Modelo de calibração para quantificação de C utilizando LIBS obtido por NICOLODELLI para solos brasileiros



Fonte: NICOLODELLI (Adaptado).

CAPÍTULO 3 – ESTADO DA ARTE DE EQUIPAMENTOS LIBS PORTÁTEIS

Desde a sua concepção, o LIBS já teve um foco comercial uma vez que ele é o resultado de iniciativas entre Universidade-Indústria. Atualmente existe uma profusão de empresas que vendem sistemas LIBS completos capazes de analisar diferentes tipos de materiais. No entanto a grande maioria são sistemas caros e grandes, e precisam ser instalados em locais com infraestrutura adequada para não danificar o equipamento. Apesar disso, existem algumas iniciativas em desenvolver equipamentos portáteis, cada um com um foco próprio. Neste capítulo, selecionamos alguns exemplos de equipamentos LIBS portáteis existentes.

3.1.LIBSCAN 25+ (Applied Photonics)

O LIBSCAN 25+ (Figura 10) é um LIBS portátil fabricado pela *Applied Photonics* que adota uma proposta modular. O equipamento foi desenvolvido tanto para uso em laboratório quanto para aplicações em campo, para tal, utilizando uma bateria de 12 VDC de íon de lítio, permitindo seu uso por até 4 horas. O tamanho da cabeça é de 340 mm de largura, 100 diminuindo até 67 mm do corpo e 140 mm do pegador, pesando 2kg. Já o console, tem dimensões de 500 x 320 x 160 mm e pesa 12kg.

Figura 10 – Equipamento LIBSCAN25+ da Applied Photonics (cabeça e console)

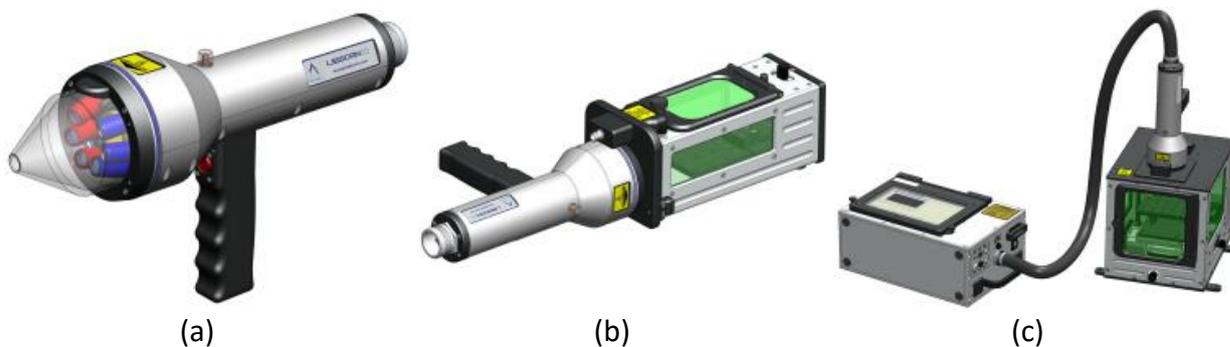


Fonte: http://www.appliedphotonics.co.uk/products_services/libscan_25.htm

É composto por um laser de um Q-switched laser de Nd:YAG operando a 1064 nm, energia entre 40 e 50 mJ e pulso de duração entre 4 e 6 ns com taxa de repetição de 1 Hz. Possui até seis canais, o que possibilita a instalação de até seis espectrômetros compactos podendo detectar emissões entre 185-900 nm, com resolução entre 0,06-0,2 nm, dependendo do espectrômetro instalado. O módulo principal, a cabeça, pode ser acoplada em uma grande

quantidade de módulos adicionais, incluindo câmaras para medição, como pode ser visto na Figura 11.

Figura 11 – Cabeça do LIBSCAN25+ (a) e módulos adicionais de câmaras de ablação (b-c)



Fonte: http://www.appliedphotonics.co.uk/products_services/libscan_25.htm

Entre as aplicações do equipamento estão a análise de resíduos de tiros por arma de fogo (HONDROGIANNIS, 2013), explosivos caseiros, amostras de nitrato de amônio e clorato de sódio (MIZIOLEK, 2012) e análise em tumbas antigas (GROZEVA, 2012).

3.2. QUANTOM (Bertin Technologies)

É um equipamento desenvolvido pela *Bertin Technologies* para detecção de elementos em polímeros plásticos, voltado para destinação apropriada e reciclagem (Figura 12).

Figura 12 – QUANTOM da Bertin Technologies efetuando uma medição em carcaça de TV CRT



Fonte: <http://www.bertin-technologies.com/quantom.aspx>

Suas dimensões são 760 x 580 x 350 mm (caixa) e 810 x 260 x 130 mm (sonda). A caixa pesa 47 kg e a sonda pesa 12 kg. O fabricante não especifica informações

sobre os lasers e espectrômetros, apenas o equipamento já é calibrado para detector Sb, P, Ca, Si, Mg e Al para as amostras de plástico.

3.3. EasyLIBS (Ivea Solution)

Desenvolvido pela *Ivea Solution*, o EasyLIBS (Figura 13) consiste em um laser pulsado de Nd:YAG de 1024 nm, energia inferior a 25 mJ e 1 Hz de taxa de repetição. Pode ter instalado 3 espectrômetros, cobrindo a faixa espectral de 190-950 nm e com resolução de 0,2 nm. O equipamento pesa 7 kg.

Figura 13 – EasyLIBS da Ivea Solution, com um conjunto de 3 espectrômetros



Fonte: http://www.ivea-solution.com/wp-content/uploads/2016/08/EasyLIBS_EN.compressed.pdf

O equipamento já vem calibrado para a detecção de H, Li, Be, Na, Mg, K, Ca e Al em alguns tipos de amostras, incluindo solos contaminados, detecção de agentes tóxicos e análise de ligas de alumínio.

3.4. StellarCASE-LIBS (StellarNet)

Equipamento fabricado pela *StellarNet Inc*, é a evolução de um dos primeiros equipamentos LIBS com foco em portabilidade a chegar ao mercado (o PORTA LIBS). O StellarCASE-LIBS (Figura 14) é composto por um laser pulsado de Nd:YAG operando a 1064 nm, com 1 Hz de taxa de repetição.

Figura 14 – StellarCASE-LIBS da StellarNet



Fonte: <http://www.stellarnet.us/systems/stellarcase-libs-portable-instrument-for-elemental-analysis/specifications-pricing/>

Pode vir equipado com até três espectrômetros com um detector CCD de 2048 pixels, podendo detectar elementos que emitem entre 200-800 nm. Dentre as aplicações, destacam-se o uso para análise de materiais industriais, mineração e prospecção, monitoramento ambiental e análise forense.

3.5.Z-200 (SciAps)

O Z-200 (Figura 15) é uma das soluções mais portáteis existentes atualmente. Diferentemente dos outros modelos já apresentados, ele tem um laser de menor energia 5-6 mJ, 1 ns de duração do pulso e 50 Hz de taxa de repetição.

Figura 15 – LIBS Z-200 da SciAps



Fonte: <https://www.sciaps.com/libb-handheld-laser-analyzers/z-200/>

Possui um espectrômetro capaz de detectar elementos que emitam entre 190-615 nm, exceto H, F, Br, Cl, N, O, S, Rb, Ce and K. É calibrado para detectar vários elementos, incluindo Al, Si, Cr, Mg, Fe, Ni em ligas metálicas e Li em amostras geológicas.

3.6.ChemLite (TSI)

Desenvolvido pela *TSI*, o ChemLite (Figura 16) é um LIBS de mão similar ao Z-200, mas com foco em análise de metais.

Figura 16 – Equipamento ChemLite da TSI



Fonte: <http://www.tsi.com/ChemLite-Portable-LIBS-Metal-Analyzer/>

A fabricante informa muito pouco acerca dos componentes internos do equipamento, entretanto informa que ele consegue detectar metais base como Al, Mg e Ti, e em ligas, Al, Bi, Cr, Co, Cu, Fe, Pb, Li, Mg, Mn, Mo, Ni, Nb, Si, Ta, Sn, Ti, W, V, Zn, e Zr. Possui um limite de detecção de aproximadamente 0,01% e sua bateria permite efetuar centenas de medições.

CAPÍTULO 4 – DESENVOLVIMENTO DO PROJETO DE LIBS PORTÁTIL

Conforme foi descrito no Capítulo 3, existem iniciativas internacionais muito boas no que concerne ao desenvolvimento de equipamentos LIBS portáteis. Entretanto, é reduzido o número delas que tem como foco a análise de C em solos, com a maioria focando na análise de ligas metálicas. Adentrando no território nacional, o número de iniciativas de desenvolvimento de um LIBS portátil cai vertiginosamente, fazendo deste um dos projetos pioneiros na área. Assim, esse trabalho de mestrado originou-se da necessidade de miniaturizar uma técnica já consolidada pela comunidade científica, mas que possibilitasse a sua instalação e manuseio em laboratórios de pouca infraestrutura – no campo, por exemplo. Neste capítulo, faremos uma breve introdução, apresentando equipamentos que serviram de referência para este trabalho. Depois, serão explicitados os procedimentos envolvidos no desenvolvimento do protótipo, incluindo sistema para acionamento do laser, softwares e drivers para a aquisição de sinal, e modelos 3D do equipamento portátil.

4.1. Equipamentos LIBS de referência

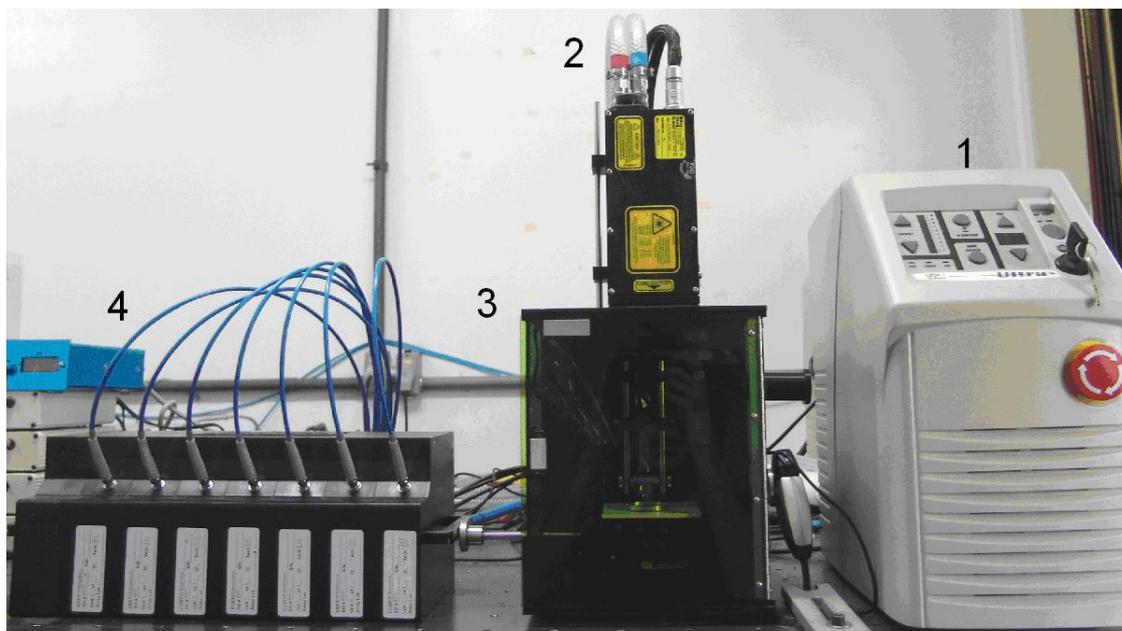
Nosso ponto de partida foi a expertise que o Laboratório de Óptica e Fotônica da Embrapa Instrumentação já tem com técnica. Dois dos equipamentos que o Laboratório possui serviram como base para o desenvolvimento do protótipo portátil: um LIBS de bancada Ocean Optics LIBS2500Plus, e o PORTA LIBS StellarNet.

4.1.1. Ocean Optics LIBS2500Plus

O Laboratório de Óptica e Fotônica da Embrapa Instrumentação em São Carlos – SP possui um equipamento convencional de bancada Ocean Optics LIBS 2500Plus, que serviu como referência para todos os métodos e procedimentos utilizados nesse projeto de mestrado. Esse sistema é composto por um Q-switched laser de Nd:YAG (Quantel, modelo Big Sky Laser Ultra 50) com pulsos de 10 ns de duração e taxa de repetição de até 500 Hz, comprimento de onda de emissão de 1064 nm e energia máxima em torno de 50 mJ. O sistema de detecção é composto por um conjunto de sete mini-espectrômetros de alta resolução (0,1 nm) e com um detector do tipo CCD com 2048 pixels cada, que juntos abrangem toda faixa espectral que vai do ultra-violeta profundo – 189 nm – até o infravermelho médio – 966 nm. Essa faixa espectral permite caracterizar um grande número de espécies atômicas que constituem a amostra, uma vez que todos os elementos da tabela periódica possuem linhas de emissão nessa faixa espectral. O intervalo de tempo entre o pulso

do laser e a aquisição do sinal (tempo de atraso) do sistema é fixo em 2 μ s e todas as medidas efetuadas com o nosso sistema LIBS foram realizadas a atmosfera ambiente. A Figura 17 apresenta uma foto desse sistema LIBS, onde maiores detalhes encontram-se na legenda.

Figura 17 – Sistema LIBS de bancada modelo LIBS2500Plus: (1) fonte de energia do laser; (2) laser; (3) câmara de ablação; (4) conjunto de espectrômetros



Fonte: DA SILVA

Esse equipamento vem com um conjunto de softwares próprios para controle de medição e aquisição do espectro denominado OOILIBS. Os espectros medidos são salvos em um arquivo de texto como o ilustrado na Figura 18, de extensão *.ols. O arquivo de dados é composto por um cabeçalho e por duas colunas de dados numéricos com 14366 linhas, isto é, que corresponde a leitura feita em cada pixel da CCD. A primeira coluna registra o comprimento de onda e a segunda a intensidade registrada naquele pixel.

Figura 18 – Arquivo resultante da medição de uma amostra no LIBS2500Plus

```

1.ols (~/Desktop/Embrapa/Workspace/spectra/01_19)
File Edit View Search Tools Documents Help
Open Save Undo
1.ols x
00I LIBS Data File
Wednesday Jan 03 2017
Averages 1
QSwitch Delay 50.0us

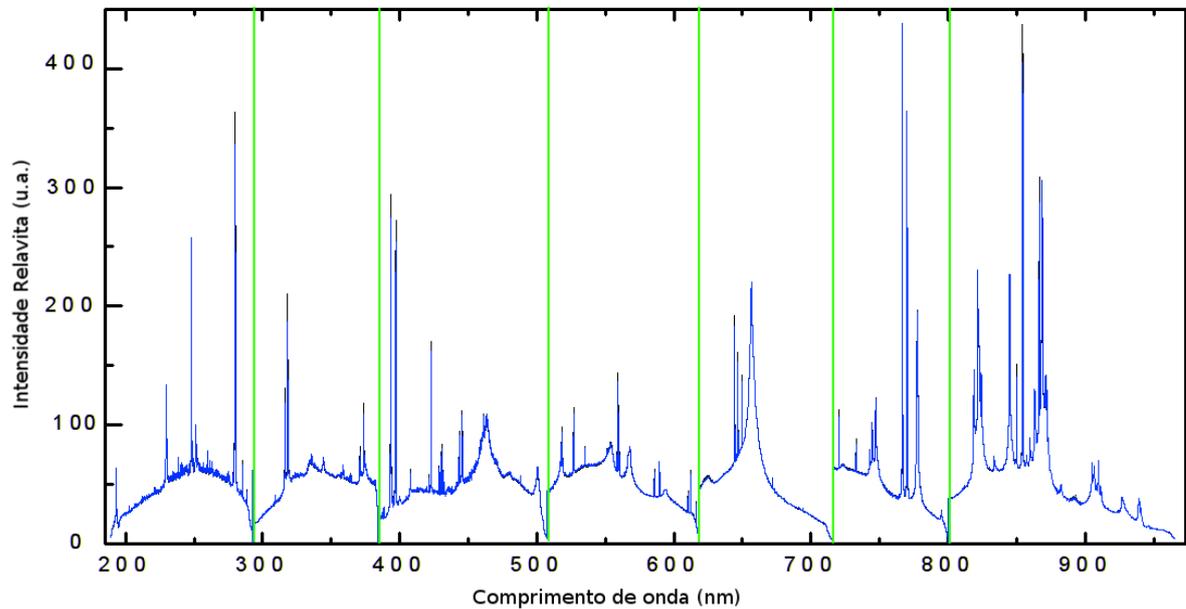
Data
Wavelength Counts
188.83832 0.0000
188.89378 177.0000
188.94923 178.0000
189.00468 178.0000
189.06012 181.0000
189.11556 181.0000
189.17100 182.0000
189.22643 181.0000
189.28186 182.0000
189.33728 179.0000
189.39269 181.0000
189.44810 180.0000
189.50351 179.0000
189.55891 181.0000
189.61431 179.0000
189.66970 182.0000
189.72509 183.0000
Saving file '/home/k/Desktop/Embrapa/Worksp... Plain Text Tab Width: 4 Ln 6, Col 5 INS

```

Fonte: Autoria própria

A partir dos dados salvos nas duas colunas, *Wavelength* e *Counts* é possível traçar um gráfico e visualizar o espectro característico como na Figura 19. Como o LIBS2500Plus é composto por sete espectrômetros, isso gera regiões de descontinuidade na interface de modo que é possível observar a separação entre as regiões espectrais marcadas por linhas verdes na Figura 19. Os intervalos para as regiões espectrais de cada espectrômetro do conjunto estão descritos na Tabela 4.

Figura 19 – Espectro característico obtido com o LIBS2500Plus separado por regiões espectrais



Fonte: Autoria própria

Tabela 4 – Intervalo espectral referente a cada um dos 7 mini-espectrômetros do equipamento LIBS2500Plus

Espectrômetro	Região espectral (nm)
1	188,8 – 292,6
2	292,6 – 384,4
3	384,4 – 507,1
4	507,1 – 618,3
5	618,3 – 716,1
6	716,1 – 800,2
7	800,2 – 966,2

Fonte: Autoria própria.

A partir do espectro obtido na medição já pode ser feita uma análise qualitativa. Esta aborda desde a identificação das linhas de emissão e composição elementar da amostra analisada até a separação de amostras com concentrações ou características diferentes. A etapa de identificação das linhas espectrais envolve inúmeras tomadas de decisão e grande conhecimento e experiência (RANULFI, 2014A; RANULFI, 2014B).

Durante a identificação, é importante checar a presença do que se denominam linhas persistentes no espectro, que são as linhas que continuam presentes no espectro de emissão mesmo a baixas concentrações do elemento em questão e consultas a tabelas e bases de dados. A ferramenta online mais consultada para a identificação das linhas espectrais é a disponibilizada pelo *National Institute of Standards and Technology* (NIST), o “*NIST Atomic Spectra Database Lines Form*”¹. Na sua página principal (Figura 20) é possível inserir o elemento de busca, limites inferiores e superiores do comprimento de onda, e outras características, como probabilidade de transição por exemplo. Após a busca, ele retorna uma lista com as emissões que correspondem à pesquisa (Figura 21).

Figura 20 – Página inicial do NIST Atomic Spectra Database Lines Form

ASD DATA INFORMATION
 LINES LEVELS List of Spectra GROUND STATES & IONIZATION ENERGIES Bibliography Help

NIST
 National Institute of Standards and Technology
 Physical Meas. Laboratory

NIST Atomic Spectra Database Lines Form

Best viewed with the latest versions of Web browsers and JavaScript enabled

Spectrum e.g., Fe I or Na; Mg; Al or mg i-iii

Lower Wavelength: or Upper Wavenumber (in cm⁻¹)

Upper Wavelength: or Lower Wavenumber (in cm⁻¹)

Units: nm ▾

Fonte: http://physics.nist.gov/PhysRefData/ASD/lines_form.html (Adaptado).

¹ Pode ser acessado em <http://physics.nist.gov/PhysRefData/ASD/lines_form.html>

Figura 21 – Resultado de uma busca por emissões de C em um intervalo de 190 a 194 nm utilizando o NIST

ASD

DATA

LINES LEVELS

INFORMATION

List of Spectra Ground States & Ionization Energies

Bibliography Help



NIST Atomic Spectra Database Lines Data

[C \(all spectra\): 30 Lines of Data Found](#)

Example of how to reference these results:
 Kramida, A., Ralchenko, Yu., Reader, J., and NIST ASD Team (2015). *NIST Atomic Spectra Database* (ver. 5.3). [Online]. Available: <http://physics.nist.gov/asd> [2017, February 5]. National Institute of Standards and Technology, Gaithersburg, MD.
[BibTex Citation](#) (new window)

Wavelength range: 190 - 194 nm
 Wavelength in: vacuum below 200 nm, air between 200 and 2000 nm, vacuum above 2000 nm
 Highest relative intensity: 1000

Query NIST Bibliographic Databases for
C (new window)

[C Energy Levels](#)

[C Line Wavelengths and Classification](#)

[C Transition Probabilities](#)

Ion	Observed Wavelength Vac (nm)	Ritz Wavelength Vac (nm)	Rel. Int. (?)	A_{ki} (s^{-1})	Acc.	E_i (cm^{-1})	E_k (cm^{-1})	Lower Level Conf., Term, J		Upper Level Conf., Term, J		Type	TP Ref.	Line Ref.
C III		190.6683		5.19e-03	B+	0.00 - 52 447.11		1s ² 2s ²	1S 0	1s ² 2s2p	3P ^o 2	M2	T4308,T3474	
C III		190.8734		1.14e+02	B+	0.00 - 52 390.75		1s ² 2s ²	1S 0	1s ² 2s2p	3P ^o 1		u24,T6067	
C II		191.0212		2.80e+07	B	131 724.37 - 184 074.59		2s ² 3p	2P ^o 1/2	2s ² 6d	2D 3/2		T5109LS	
C II		191.0593		3.36e+07	B	131 735.52 - 184 075.28		2s ² 3p	2P ^o 3/2	2s ² 6d	2D 5/2		T5109LS	
C II		191.0619		5.60e+06	B	131 735.52 - 184 074.59		2s ² 3p	2P ^o 3/2	2s ² 6d	2D 3/2		T5109LS	

Fonte: http://physics.nist.gov/PhysRefData/ASD/lines_form.html (Adaptado).

Para análises quantitativas, é muito importante a calibração do sistema através de medidas prévias de amostras de concentrações variadas e conhecidas para uma maior certeza das análises. As análises quantitativas também são possíveis de serem realizadas com os espectros obtidos com o sistema LIBS uma vez que sua abundância na amostra é proporcional à intensidade do sinal de emissão medido no plasma, ou a área do pico de emissão do elemento de análise. Apesar de complicações como o efeito de matriz, curvas de calibração são levantadas para avaliar quantitativamente a concentração dos elementos que compõe o material.

4.1.2. PORTA LIBS StellarNet

Outro equipamento utilizado como base para o protótipo do LIBS portátil é o PORTA LIBS da StellarNet (versão anterior ao StellarCASE-LIBS apresentado na seção 3.4 deste trabalho). Ele é composto por um laser Kigre MK-367 (Figura 22), de Nd:YAG, pulsado, que opera no comprimento de onda de excitação de 1064 nm, com energia inicial por pulso de 25 mJ, duração do pulso de 4 ns e taxa de repetição de 1 Hz. O consumo desse laser é em torno de 300 mA. O laser é focalizado na amostra utilizando uma lente de sílica fundida biconvexa de 1/2 polegada de diâmetro e 25 mm de distância focal. A energia de disparo pode ser aumentada variando a quantidade de múltiplos pulsos, conforme ilustra o

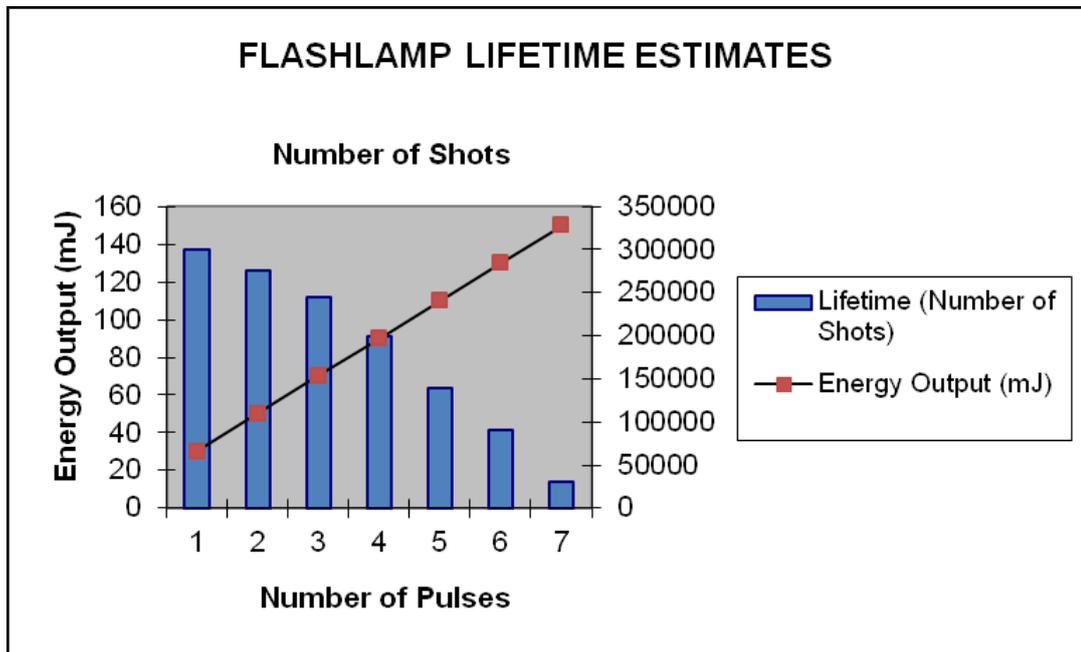
gráfico da Figura 23 retirada do datasheet do laser. Segundo o fabricante, para se obter maiores energias de pulso laser é necessário operá-lo em modo de múltiplos pulsos. No entanto, nesse modo de operação a vida útil do laser reduz inversamente com o número de pulsos acumulados.

Figura 22 – Kit completo do laser Kigre MK-367



Fonte: KIGRE.

Figura 23 – Energia do laser em função do número de pulsos e total de disparos



Fonte: KIGRE

As linhas espectrais são resolvidas no espectrômetro da StellarNet EPP2000-LR (Figura 24), com resolução óptica em torno de 0,2 nm. Este espectrômetro possui um digitalizador integrado de alta velocidade com uma memória de varredura espectral de 4K palavra, oferecendo assim uma captura espectral instantânea de alta sensibilidade com uma CCD de 2048 pixels, uma faixa espectral de detecção de 190 a 400 nm.

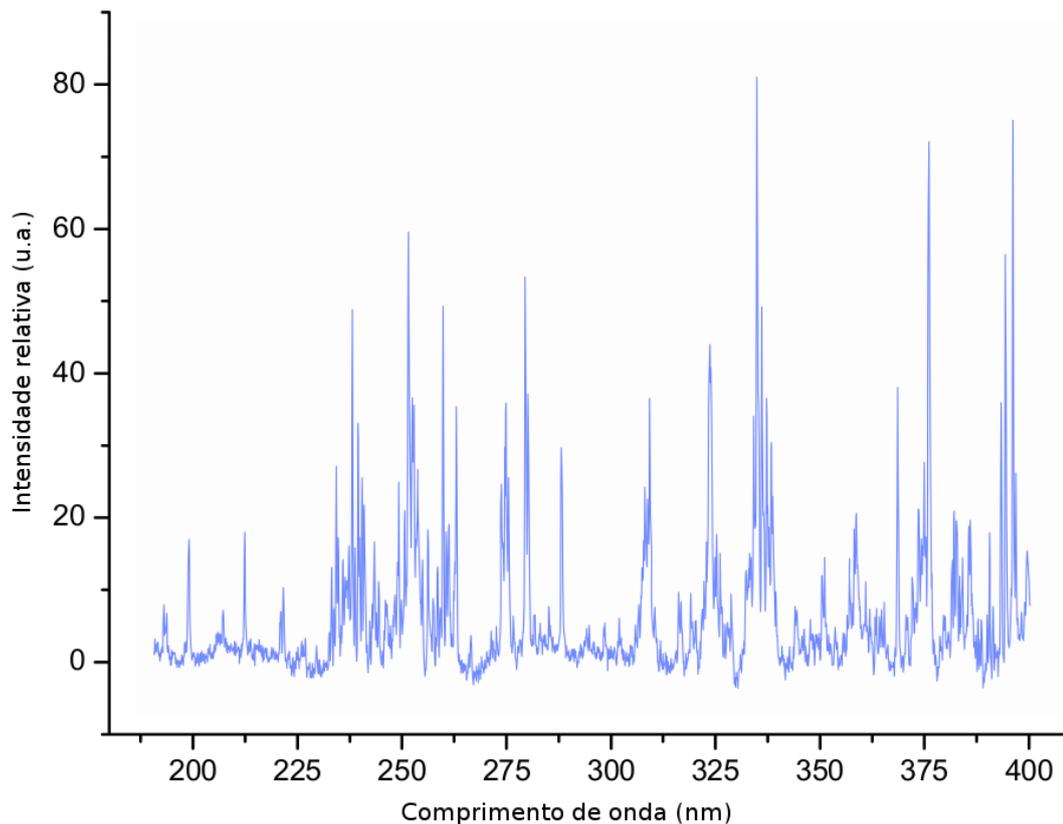
Figura 24 – Espectrômetro StellarNet EPP2000-LR



Fonte: STELLARNET.

A luz do plasma é coletada por uma lente de sílica fundida biconvexa de ½ polegada de diâmetro e 100 mm de distância focal, em um ângulo de 45° do eixo do disparo do laser. A luz coletada pela lente é enviada uma fibra óptica. O controle do espectrômetro com o computador e transferência de dados é feito através de uma porta USB 2.0 (STELLARNET, 2015). A Figura 25 mostra um espectro característico obtido com esse sistema.

Figura 25 – Espectro característico obtido com o LIBS portátil StellarNet



Fonte: DA SILVA (Adaptado).

4.2.Desenvolvimento do projeto de LIBS portátil

Conforme exemplificado na seção 2.2 (Figura 5), um sistema LIBS convencional de bancada possui, além de um arranjo com lentes e fibras ópticas, espectrômetros e laser. Um equipamento comercial como o LIBS2500Plus (seção 4.1.1, Figura 17) é inviável para a portabilização porque possui diversos elementos de grande dimensão, como o conjunto de sete espectrômetros, uma câmera de ablação e um laser bem robusto, que por sua vez necessita de uma fonte também muito robusta, com um sistema de

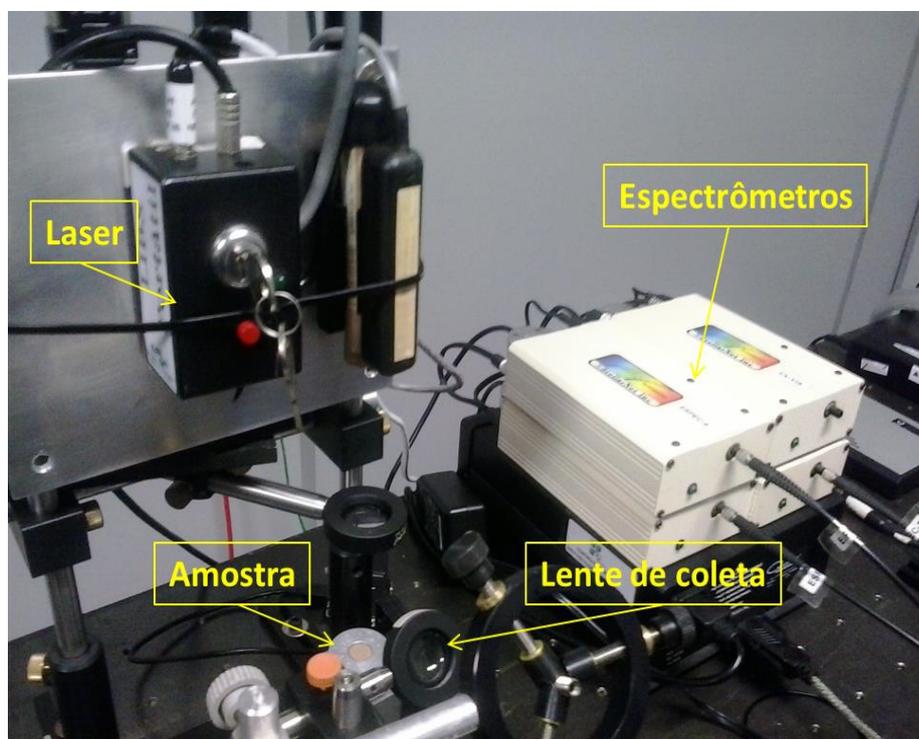
resfriamento bem eficiente. Além de tudo isso, tem um computador (*desktop*) completo no sistema, incluindo teclado, mouse e monitor.

Existem equipamentos menores e com menor demanda energética que podem em conjunto, compor uma versão mais portátil (ou compacta) do LIBS. Assim, o primeiro passo para a portabilidade é a unificação desses equipamentos menores em um espaço reduzido. Para obter tal resultado, foi necessário escolher os dois componentes essenciais: um sistema LIBS (laser, controle de delay, espectrômetros, lentes e fibras) compatível com a portabilidade e um minicomputador (SILVA, 2014).

4.2.1. Sistema LIBS

Para esse projeto, foi utilizado o LIBS portátil descrito na seção 4.1.2, que contém o laser Kigre MK-367 e espectrômetro StellarNet EPP2000-LR. Usualmente, esse equipamento é utilizado com um conjunto de espectrômetros, e é montado ou em uma maleta (para fácil transporte), ou em bancada (como pode ser visto na Figura 26).

Figura 26 – PORTA LIBS StellarNet montado em bancada



Fonte: Autoria própria.

Ainda assim, essa configuração depende de um item essencial: um computador completo, *desktop* ou *notebook*, o que dificulta a portabilização. No entanto, se as funções a

serem executadas no equipamento portátil basicamente são de controle de disparo e análise de espectros, um computador completo não é estritamente necessário. Como buscamos uma solução mais compacta, focando na portabilidade, completa, optou-se por utilizar um minicomputador.

4.2.2. Minicomputador Raspberry Pi

A escolha do minicomputador é de extrema importância para o desenvolvimento do equipamento, já que a essa escolha fica atrelada o potencial computacional, velocidade com a qual as medidas são feitas, e sistema operacional utilizado. Dentre as muitas possibilidades existentes, optou-se por utilizar uma placa de desenvolvimento Raspberry Pi².

O Raspberry Pi (também abreviado como RPi) é um computador de baixo custo (atualmente entre 25 a 45 dólares) com a capacidade de rodar um sistema operacional Linux. Ele foi concebido com a ideia básica de ser um computador de ensino, voltado à educação de crianças, e foi desenvolvido pela Raspberry Pi Foundation, que é uma instituição de caridade no Reino Unido. A instituição promove o estudo de ciências da computação, especialmente em nível das escolas básicas, buscando trazer elementos lúdicos ao aprendizado. Posteriormente, o RPi acabou por possuir inúmeras aplicações ao redor do mundo. Hoje o RPi já está na sua terceira versão, sendo fabricado e vendido em parceria com uma grandes distribuidoras industriais, a Premier Farnell/Element 14, a RS Components, e o distribuidor chinês Egoman Technology. Também, todos os parceiros que fabricam o RPi fornecem áreas de discussão sobre o equipamento, como fóruns de discussão, artigos ensinando como fazer diversas aplicações, além de grande acervo de informações técnicas.

Para esse projeto, utilizamos o Raspberry Pi 2 (Figura 27), por já suprir todas as necessidades para bom funcionamento do protótipo. É de tamanho pequeno, com as dimensões aproximadas de um cartão de crédito, necessita de uma alimentação de apenas 5 V e possui consumo de aproximadamente 1 W.

² Tutoriais simples de introdução podem ser obtidos em <<http://www.raspberrypi.org/quick-start-guide>>

Possuindo um “sistema em um chip”³ (SoC) Broadcom BCM2836, com processador de quatro núcleos rodando à 900 MHz e uma GPU Videocore 4. Esta GPU é capaz de gerenciar exibição de vídeo com qualidade de BluRay (1080p), usando um codec H.264 a 40MBits/s. Também, tem rápido núcleo 3D usando bibliotecas OpenGL ES2.0 e OpenVG. É munido de 1Gb de RAM, quatro portas USB, entrada para módulo de câmera e saída de vídeo composto e HDMI. Para controle de equipamentos, possui 40 pinos GPIO (*General Purpose Input Output*).

Figura 27 - O Raspberry Pi 2



Fonte: <https://www.sparkfun.com/products/13297>

4.2.3. Linux

O Linux é um sistema operacional para computadores extremamente robusto, similar aos Microsoft Windows e Apple OSX. Porém, diferentemente desses, que necessitam um conjunto específico de hardwares para operarem, os sistemas operacionais Linux podem ser obtidos em diferentes variedades e configurações. Isso implica que, um usuário pode encontrar uma distribuição (versão do Linux) que melhor se adapte ao equipamento que ele possui.

Como o Raspberry Pi foi desenvolvido como um projeto de código aberto, ele dá preferência ao uso de sistemas operacionais livres. Como a maioria das distribuições Linux

³ Sistema em um chip (ou system on chip, SoC) é um circuito integrado que une todos os componentes de um computador ou outro sistema eletrônico em um único chip.

são gratuitas, existe maior facilidade para desenvolver aplicações para o RPi uma vez que não há necessidade de adquirir licenças para uso do seu sistema operacional. Desde a versão 2, entretanto, o RPi ganhou uma distribuição especial do Microsoft Windows 10, mas o mesmo possui uma comunidade ativa bem reduzida quando comparado com a comunidade Linux.

Embora muitas pessoas com acesso à tecnologia não façam o uso de Linux em seus PCs, com certeza essas pessoas já o utilizaram sem nem mesmo saber: ele é utilizado como sistema em caixas automáticos de bancos, tevês, leitores de livros eletrônicos etc. O Android⁴, sistema operacional muito popular nos celulares atuais, também é um tipo de Linux. Dentre várias opções disponíveis, a distribuição Linux escolhida para o desenvolvimento do projeto foi a Raspbian⁵.

Raspbian é um sistema operacional gratuito baseado em uma das mais famosas e estáveis distribuições Linux, o Debian⁶. O Debian é um sistema para computadores que inclui um conjunto básico de programas e utilitários necessários para que o computador rode, juntamente com milhares de outros programas. Debian tem a reputação dentro da comunidade Linux de ser de altíssima qualidade, estável, e dimensionável (pode ser executado nos mais variados tipos de equipamentos). Também tem uma extensiva e amigável comunidade que pode ajudar novos usuários, com suporte para praticamente todos os tipos de problemas. Isso torna o Debian uma opção ideal de sistema operacional para o Raspberry Pi, pois seu público abrange variados tipos de usuários, sejam crianças ou adultos.

Raspbian é uma distribuição não oficial customizada do Debian Jessie (existem sempre no mínimo três versões do Debian ativas e em manutenção. Estas são as estáveis, as de teste, e as instáveis. “Jessie” é a atual versão estável), para arquitetura ARM, com uma compilação de configurações ajustadas para utilizar “ponto flutuante por hardware”, fornecida pelo RPi.

4.2.4. Python

O desenvolvimento do protótipo envolve diversas áreas do conhecimento: física para a montagem do sistema óptico, além do entendimento dos fenômenos ocorrendo

⁴ Mais informações em <www.android.com>

⁵ Mais informações em <www.raspbian.org>

⁶ Mais informações em <www.debian.org>

durante a medição; programação, para a escrita dos programas e algoritmos a serem utilizados nas análises; e eletrônica, para se desenvolver circuitos capazes de transmitir a informação entre o computador e os equipamentos. Para auxiliar a resolução desses desafios, é imperativo que a linguagem de programação usada seja extremamente versátil, mas ainda se mantendo fácil de trabalhar. Para tal, escolhemos o Python⁷.

Python é uma linguagem consideravelmente nova, idealizada e criada por Guido van Rossum, na Holanda. Sua primeira versão 0.9.0 é datada de fevereiro de 1991. Sua filosofia enfatiza a legibilidade de código, bem como sua sintaxe elegante permite escrever programas sofisticados com razoável simplicidade, se comparado com linguagens como C++ ou Java. Hoje, o Python está na versão 3.6.0, lançada em 23 de dezembro de 2016. Das vantagens de se trabalhar com Python, podemos citar:

- É open source (gratuito, de código aberto);
- Usa uma sintaxe elegante, fácil de ser lida;
- Pode ser usado em modo interativo (similar ao MATLAB®);
- Suporta orientação a objeto, com classes e herança;
- Possui um mecanismo robusto de detecção e previsão de erros;
- Vem com uma grande variedade de tipos de dados: números (ponto flutuante, complexos e inteiros de grandes ordens), strings, listas e dicionários;
- Seu gerenciamento de memória isenta o usuário de alocar e liberar memória no programa;
- Tem uma quantidade enorme de bibliotecas adicionais, que expandem exponencialmente sua funcionalidade.

Pelo fato do Python ser uma linguagem multiplataforma, ele funciona perfeitamente no RPi. Além disso, o Python é a linguagem oficial do Raspberry Pi, segundo a

⁷ Mais informações em <<http://www.python.org>>

própria Raspberry Foundation: “O Raspberry Pi, por padrão, suporta Python como linguagem educacional.”.

4.2.5. Equipamentos adicionais: monitor e fonte

Definido o laser, espectrômetros e minicomputador para o protótipo, foi possível escolher outros dois equipamentos essenciais ao projeto: um monitor e uma fonte. Para o monitor, idealizou-se um modelo onde se teria uma tela sensível ao toque, na qual o usuário inseriria os parâmetros necessários para a medição de modo simplificado. Também, para fornecer energia para todos os componentes, teríamos uma fonte de computador, pois esta possui várias saídas de tensão consideravelmente limpas e com corrente contínua. Para tal adquiriu-se um monitor de 10,1 polegadas sensível ao toque PlayTix Spider Essence e uma fonte Corsair CX-430, ilustrados nas Figuras 28 e 29.

Figura 28 – Monitor PlayTix Spider Essence 10,1”



Fonte: <https://www.playtix.com.br/loja/monitor-touch-screen-open-frame-10-1-spider-essence.html>

Figura 29 – Fonte de computador Corsair CX430



Fonte: <http://www.corsair.com/pt-br/cx430-80-plus-bronze-certified-power-supply>

4.3.Sistema para acionamento do disparo do laser

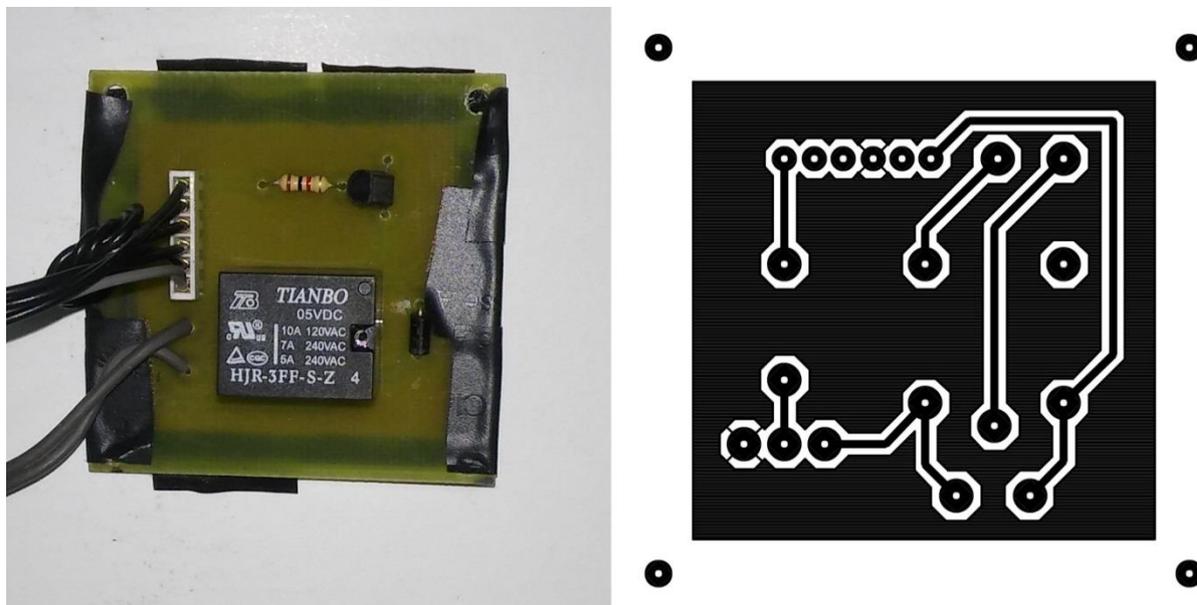
O laser Kigre utilizado nesse projeto vem com uma caixa de controle (Figura 22) com um botão de disparo. E, juntamente com o espectrômetro EPP2000 da StellarNet, já obtivemos um sistema que gera um atraso entre o pulso do laser e o início da aquisição do espectro pelo espectrômetro, não sendo necessário o desenvolvimento de um gerador de delay exclusivamente para essa finalidade. Assim, ao efetuar o disparo do laser pela caixa de controle modificada pela StellarNet tem-se uma sincronia entre o pulso do laser e tempo de integração.

Mesmo funcionando bem dessa forma, em equipamentos comerciais o disparo do laser, e conseqüentemente, a aquisição do sinal pelo espectrômetro é feito inteiramente via software, sem a necessidade de apertar nenhum botão “físico” no equipamento. E esse foi o modelo que se buscou implementar no protótipo do LIBS compacto .

O Raspberry Pi possui diversas portas GPIO, capazes de enviar um sinal digital sincronizado que, em princípio, poderia atuar em um circuito eletrônico, acionando o laser. E, na caixa de disparo e delay existe uma entrada justamente para acionamento remoto via software do laser. Foi montado um circuito bem simples (Figura 30), com um relê que,

quando acionado, fecha um curto, disparando o laser. Tal acionamento pode ser facilmente incorporado dentro de um programa em Python com a biblioteca `python-rpi.gpio`⁸.

Figura 30 – Circuito eletrônico de disparo do laser, final (esquerda) e projeto (direita)



Fonte: Autoria própria.

4.4.Driver Linux para a aquisição de sinal

Ao se adquirir um equipamento LIBS comercial de bancada, o fabricante disponibiliza um programa para que seja feita a medição, bem como a instrumentação necessária para garantir que todos os equipamentos estão sincronizados. Como vimos no tópico anterior, utilizando uma das GPIO do RPi e um pequeno circuito eletrônico, foi possível enviar um sinal digital que por sua vez inicia o disparo do laser. Já, no caso da aquisição de sinal pelo espectrômetro, existe maior grau de dificuldade, pois a fabricante do equipamento disponibiliza gratuitamente apenas o driver para sistemas operacionais Microsoft Windows®.

Entretanto, o pesquisador Björn Rehm desenvolveu um driver próprio para os espectrômetros StellarNet EPP2000-LR mesmo espectrômetro utilizando linguagem C (REHM, 2013), e liberou o mesmo para a comunidade científica. Chamado de *Estrella*⁹, esse driver depois de instalado disponibilizava várias bibliotecas que permitiam o controle do espectrômetro. Mas tal driver possuía linguagem bem anti-intuitiva, toda alteração necessária

⁸ Documentação e maiores informações em <<https://pypi.python.org/pypi/RPi.GPIO>>

⁹ Driver open source para espectrômetros StellarNet. Disponível em <<https://github.com/shugaa/estrella>>

demandava compilar o código do programa de aquisição, e esses executáveis compilados só rodavam na arquitetura específica para a qual foram compiladas (se o programa foi compilado no RPi, funcionava apenas nele; o mesmo se fosse compilado em um desktop ou notebook).

Como grande parte do programa já havia sido escrita em Python, para facilitar o uso do programa para o usuário é imprescindível que tanto o controle de disparo quanto a aquisição se deem dentro do Python. Por ser uma linguagem interpretada (é necessário ter o Python instalado no computador, bem como as bibliotecas utilizadas), e o seu interpretador ser o Cython (escrito em C), faz sentido poder utilizar o driver Estrella e suas bibliotecas compartilhadas open source diretamente no Python. E isso de fato é possível, com a utilização de outra biblioteca, o Ctypes¹⁰.

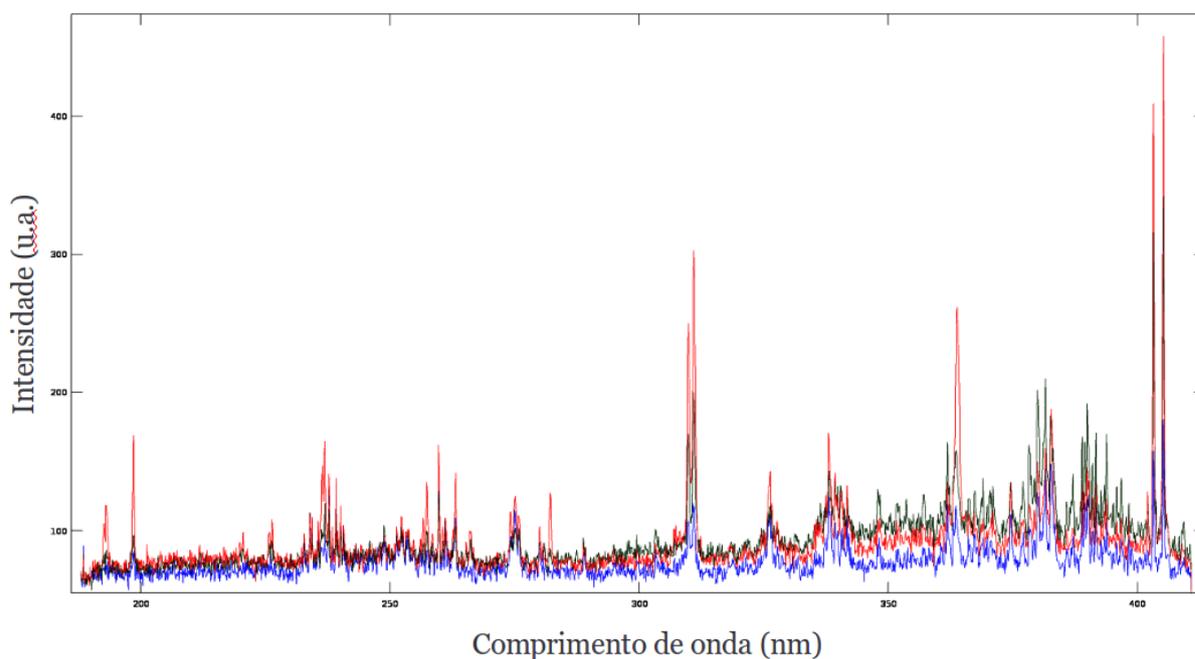
Assim, foi desenvolvido um módulo Python para controle do espectrômetro, usando apenas as bibliotecas do driver em C. Tal módulo em Python também foi disponibilizado para a comunidade junto do projeto original, tornando o autor desse trabalho um colaborador desse driver de código aberto¹¹. O código fonte do driver em Python para os espectrômetros StellarNet EPP2000 está melhor descrito no Apêndice C.

Com o novo módulo, foi possível unir tanto o disparo do laser como a coleta de dados do espectrômetro em um só programa, chamado de **Python Portable LIBS**, compondo uma linguagem unificada. Após um único disparo, o programa recebe um vetor contendo as intensidades relativas para os variados comprimentos de onda. Para disparos consecutivos, ele cria uma matriz onde as intensidades medidas pelo espectrômetro são colunas dessa matriz. Ao final da execução, tem-se um arquivo de texto com uma matriz final, onde a primeira coluna é referente ao comprimento de onda, e as demais são as intensidades para cada disparo. Um espectro de uma amostra de solo obtida com o sistema portátil proposto pode ser visualizado na Figura 31.

¹⁰ Maiores informações em <<https://docs.python.org/3/library/ctypes.html>>

¹¹ Módulo em Python disponível em <https://github.com/shugaa/estrella/tree/master/python_controller>

Figura 31 – Espectro total (4 tiros) de uma amostra de solo obtido com o LIBS portátil proposto no trabalho montado em bancada



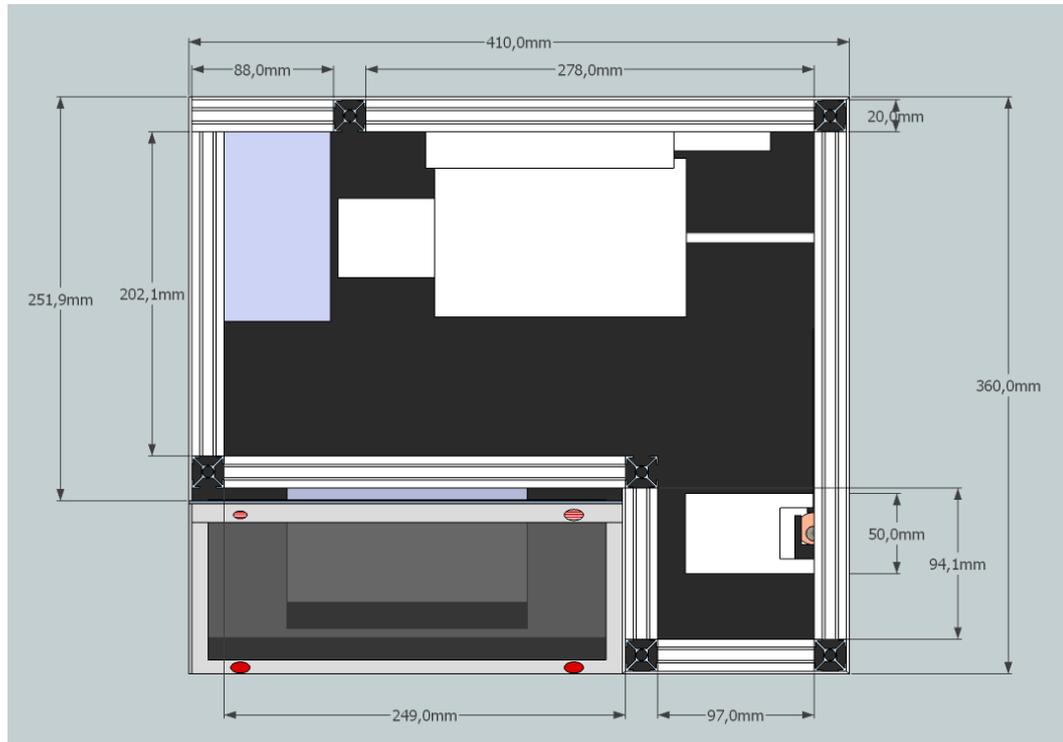
Fonte: Autoria própria.

4.5. Modelos 3D para fabricação do protótipo

De posse da medida de todos os equipamentos que compõem LIBS portátil, foi feito um modelo tridimensional do protótipo utilizando a ferramenta de edição gráfica Google SketchUp 8¹², por ser de uso intuitivo e também gratuita. Nas Figuras 32 a 38 temos algumas das vistas do futuro equipamento.

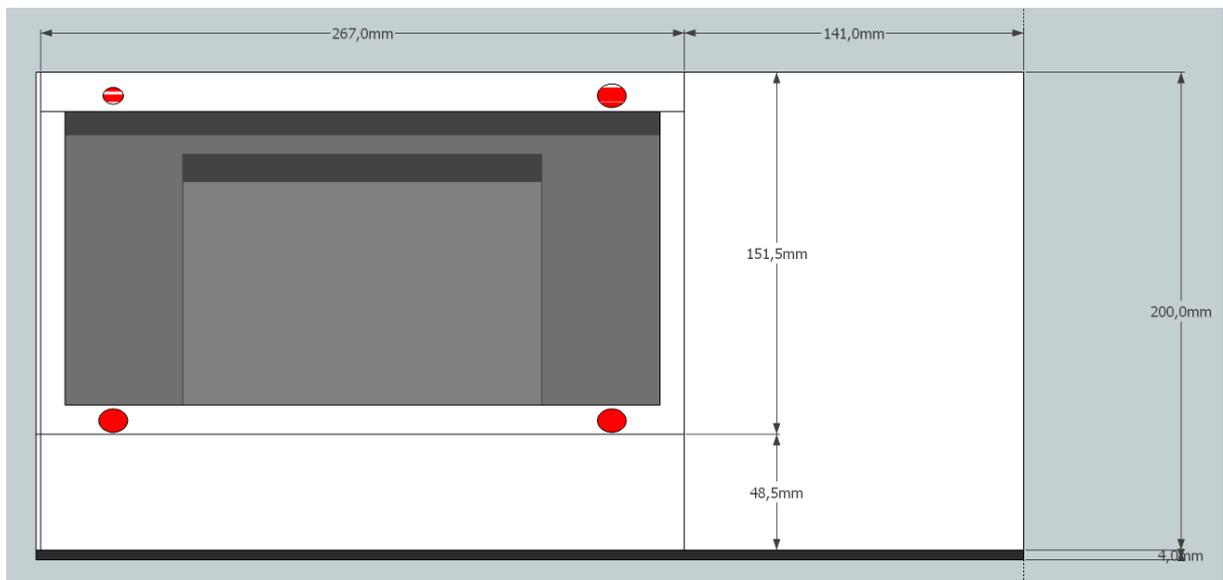
¹² Maiores informações em <<http://www.sketchup.com/pt-BR>>

Figura 32 – Modelo 3D do protótipo: vista superior com abertura



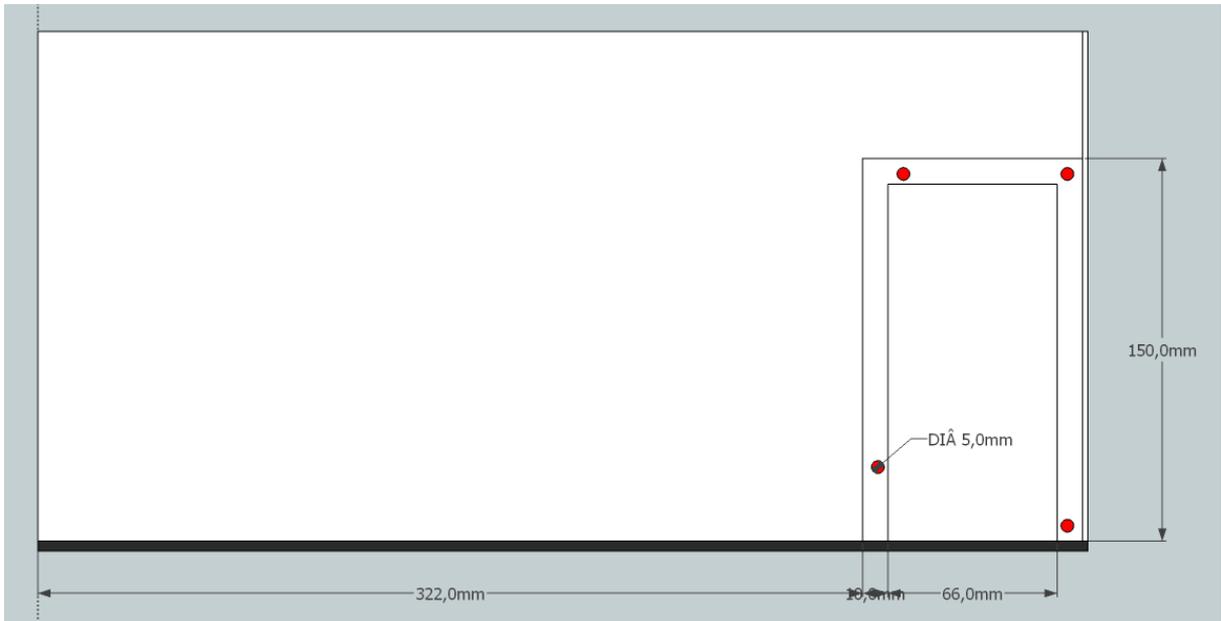
Fonte: Autoria própria.

Figura 33 - Modelo 3D do protótipo: vista frontal



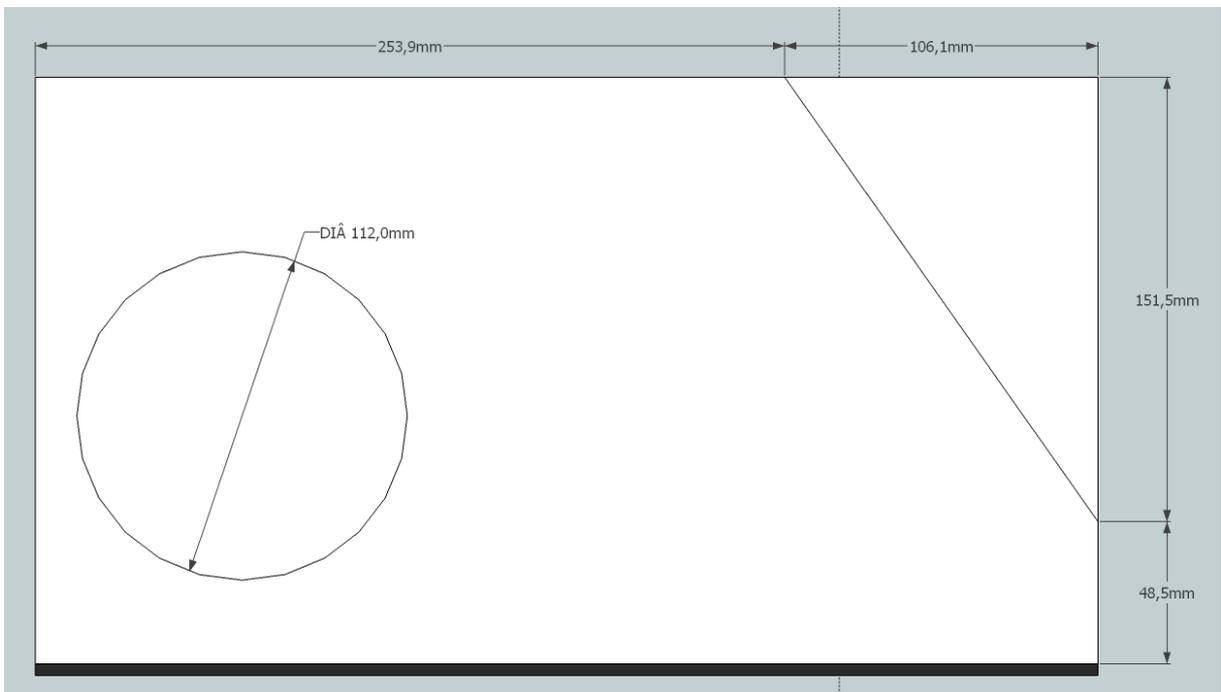
Fonte: Autoria própria.

Figura 34 - Modelo 3D do protótipo: vista trazeira



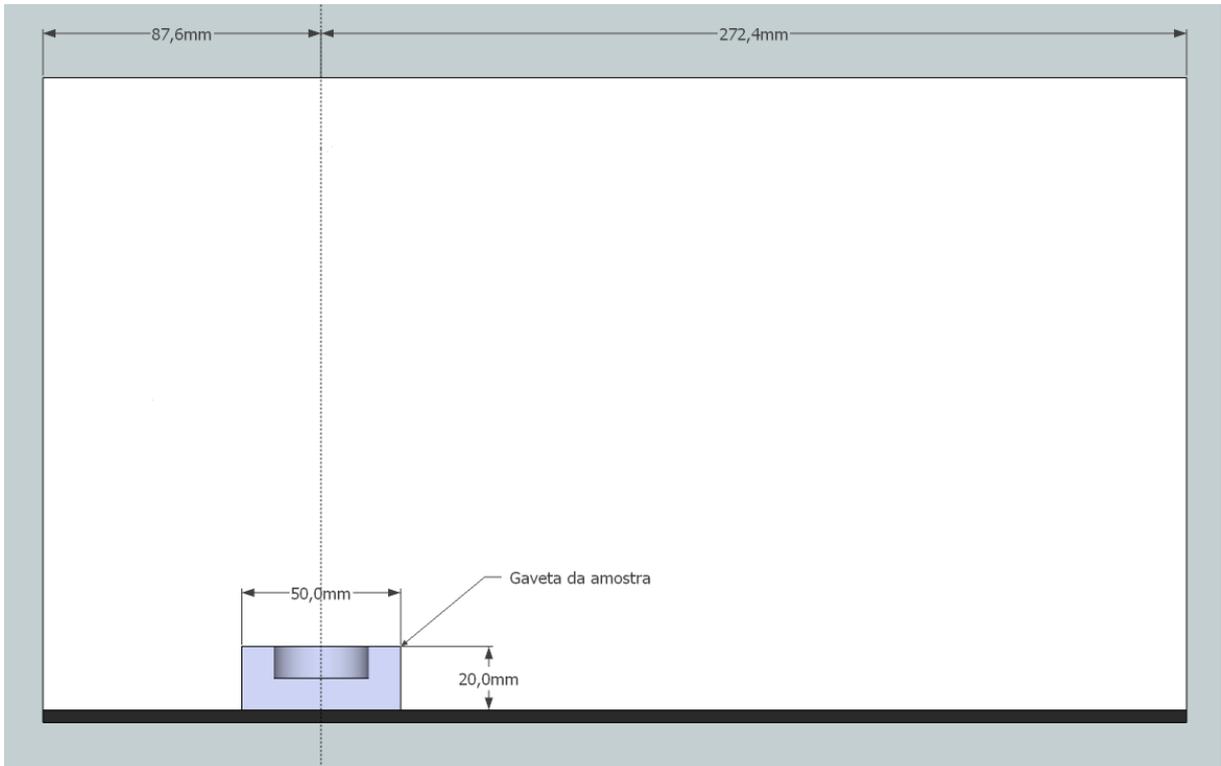
Fonte: Autoria própria.

Figura 35 - Modelo 3D do protótipo: vista direita



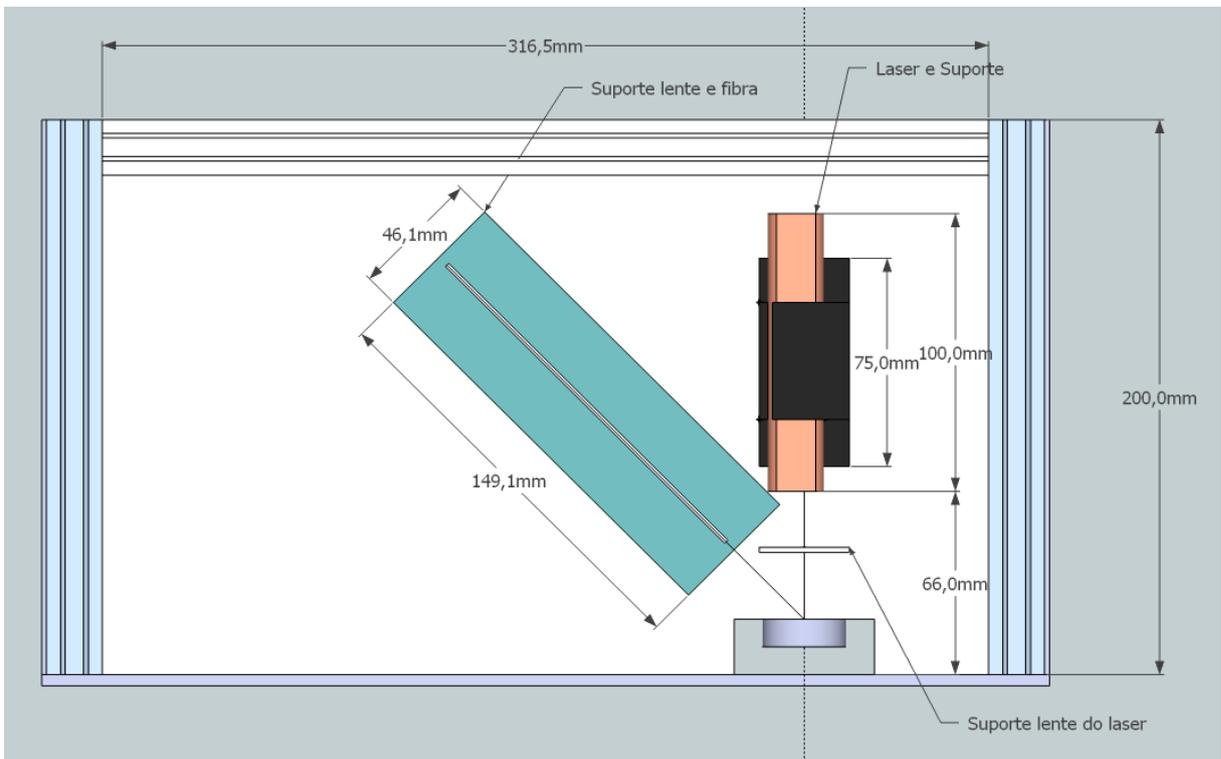
Fonte: Autoria própria.

Figura 36 - Modelo 3D do protótipo: vista esquerda



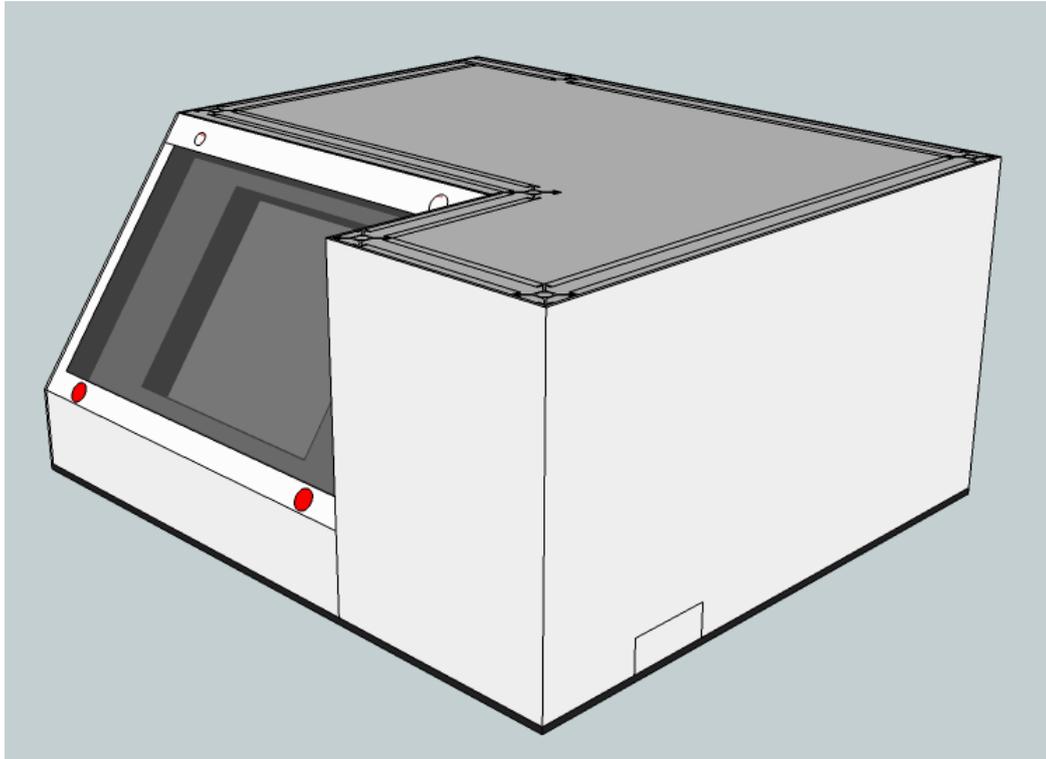
Fonte: Autoria própria.

Figura 37 - Modelo 3D do protótipo: vista interna da parede esquerda



Fonte: Autoria própria.

Figura 38 - Modelo 3D do protótipo: vista ISO



Fonte: Autoria própria.

Este equipamento encontra-se atualmente em viés de concluir o processo de fabricação.

CAPÍTULO 5 – DESENVOLVIMENTO DE SOFTWARE DE ANÁLISE DE SINAIS

Independentemente do sistema LIBS, comercial de bancada ou portátil, a metodologia para analisar o sinal obtido é a mesma, pois ambos tem como resultado final um arquivo de texto com duas colunas, uma referente à intensidade relativa (em unidades arbitrárias) e a outra referente ao comprimento de onda (em nanômetros); e 2048 linhas (igual o número de pixels da CCD de detecção). A diferença entre ambos os sistemas é a qualidade do sinal salvo, posto que o equipamento de bancada possui um laser com mais energia e um conjunto de espectrômetros de maior resolução.

A grande dificuldade é que, os softwares comerciais abrangem apenas a aquisição do sinal. Em alguns casos, o software é capaz de identificar de linhas baseadas em uma base de dados própria, mas por ser muito geral (não conhecendo o tipo de amostra utilizada), seu uso pode resultar em uma identificação errônea das linhas. Isso obriga o operador a exportar o seu espectro para diferentes ferramentas, como OriginLab®¹³ e Microsoft Excel®¹⁴, e fazer a identificação correta dos analitos que contém em sua amostra. Na maioria dos casos, há o interesse de ser associar ao espectro LIBS, resultados quantitativos e nesse caso, cabe ao operador gerar suas próprias curvas de calibração através de ferramentas estatísticas multivariadas. Outra dificuldade é a questão da quantidade de dados para se analisar, já que é usual utilizar um grande número de tiros para se obter uma média representativa da amostra; isso em última instância quer dizer uma grande quantidade de arquivos de texto para abrir, importar dados e fazer o tratamento.

Há, assim, uma grande necessidade em automatizar esse processo de análise, buscando novas metodologias e algoritmos que auxiliem o operador a chegar às suas conclusões mais rapidamente. Isso é particularmente útil também na fabricação de equipamentos LIBS próprios, através do uso de softwares com soluções específicas para o tipo de amostra que se pretende caracterizar, e não apenas um programa mais genérico feito pela fabricante do equipamento.

Pelos motivos já descritos na seção 4.2.4, optou-se por utilizar o Python como linguagem de programação para desenvolvimento do software de análise de sinais. Uma

¹³ Maiores informações disponíveis em <<http://www.originlab.com/>>

¹⁴ Maiores informações disponíveis em <<https://www.office.com/excel>>

vantagem particular do Python para essa parte do trabalho é que ele possui uma ampla gama de bibliotecas científicas que foram essenciais no decorrer do desenvolvimento dos programas. Dentre as bibliotecas adicionais disponíveis, cita-se o NumPy/SciPy¹⁵, Matplotlib¹⁶ e Pandas¹⁷ como fundamentais no desenvolvimento de nossa aplicação. O NumPy e o SciPy oferecem a estrutura de matrizes, vetores, algoritmos matemáticos e rotinas de otimização; Matplotlib é a biblioteca utilizada para traçar os gráficos; Pandas fornece ferramentas especiais de leitura de arquivos de texto, além de possuir também um eficiente mecanismo de indexação.

5.1. Módulos do software de análise de espectros LIBS

Em poucos passos, o processo de análise dos espectros LIBS é composto por:

- Efetuar a leitura dos múltiplos espectros (geralmente da ordem de 100 por amostra) e converter as duas colunas contidas no arquivo de texto resultante em uma matriz onde a primeira coluna seja o comprimento de onda, e as demais as intensidades relativas;
- Efetuar a correção de linha de base, subtraindo o offset eletrônico para cada região espectral (no caso de se usar múltiplos espectrômetros, como no caso do LIBS2500Plus). Nessa etapa, também se utilizou mecanismos para remoção de eventuais medidas fora do padrão (*outliers*);
- O usuário fornece um intervalo espectral acerca do pico do elemento de interesse e o software deve isolar essa região. Dentre as opções, o algoritmo deve corrigir a linha de base em torno do pico (inclinação) e fazer algum tratamento matemático necessário;
- Calcular a área do pico do elemento ou por integração numérica através da regra Trapézio, ou por meio do ajuste da curva do pico, Lorentziano ou Gaussiano, e por fim integrando essa função;
- Correlacionar a área do pico com valores de uma técnica de referência, obtendo-se assim um modelo de calibração.

¹⁵ Mais informações em <<http://www.scipy.org>>

¹⁶ Mais informações em <<http://matplotlib.org>>

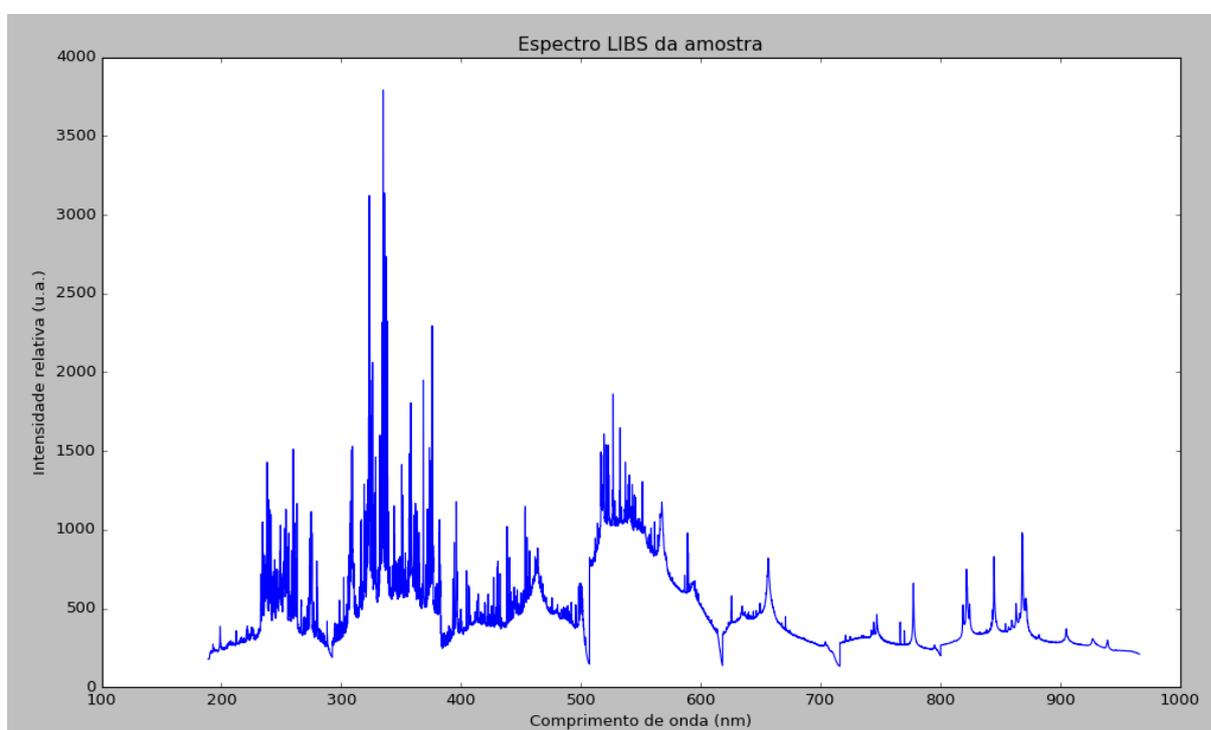
¹⁷ Tutorial explicativo disponível em <<http://manishamde.github.io/blog/2013/03/07/pandas-and-python-top-10>>

Assim, para se obter uma solução completa para análise de espectros LIBS, todos esses passos devem estar incluídos no software de análise.

5.1.1. Módulo de leitura de espectros

Conforme vimos na seção 4.1.1, na Figura 18, o resultado final da medição com um LIBS é um arquivo de texto com uma coluna sendo o comprimento de onda, e outra a intensidade relativa. Ao traçar em um gráfico esses valores, obtém-se um espectro como o da Figura 39.

Figura 39 – Espectro LIBS característico obtido com o equipamento LIBS2500 Plus



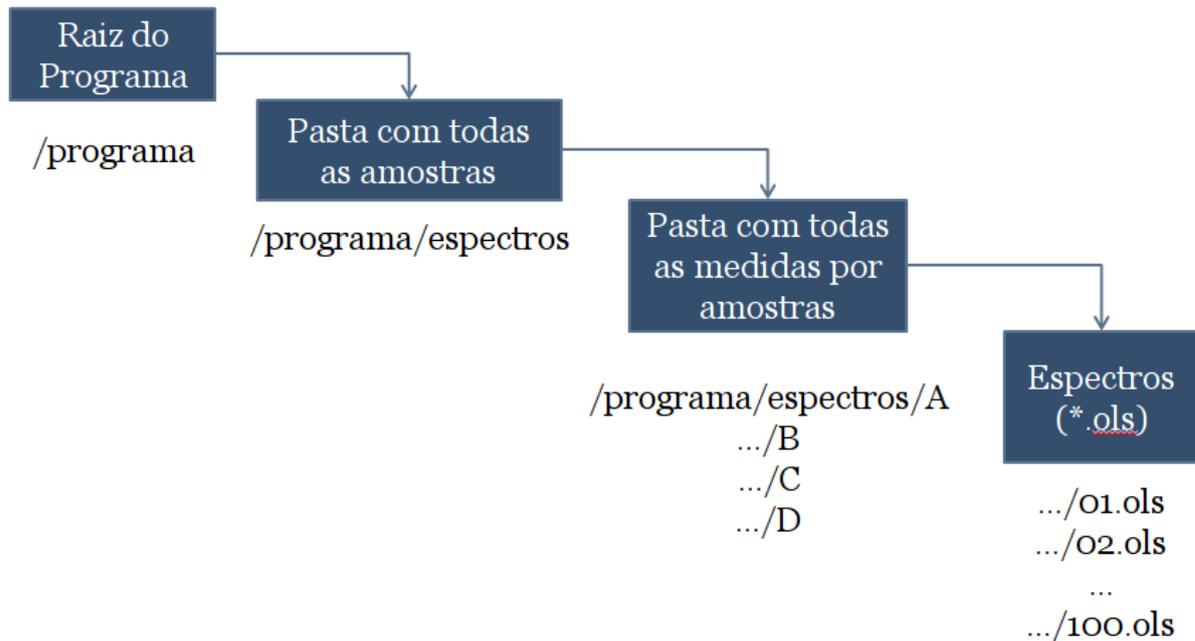
Fonte: Autoria própria.

No entanto, não é usual trabalhar com apenas um espectro por amostra, como já foi citado anteriormente no trabalho. Cada espectro salvo corresponde a um arquivo de texto que precisa ser lido pelo computador, o que deixa o programa mais pesado e todo o processo de análise mais lento. Para tal optou-se em trabalhar com a biblioteca Pandas, pois ela possui algoritmos otimizados para leitura de texto, dentre eles, o `pandas.read_csv`¹⁸ capaz de gerenciar em alta velocidade o acesso desses arquivos.

¹⁸ Maiores informações em <http://pandas.pydata.org/pandas-docs/stable/generated/pandas.read_csv.html>

O programa funciona entrando em uma pasta contendo todas as amostras de interesse. Por sua vez, na pasta de cada amostra estão os arquivos de texto (no caso, *.ols) com os dados de cada espectro. A estrutura de arquivos está esquematizada na Figura 40 abaixo:

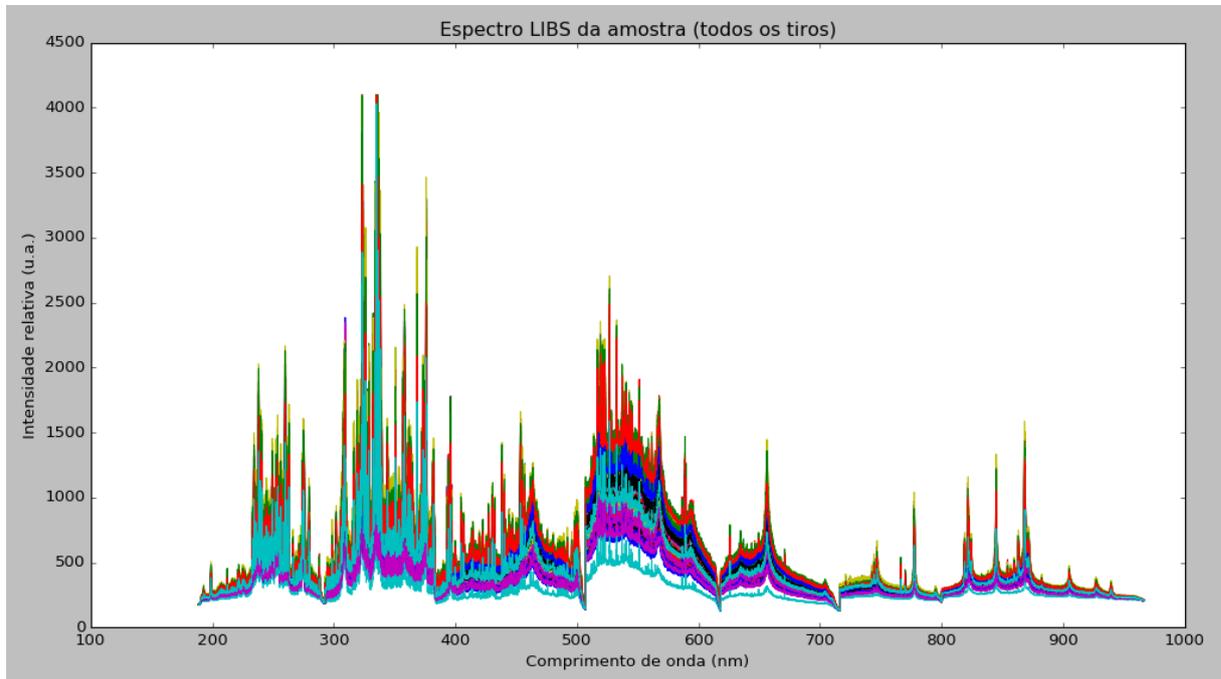
Figura 40 – Esquema da estrutura de pastas no programa desenvolvido



Fonte: Autoria própria.

Utilizando a função `read_csv` da biblioteca Pandas, é lido cada um dos arquivos de espectros individuais, seus valores são armazenados em vetores (*arrays*, da biblioteca Numpy) e guardados. Ao fim, todos os vetores compõem uma matriz, onde a primeira coluna é o comprimento de onda (em nm) e as demais são as intensidades relativas (em u.a.), uma para cada amostra. Executado o programa é possível traçar um novo gráfico com todas as amostras juntas (Figura 41).

Figura 41 – Espectro LIBS total (todos os tiros) de uma determinada amostra obtido com o LIBS2500Plus



Fonte: Autoria própria.

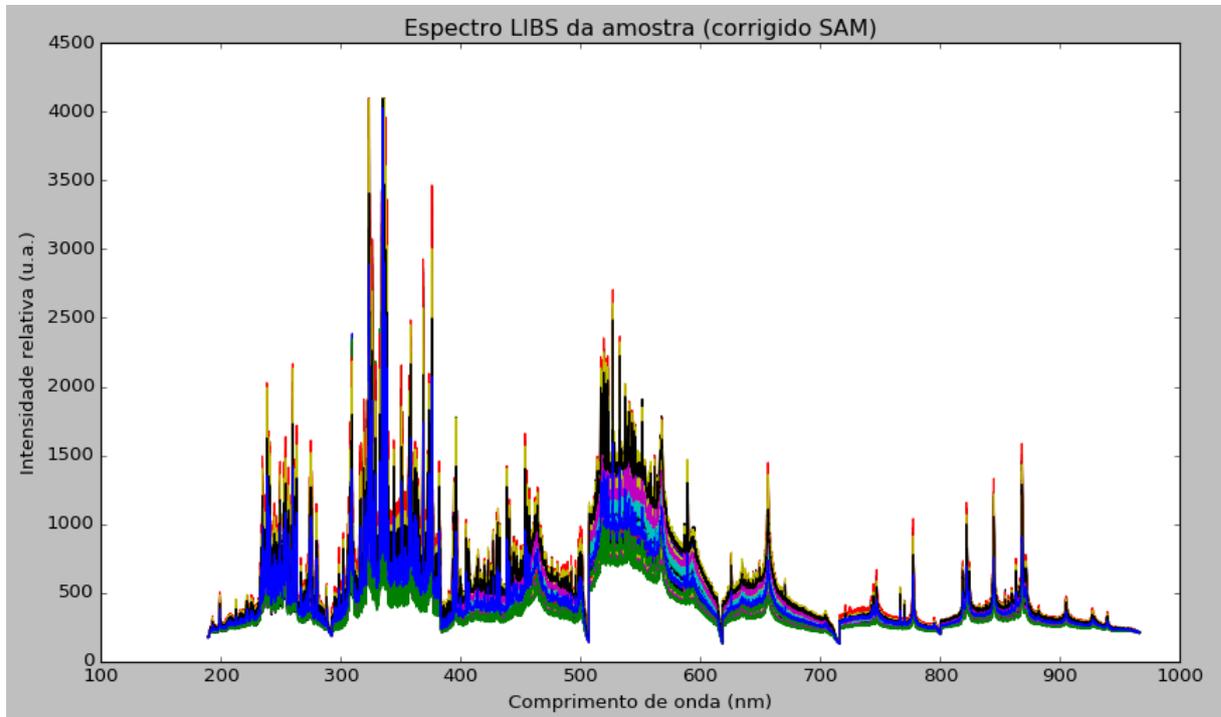
No gráfico da Figura 41 é possível notar que nem todos os espectros possuem a mesma resposta, isto é, os espectros apresentam um grande intervalo de intensidades em torno de um valor médio com alguns poucos espectros completamente fora do grupo. Essas são medidas que por algum motivo ficaram fora do padrão e são chamados *outliers*. Um método muito eficiente encontrar os outliers é o chamado *Spectral Angle Mapper*, ou simplesmente **SAM** (KESHAVA, 2004). O SAM utiliza o produto interno entre uma amostra com a média delas como um indicador do quão fora de fase esse espectro está com o conjunto:

$$\cos(\theta) = (\mathbf{y}_m * \mathbf{y}_i) / [|\mathbf{y}_m| * |\mathbf{y}_i|] \quad (2)$$

A Equação 2 mostra como funciona o SAM. Primeiramente, é tomada a média de todos os espectros (\mathbf{y}_m), e, em seguida, calcula-se o produto escalar de cada espectro (\mathbf{y}_i) com a média. Dividindo o produto escalar pela multiplicação das normas obtemos um resultado que assume valores de no mínimo 0 (completamente ortogonal) e até no máximo 1 (idêntico à média). Um limite de aceitação de $\cos(\theta)$ é definido e, conseqüentemente, um padrão para excluir espectros que não sejam representativos. Um valor de 0,8 exclui medidas muito grosseiras, enquanto que um ajuste fino a partir de 0,9 nos fornece apenas os melhores

espectros obtidos. Para SAM com um valor de 0,995 obteve-se um gráfico como o mostrado na Figura 42.

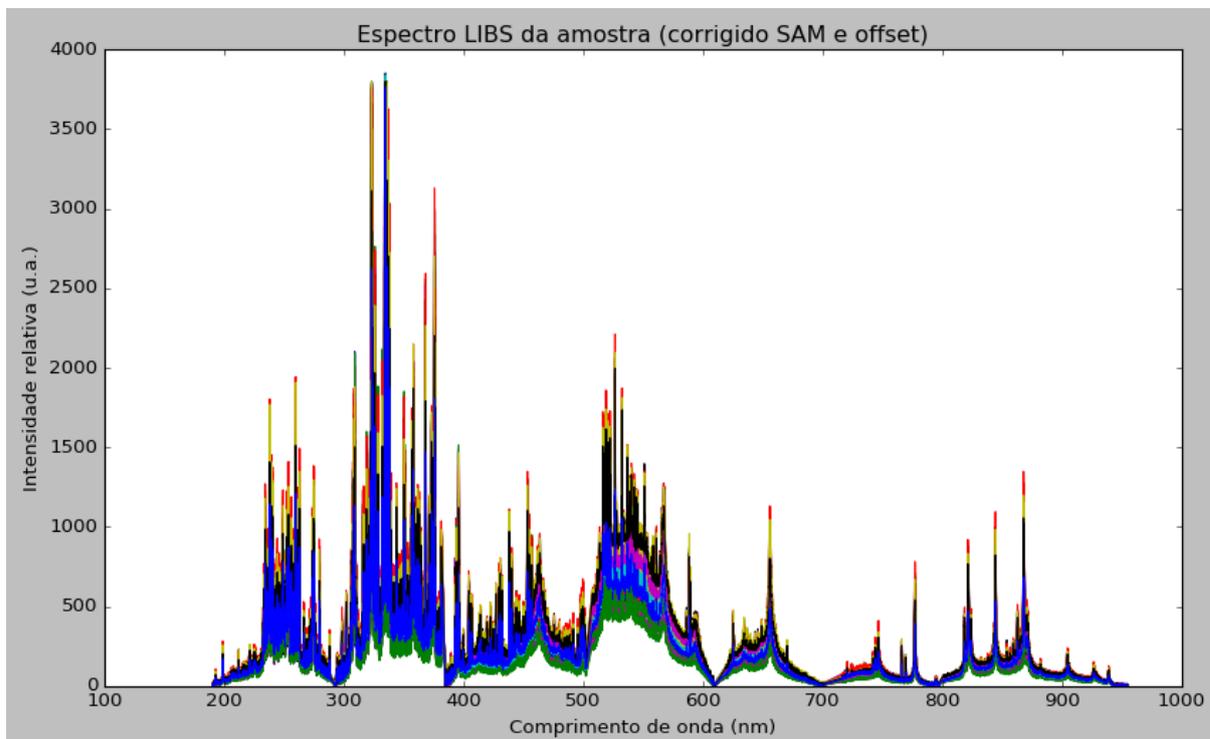
Figura 42 – Espectro total de uma amostra após a remoção de outliers com o SAM



Fonte: Autoria própria.

Como definido na Tabela 4, o espectro LIBS é composto por 7 regiões espectrais, uma para cada espectrômetro, e, como é possível observar pela Figura 42, entre cada uma das regiões há uma descontinuidade. Cada uma dessas regiões tem um offset eletrônico próprio e, para calculá-lo, verifica-se uma sub-região com o menor valor de intensidade relativa em cada espectrômetro e calculada uma média. Essa média é subtraída ponto a ponto dentro de sua região espectral, fazendo com que, no total, o espectro final vá para muito próximo do zero no eixo das intensidades. Nesse ponto, finalmente obtemos um espectro corrigido (Figura 43), pronto para as próximas etapas do programa de análise de dados. Cada espectro final é salvo em uma pasta, que será acessada pelo próximo módulo do programa, de isolamento de picos.

Figura 43 – Espectro final corrigido, pronto para uso em outras etapas da análise



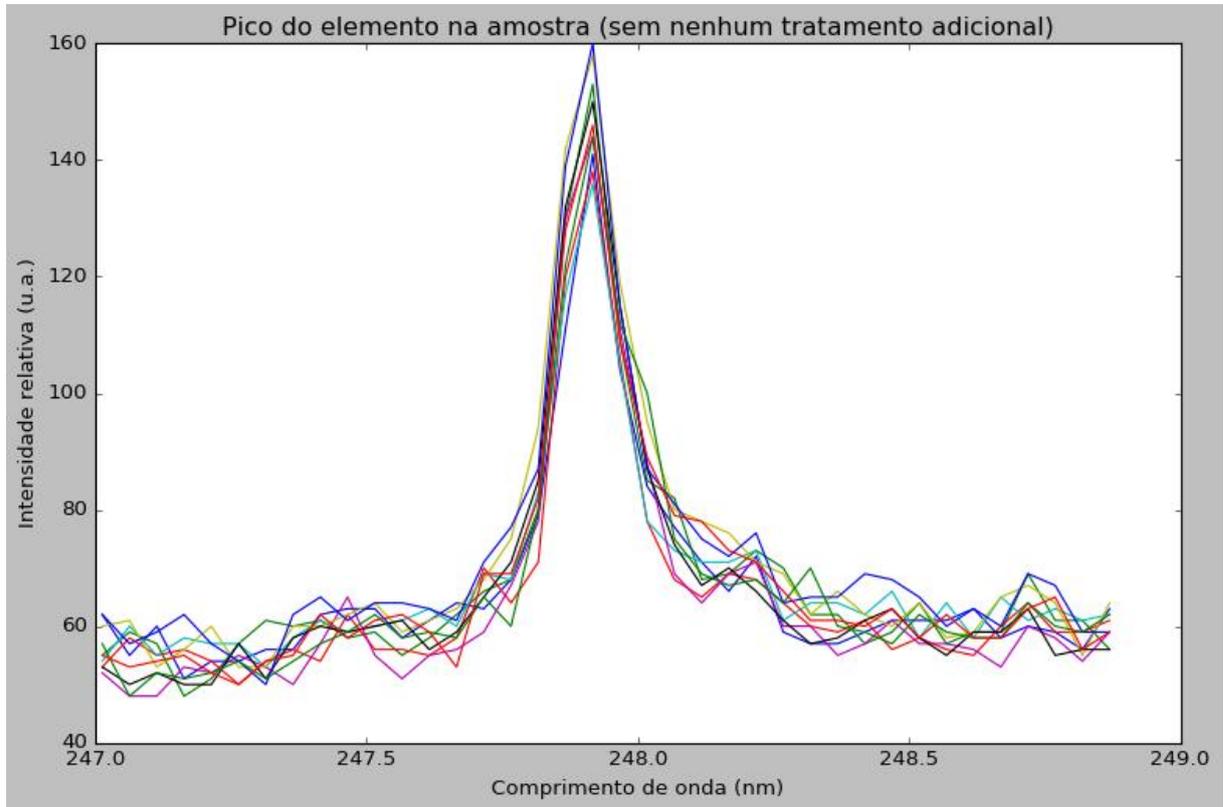
Fonte: Autoria própria.

5.1.2. Módulo de isolamento de elementos (picos)

Nessa etapa, é necessário que o operador saiba qual elemento está buscando e, principalmente, se ele se encontra presente na amostra. Isso geralmente é feito consultando uma base de dados (como o NIST) ou revisando livros, artigos e periódicos que contenham estudos contendo o elemento que se planeja analisar.

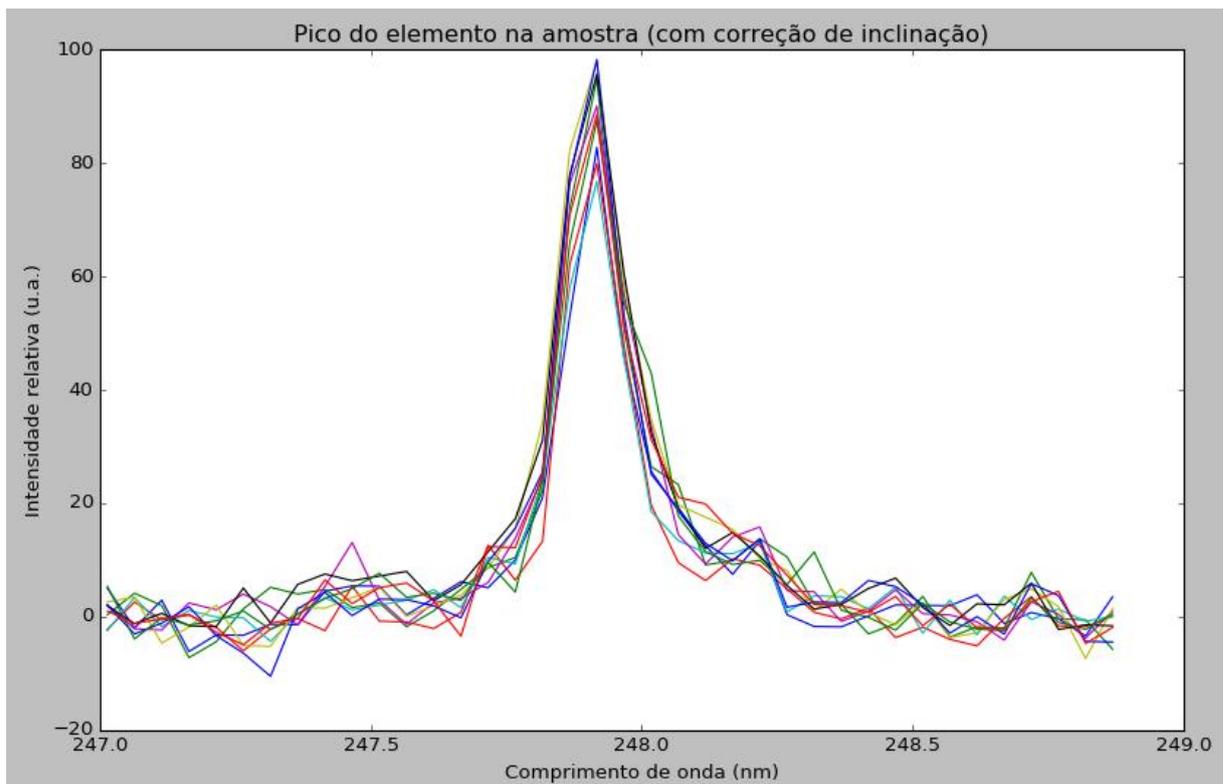
Sabendo que o elemento de interesse encontra-se na amostra, basta colocar os limites superior e inferior do pico de interesse no programa. Ele vai acessar o arquivo salvo pelo módulo de importação, e fazer o “recorte” de apenas aquela área de interesse (Figura 44). Ainda assim, geralmente o pico possui uma leve inclinação. Para corrigi-la é traçada uma reta com início e fim delimitado pelo pico; posteriormente essa reta será subtraída do pico, para cada espectro, obtendo-se assim um pico com correção de inclinação (Figura 45). Opcionalmente pode ser feita uma normalização do pico utilizando a área da reta, procedimento este que pode melhorar a qualidade do sinal obtido (MARANGONI, 2015). Finalmente, é tomada a média desse pico corrigido (Figura 46) e eventualmente esse dado é salvo para uso na próxima etapa do programa, de ajuste do pico.

Figura 44 – Pico de um elemento sem nenhum tratamento



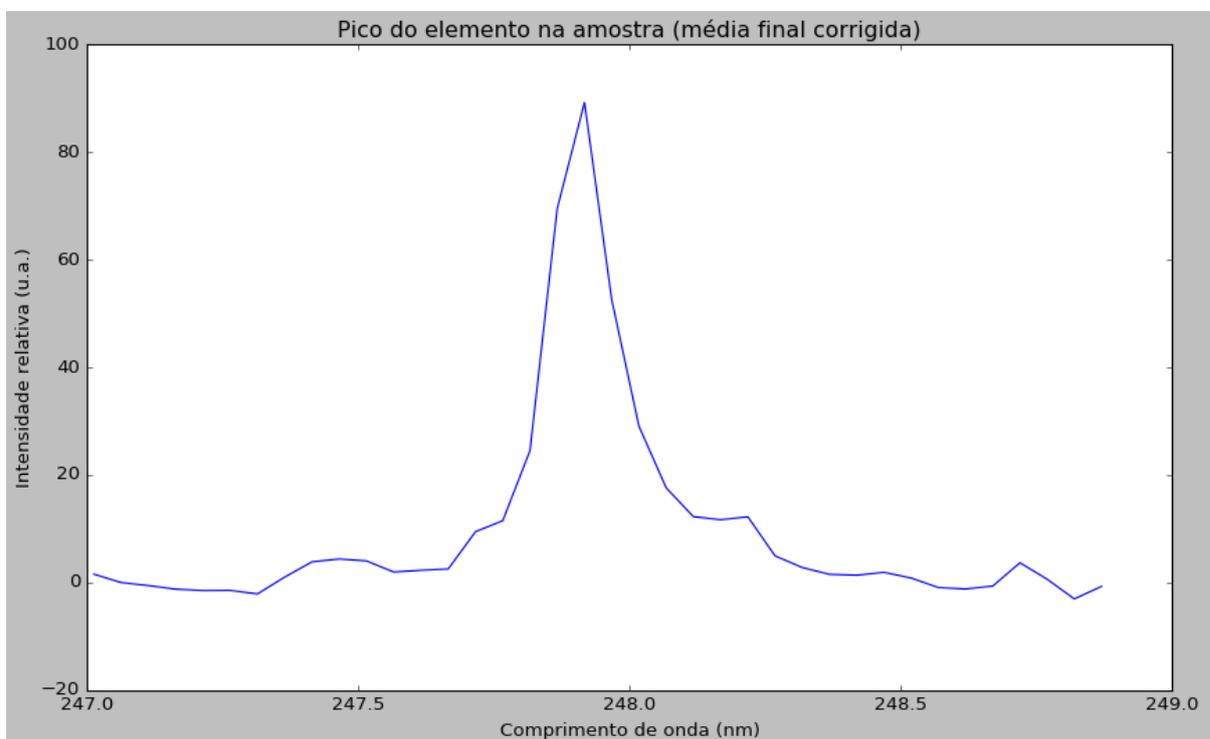
Fonte: Autoria própria.

Figura 45 – Pico de um elemento com correção de inclinação



Fonte: Autoria própria.

Figura 46 – Pico médio de um elemento após os tratamentos

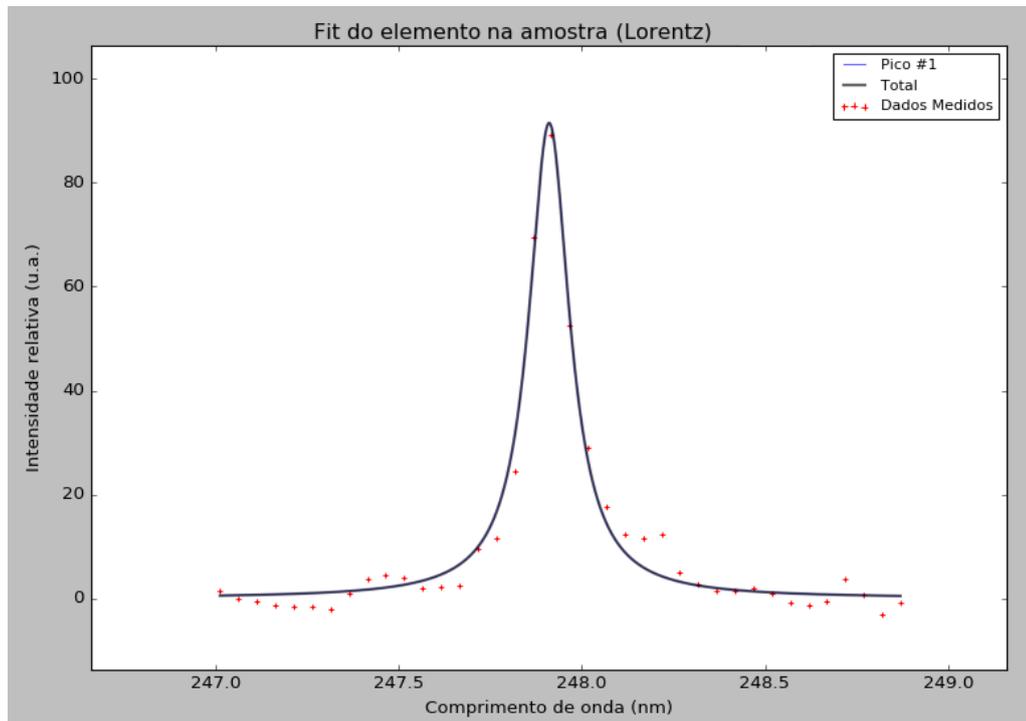


Fonte: Autoria própria.

5.1.3. Módulo de ajuste das curvas dos picos

Agora, o objeto de interesse da análise é a obtenção da área do pico estudado. Lendo o pico médio já corrigido, é possível utilizar a regra Trapezoidal e obter instantaneamente um valor da área. Esse método não é o mais preciso, mas utiliza pouco poder computacional e executa rapidamente os cálculos necessários. Outra opção é fazer o ajuste da curva do pico em questão utilizando uma função, geralmente Lorentziana (quando o evento principal na formação do pico é o Efeito Stark) ou Gaussiana (caso seja Efeito Doppler). Com o ajuste pronto (Figura 47), basta integrar a curva e obter o valor da área da curva. O programa salva um relatório em formato de planilha (Figura 48) com as áreas de cada pico e outras informações adicionais, utilizando a biblioteca Pandas. Esses valores são utilizados no último módulo, para criação de modelo.

Figura 47 – Fit do pico utilizando uma curva Lorentziana



Fonte: Autoria própria.

Figura 48 – Relatório com os valores das áreas calculadas para cada amostra analisada com o módulo de fit

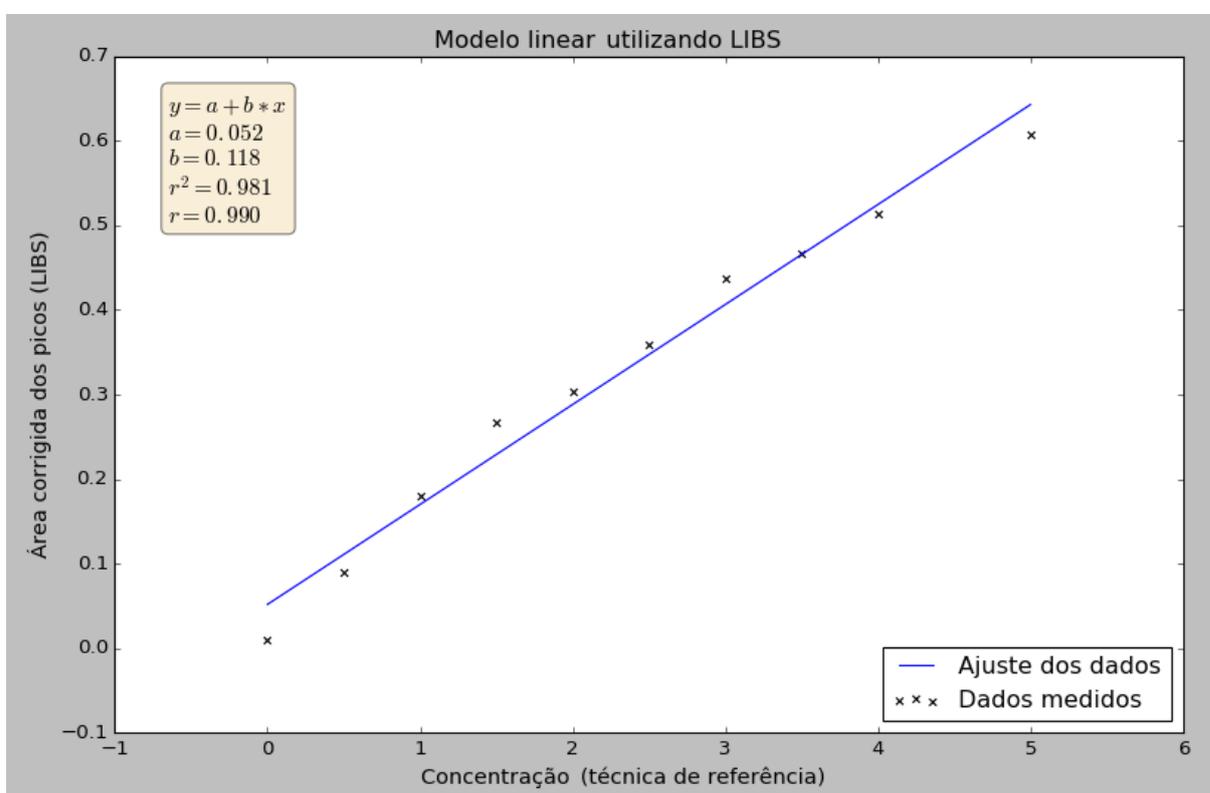
Elemento/Amostra	Curva	Centro do pico	Largura do pico	Intensidade máxima	Área
C-C00	Lorentz	192.9944962212	0.1054734306	2.3522991332	0.3897225159
C-C05	Lorentz	193.0221258962	0.209496306	30.2361811786	9.9500024028
C-C10	Lorentz	193.0276974738	0.2084489498	35.8737899276	11.7461853397
C-C15	Lorentz	193.024515437	0.2254774186	43.6840419964	15.4719747193
C-C20	Lorentz	193.0209929191	0.2280266787	54.8036232365	19.629751898
C-C25	Lorentz	193.0254892758	0.2330246309	55.5040641155	20.3163876063
C-C30	Lorentz	193.0264744595	0.2446258966	56.6776475567	21.778808881
C-C35	Lorentz	193.0223784843	0.2329556601	66.9526480685	24.499707677
C-C40	Lorentz	193.0260222299	0.240156462	64.7059876821	24.4094854503
C-C50	Lorentz	193.0275006541	0.2555904223	60.7554802991	24.3921403963

Fonte: Autoria própria.

5.1.4. Módulo de criação de modelo

Finalmente, na última etapa do programa, é feito um modelo de calibração para o elemento utilizando LIBS. Essa etapa requer o uso de uma técnica de referência, devendo ser feita a caracterização de todas as amostras com ela. O módulo então correlaciona às áreas dos picos que foram calculadas, com as concentrações obtidas com a técnica de referência, gerando uma curva de calibração (Figura 49) que pode ser utilizado para quantificação do elemento usando apenas LIBS, sem posterior necessidade do uso da técnica de referência.

Figura 49 – Modelo linear para o elemento de análise utilizando LIBS



Fonte: Autoria própria.

5.2. Interfaces de usuário

Todos os módulos do programa de análise de espectros LIBS foram escritos em Python. Para simplificar, durante o desenvolvimento, utilizou-se preferencialmente os módulos do programa em modo texto/terminal, onde cada um deles (Leitura, Isolamento, Ajuste de curvas e Criação de modelo) funcionavam como programas independentes que deveriam ser executados na ordem correta para se obter os resultados desejados. Entretanto, por mais funcional que seja, essa abordagem se distancia da proposta desse trabalho, que busca o uso de softwares mais simplificados e intuitivos para os operadores. Desse modo,

para simplificar o modo como o programa era executado, ao invés de programas individuais, um modelo mais eficiente seria o uso de uma única interface que permita ao usuário executar apenas os módulos que lhe são de interesse.

Primeiramente, foi necessário atualizar cada um dos programas para poderem receber parâmetros externos. Se antes só era necessário executar o programa e ele fazia todos os passos necessários, agora era necessário executá-lo com parâmetros de entrada, como pasta onde estão localizadas amostras, tipo de curva para ajuste, valor do critério de outliers etc. Aplicada essa lógica isoladamente para cada um dos programas escritos, foi criado um novo programa, aqui chamado de **Python LIBS Analyser**, que acessa todos os módulos, efetuando todas as operações de antes com muito mais agilidade, pois o acesso é único. A estrutura do programa é mostrada na Figura 50.

Figura 50 – Estrutura de pastas do Python LIBS Analyser

```
~/Desktop/Embrapa/Python LIBS Analyser $ tree
├── changelog
├── inputs
├── inputs.xls
├── outputs
│   ├── elements
│   ├── isolated
│   ├── mean
│   ├── model
│   ├── pictures
│   │   ├── fit
│   │   ├── isolated
│   │   └── total
│   └── total
├── pyfiles
│   ├── findalpha2.py
│   ├── findalpha.py
│   ├── findlod.py
│   ├── fitdata.py
│   ├── fitdata_sdv.py
│   ├── functions.py
│   ├── importdata.py
│   ├── __init__.py
│   ├── isodata.py
│   ├── justplot.py
│   └── modeldata.py
├── pylibsanalyser 1.18.py
├── spectra
│   ├── A
│   │   ├── 010.ols
│   │   ├── 01.ols
│   │   ├── 02.ols
│   │   ├── 03.ols
│   │   ├── 04.ols
│   │   ├── 05.ols
│   │   ├── 06.ols
│   │   ├── 07.ols
│   │   ├── 08.ols
│   │   └── 09.ols
│   └── B
│       ├── 110.ols
│       ├── 11.ols
│       ├── 12.ols
│       ├── 13.ols
│       ├── 14.ols
│       ├── 15.ols
│       ├── 16.ols
│       ├── 17.ols
│       ├── 18.ols
│       └── 19.ols
15 directories, 34 files
```

Fonte: Autoria própria.

O programa é basicamente formado pelas pastas:

1. **Raiz do programa** – É onde todos os arquivos estão. Possui as pastas *inputs* (arquivos de entrada), *outputs* (arquivos de saída), *pyfiles* (funções do programa) e *spectra* (espectros LIBS, em uma pasta por amostra). Na raiz também temos alguns arquivos:

o *changelog* (que faz a documentação das últimas mudanças e bugs encontrados no programa), o arquivo *inputs.xls* (onde o usuário coloca informações úteis para a execução, como elementos a serem analisados e intervalos de picos) e finalmente o programa principal, *pylibsanalyser.py*, atualmente em sua versão 1.18, cujo código fonte pode ser verificado no Apêndice A;

2. **Outputs** – Pasta com os arquivos de saída, e em suas subpastas são salvos todos os arquivos de texto resultantes das análises. Na pasta *total* são escritos os espectros após a importação. Na pasta *isolated* os espectros isolados, na *elements* o relatório do ajuste de curva (as áreas dos picos) de cada elemento. Na *mean* todas as médias e na *model* um arquivo de Excel (xls) com o modelo de quantificação criado, baseado no elemento que o usuário escolheu. Na pasta *pictures* são salvas todas as figuras e gráficos que o programa gera durante sua execução, em formato de imagem (png), para que o usuário possa ir acompanhando a evolução da análise.
3. **Pyfiles** – A pasta com todos os programas. Nela estão todas as rotinas a serem utilizadas, separadas em módulos/funções. Os passos da análise são: Importação (*importdata.py*) → Isolamento (*isotdata.py*) → Fit (*fitdata.py*) → Modelo (*modeldata.py*). Existem outros arquivos apenas para testes, não sendo essenciais.

5.2.1. Interface modo texto

Ao ser executado o arquivo *pylibsanalyser.py*, aparece uma interface (UI) muito simplificada para o usuário, e a partir dela (digitando números de 1 a 11) é possível escolher quais operações o usuário deseja executar. Durante a execução, o programa exibe uma barra de progresso e ao final mostra uma mensagem de que todos os processos foram executados de acordo com o esperado (Figura 51).

Figura 51 – UI do programa após a execução de todos os módulos

```

Terminal
Python LIBS Analyser 1.18

O que deseja fazer?

                Opções  Valores
    Todos os passos          1
  Apenas importar espectros  2
    Apenas isolar picos     3
  Apenas fazer fit dos picos 4
    Apenas criar modelo     5
    Importar e isolar        6
    Isolar em diante        7
      Fit em diante         8
  Limpar todas as pastas    9
  Fitar alternativo (SDV)  10
      SAIR                  11

Insira a opção desejada: 1

Importando:
100% |#####|

Isolando:
100% |#####|
100% |#####|
100% |#####|

Fitando:
100% |#####|

Criando modelo:
100% |#####|

Todas as operações foram executadas com sucesso!

-----
(program exited with code: 0)
Press return to continue
  
```

Fonte: Autoria própria.

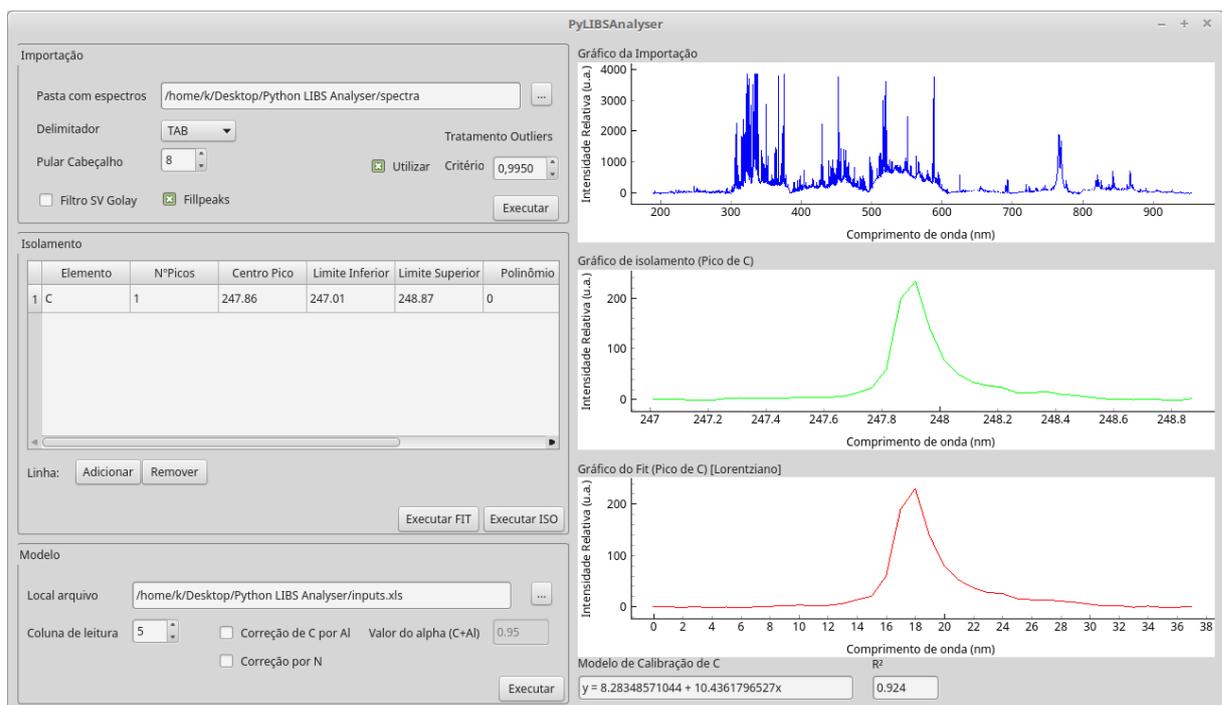
5.2.2. Interface gráfica

Apesar de mais simplificada que a versão original do programa, a UI ainda depende de diversas informações que ficam espalhadas no código fonte do *pylibsanalyser.py*, além de ter que ser preenchido previamente o arquivo *inputs.xls* (o que, em última instância, depende de um software que abre arquivos xls instalado na máquina do usuário, como o

Microsoft Excel® ou LibreOffice Calc¹⁹). Assim, desenvolveu-se uma interface gráfica – ou simplesmente GUI – para poder integrar todos os módulos de processamento.

A nova interface gráfica (Figura 52), nada mais é que uma janela para interação com o usuário que engloba todas essas funcionalidades da UI. A seleção de opções de 1 a 11 agora virou um botão “Executar” na GUI. Nesta versão, os gráficos podem ser visualizados em pequenas janelas laterais, o que antes só era possível visualizando a pasta *pictures* ou traçando os gráficos com os dados manualmente. Ao final, o modelo é exibido diretamente na tela do usuário, sem a necessidade de verificar seu valor seja na imagem salva ou no arquivo xls salvo na pasta *model*. O código fonte dessa interface gráfica pode ser verificado no Apêndice B.

Figura 52 – GUI do Python Libs Analyser



Fonte: Autoria própria.

¹⁹ Pode ser obtido em <<http://www.libreoffice.org/>>

CAPÍTULO 6 – RESULTADOS E DISCUSSÕES

O desafio de se desenvolver um sistema LIBS portátil para análise de C em solos, proposto nessa dissertação, foi trabalhado a partir de três frentes, distintas, mas interconectadas: 1) escolher os equipamentos integrantes do LIBS portátil e dimensionar um protótipo para fabricação, 2) desenvolver e testar o disparo do laser e aquisição de espectros e 3) desenvolver e testar os softwares de análise de espectros. Assim, os resultados estão separados nesses três temas.

6.1. Desenvolvimento do protótipo

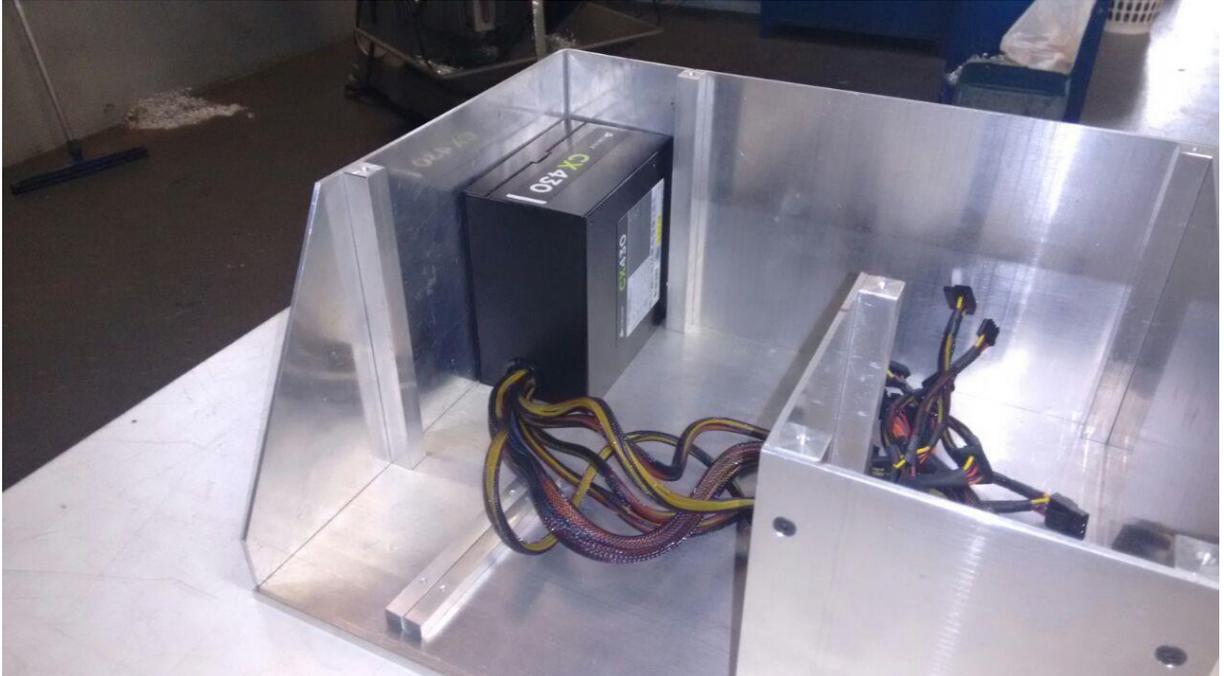
Todos os equipamentos necessários a montagem do sistema foram adquiridos: monitor touch, lentes, fibras ópticas, laser e espectrômetros. A partir das figuras desenhadas para a fabricação da caixa que abrigará todo o sistema (seção 4.5), iniciou-se a fabricação do mesmo. As figuras 53 a 58 abaixo mostram as etapas do processo de fabricação até o presente estado.

Figura 53 – Fabricação do protótipo: base, laterais e abertura para a fonte de computador



Fonte: Autoria própria.

Figura 54 – Fabricação do protótipo: efetuado corte para fixação do monitor touch



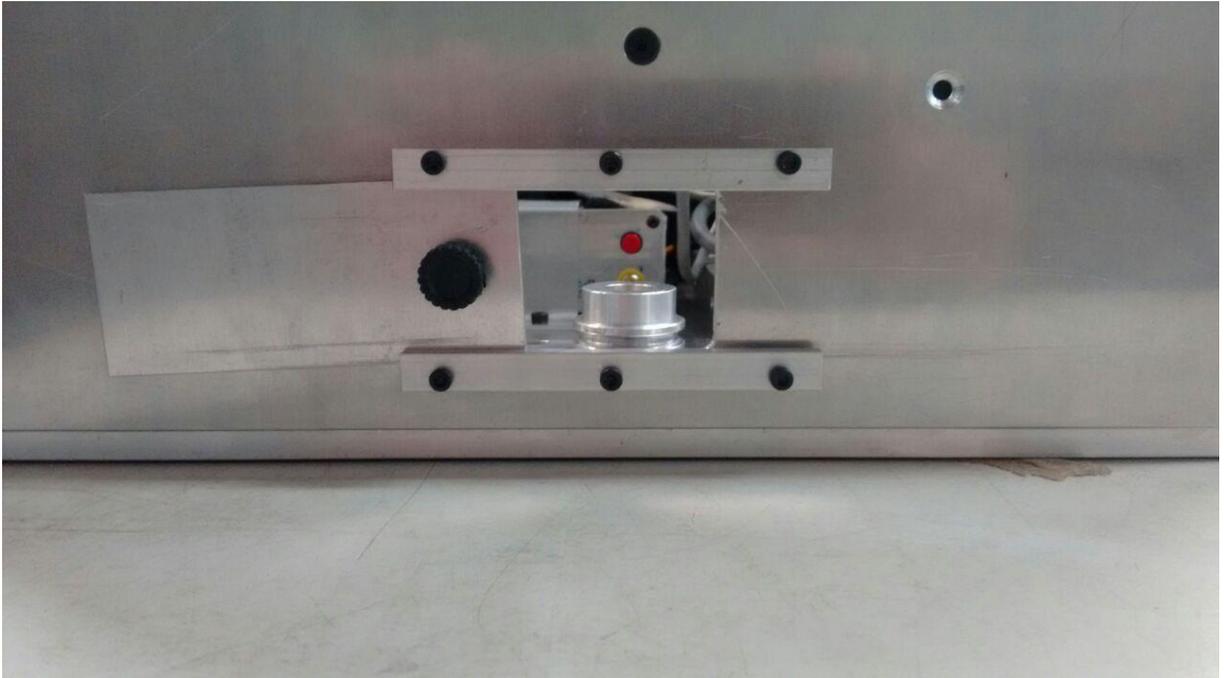
Fonte: Autoria própria.

Figura 55 – Fabricação do protótipo: laser fixado e base do porta amostra com eixo para rotação



Fonte: Autoria própria.

Figura 56 – Fabricação do protótipo: abertura lateral para inserção do porta amostra



Fonte: Autoria própria.

Figura 57 – Fabricação do protótipo: furo lateral para respiro do cooler da fonte do computador e tampa fixada



Fonte: Autoria própria.

Figura 58 – Fabricação do protótipo: tampa superior e monitor touch fixado com inclinação de 50°



Fonte: Autoria própria.

Figura 59 – Fabricação do protótipo: equipamento após anodização na cor preta, sem tampa superior



Fonte: Autoria própria.

6.2. Testes com o sistema de disparo do laser e aquisição de espectros

Utilizando o software Python Portable LIBS, que engloba o controle da placa de disparo (utilizando uma das portas GPIO do Raspberry Pi) e a aquisição do sinal do espectrômetro (utilizando o driver desenvolvido para funcionamento em Python), foi possível efetuar medições com o equipamento montado em bancada. Um dos espectros é o que foi mostrado na Figura 31.

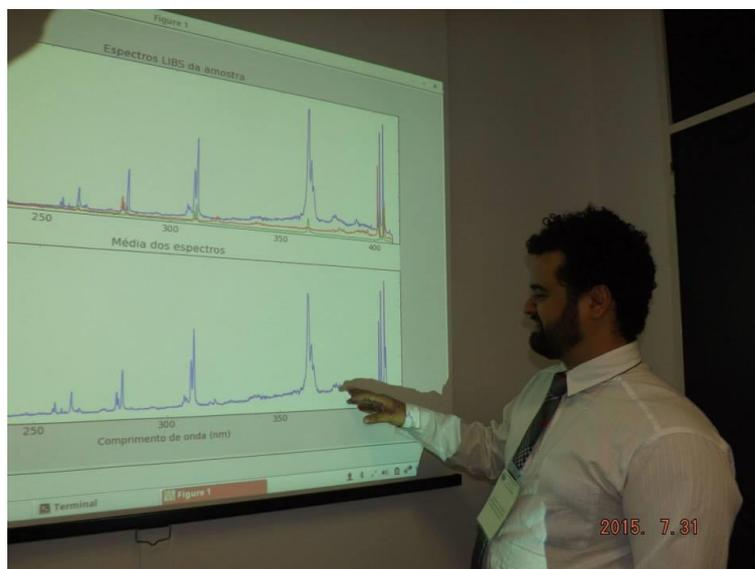
Também, esse conjunto de equipamentos foi montado em um rover, o Robô Mirã, apresentado durante o no II Workshop Sobre Laser em Química Analítica, na Universidade Federal de São Carlos e Embrapa Instrumentação nos dias 27 a 31 de julho de 2015. Durante um experimento com os participantes do evento, foi ministrado uma palestra, e exibidos os espectros medidos durante o experimento (Figuras 60 e 61).

Figura 60 – Robô Mirã durante o II Workshop Sobre Laser em Química Analítica



Fonte: <https://www.facebook.com/wlqa2015>

Figura 61 – Apresentação dos espectros medidos durante um dos experimentos com o robô Mirã



Fonte: <https://www.facebook.com/wlqa2015>

O sistema também foi testado e apresentado durante a 67ª Reunião Anual da Sociedade Brasileira para o Progresso da Ciência, que ocorreu em São Carlos – SP de 12 a 18 de julho de 2015, ganhando considerada visibilidade, sendo feita inclusive uma reportagem na Revista Globo Rural²⁰, além de uma matéria completa na TV Terra Viva²¹, em outubro de 2015.

6.3. Validação do software de análise de espectros

Para validar o software de análise de dados do LIBS compacto escolheu-se analisar o Carbono. Para tal utilizou-se dois tipos de amostras.

1. Um grupo de controle (para teste) com o objetivo de verificar se os algoritmos escritos estão chegando a resultados coerentes e;
2. Um grupo de análise propriamente dita, onde é feita a caracterização de C em solos de pastagens brasileiras.

²⁰ Matéria disponível em <<http://revistagloborural.globo.com/Tecnologia-no-Campo/noticia/2015/09/robo-emite-luz-capaz-de-mapear-lavouras.html>>

²¹ Reportagem disponível em <<http://mais.uol.com.br/view/8tncj14f713t/embrapa-desenvolve-robo-especifico-para-analise-do-solo-04028D183064D4B15326>>

Como o equipamento proposto está em fase final de construção, não foi possível utilizá-lo para a validação dos softwares. Assim, foi feita a caracterização das amostras utilizando o equipamento de bancada LIBS2500Plus, e esses espectros foram utilizados no Python LIBS Analyser.

6.3.1. Amostras – Grupo de controle

O conjunto de amostras de testes tratam-se de pastilhas de KBr com grafite. Elas foram misturadas estequiometricamente para se obter 10 amostras, cada uma com uma quantidade de teor de C, como mostra a Tabela 5:

Tabela 5 – Teores de C para as amostras do grupo de controle

Amostra	Teor de C (%)
1	0,0
2	0,5
3	1,0
4	1,5
5	2,0
6	2,5
7	3,0
8	3,5
9	4,0
10	5,0

Fonte: Autoria própria.

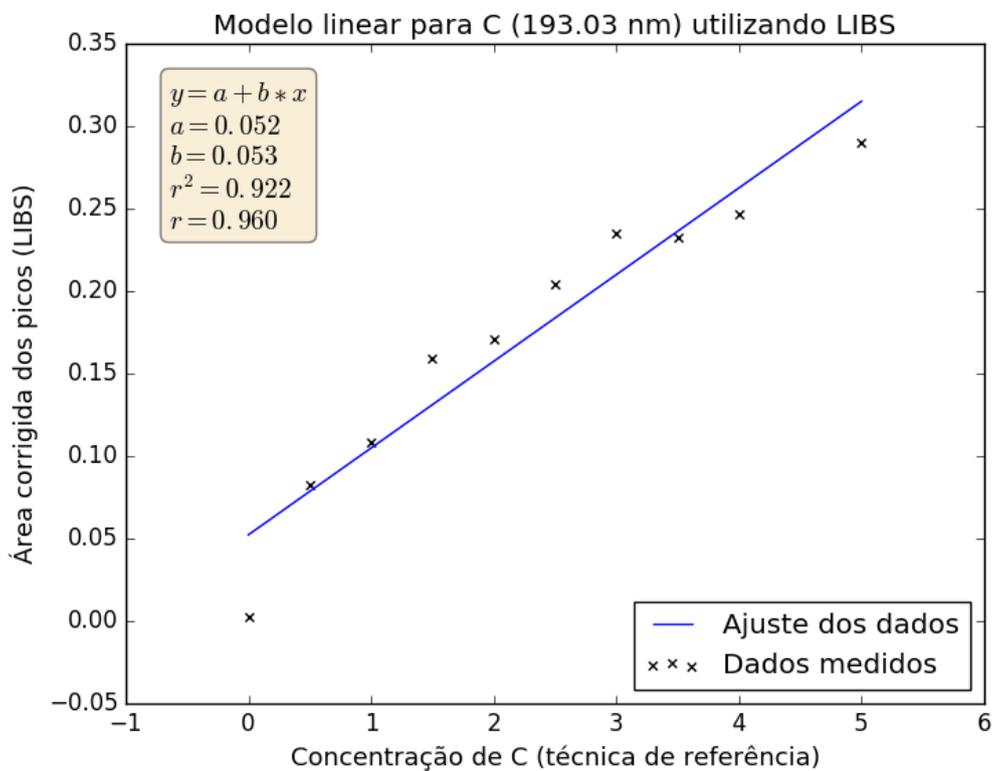
Após a mistura e homogeneização, cada um dos grupos foi prensado à 6 T por 30 s para a confecção de pastilhas. Essas pastilhas então foram caracterizadas com o equipamento LIBS 2500Plus, obtendo-se 10 espectros por amostra.

6.3.1.1. Resultados das amostras do grupo de controle

As amostras do grupo de teste, compostas pelas amostras controladas de KBr e grafite, foram submetidas primeiramente ao software Python LIBS Analyser. Determinou-se a

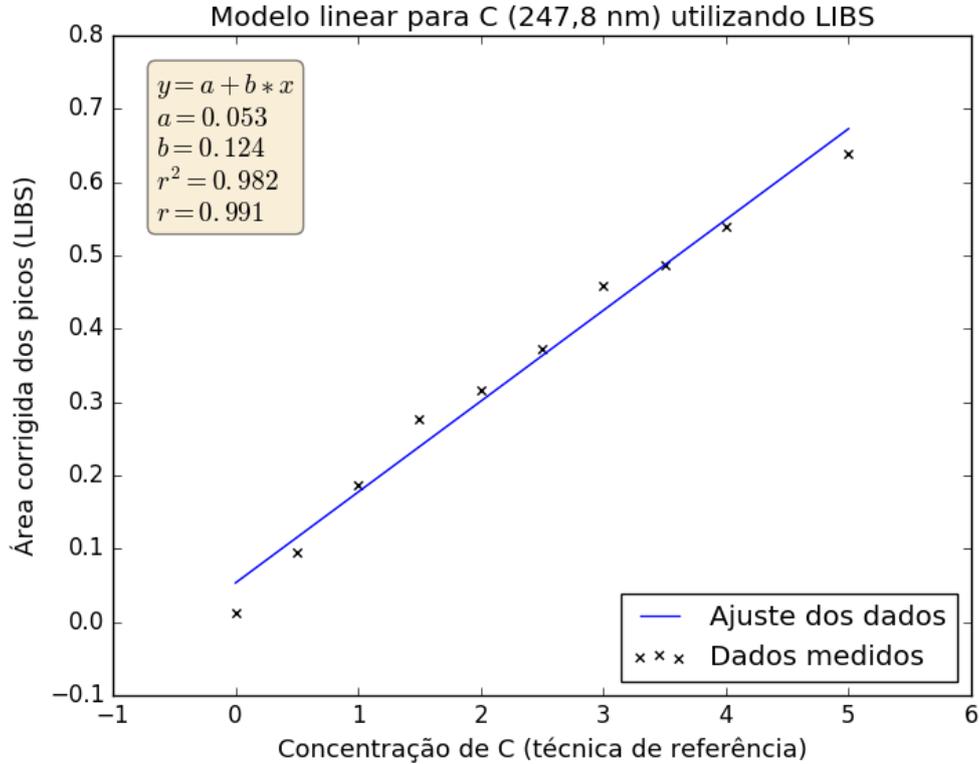
área dos picos de Carbono 193,03 e 274,8 nm estabelecendo 0,9916 para o critério de outliers e sem a normalização pela área da reta abaixo do pico (utilizada para corrigir a inclinação). Em seguida determinou-se a correlação da área de ambos os picos de carbono com o teor de carbono presente na pastilha, obtendo-se os coeficientes de correlação de 0,96 para o pico de carbono em 193,03 nm e 0,99 para o pico em 247,8 nm. As curvas de calibração estão discriminadas nas Figuras 62 e 63.

Figura 62 – Modelo linear para a quantificação de C nas amostras controladas para o pico em 192,03 nm



Fonte: Autoria própria.

Figura 63 – Modelo linear para a quantificação de C nas amostras controladas para o pico em 247,8 nm

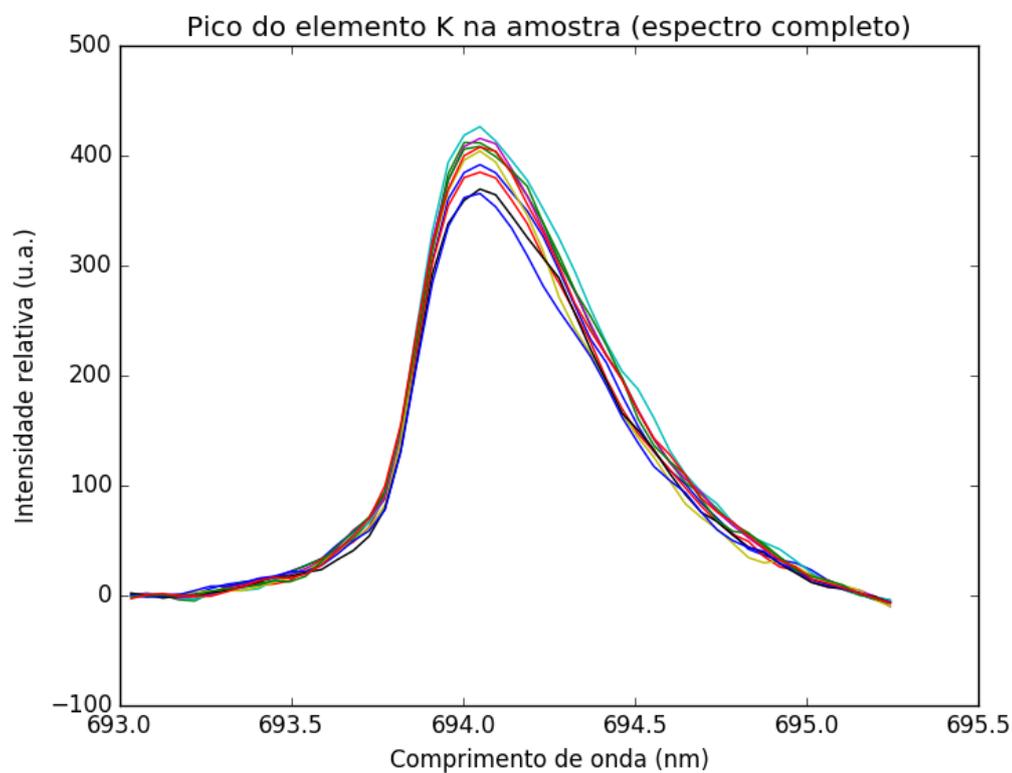


Fonte: Autoria própria.

Tais resultados com altíssimas correlações para as amostras controladas de KBr com grafite não só corroboram com a hipótese de os softwares estarem funcionando apropriadamente em todos os seus módulos, desde a importação até a geração do modelo, mas também mostram a capacidade do LIBS em determinar o teor de C total quando comparado com valores conhecidos.

Além de C, o software também conseguiu detectar o pico de potássio (K) em 693,87 nm (Figura 64), que é um dos macronutrientes mais importantes relacionados à fertilidade do solo. No caso do K não foi feita caracterização com uma técnica de referência, mas eventualmente pode ser feita a mesma metodologia de análise, obtendo-se um modelo de calibração.

Figura 64 – Pico de K em 693,87 nm detectado nas amostras de teste



Fonte: Autoria própria.

6.3.2. Amostras – Grupo de análise

As amostras de solo utilizadas para a validação foram coletadas na Embrapa Pecuária Sudeste - Fazenda Canchim, na cidade de São Carlos – SP, situada à Rodovia Washington Luiz, Km 235. Trata-se de um experimento de campo que consiste em quatro diferentes sistemas de manejo, mais a mata nativa como área de referência (Tabela 6).

Tabela 6 – Características das áreas experimentais utilizadas para a coleta do conjunto de amostras de solo caracterizadas

Área	Manejo	Textura	Vegetação
A1	Irrigado	Média arenosa	<i>Panicummaximum</i>
A2	Intensivo sequeiro	Argilosa	<i>Panicummaximum</i>
A3	Em recuperação	Argilosa	<i>Brachiariadecumbens</i>
A4	Degradado	Média arenosa	<i>Brachiariadecumbens</i>

Fonte: XAVIER (Adaptado).

Para a coleta das amostras do solo foram abertas trincheiras com largura de 100 cm e com profundidade de aproximadamente 120 cm. As amostras foram coletadas em oito diferentes profundidades, de 0 a 100 cm, nos intervalos de 0-5, 5-10, 10-20, 20-30, 30-40, 40-60, 60-80 e 80-100 cm, em seis trincheiras para cada uma das áreas. Totalizando 48 amostras por área. Coletou-se aproximadamente 200 g de solo para cada amostra, este material foi levado ao laboratório, onde se iniciou o preparo de amostras. Tal preparo consistiu na secagem dos solos à temperatura ambiente e na remoção de raízes e restos vegetais por catação, seguido por homogeneização (XAVIER, 2014).

O solo foi passado em peneira de 2 mm para a remoção de pedras e gravetos. Depois, foi triturado com a utilização de almofariz e pistilo. Uma porção, de aproximadamente 5g, dessas amostras foi remoída e passada em peneira de 0,150 mm (100 mesh) para as análises com LIBS. Por fim, as amostras de solos foram prensadas em pastilhas, a aproximadamente 8 toneladas, tendo dimensões próximas de 1 cm de diâmetro, 2 mm de espessura e 0,5 g de massa. Ao final, essas pastilhas foram medidas no equipamento LIBS2500Plus num total de 60 tiros por amostra, 30 em cada face.

Ao todo foram medidas 240 amostras, mas para nossos testes foram utilizadas apenas 16 amostras. Escolhemos um número pequeno para minimizar custos computacionais, e essas amostras foram escolhidas englobando uma boa faixa de teor de C. Para garantir um conjunto bem heterogêneo, utilizamos amostras das áreas A1, A2, A3 e A4, não fazendo nenhuma distinção de grupo, seja com relação à textura do solo ou ao manejo.

Tabela 7 – Valores de concentração de C e tipo de textura de solo para o conjunto de amostras de solos utilizados

Amostra	Concentração de C	Textura do solo
0	1.24	Média arenosa
2-20	0.78	Média arenosa
3-31	0.61	Média arenosa
4-38	0.77	Média arenosa
5-39	0.68	Média arenosa
6-49	1.83	Argilosa
7-51	1.49	Argilosa
8-52	1.32	Argilosa
9-142	1.16	Argilosa
10-143	0.94	Argilosa
11-155	1.11	Argilosa
12-167	0.64	Média arenosa
13-172	1.07	Média arenosa
14-174	0.81	Média arenosa
15-175	0.69	Média arenosa
16-184	0.58	Média arenosa

Fonte: Autoria própria.

6.3.2.1. Técnica de referência para análise de C – CHNS

A análise elementar (CHNS) é uma técnica instrumental que já dispensa boa parte do preparo de amostra e assim gera uma quantidade menor de resíduos químicos. Atualmente, é a técnica mais confiável na determinação de carbono em amostras de solo, podendo ser utilizada como referência na estimativa de carbono por outras técnicas instrumentais.

Essa técnica permite a determinação porcentual dos quatro elementos químicos, carbono (C), hidrogênio (H), nitrogênio (N) e enxofre (S) de uma amostra. A tecnologia associada à análise CHNS é a combustão seca, dessa forma a amostra é totalmente queimada e em seguida os gases da sua combustão são analisados. Os produtos dessa combustão são CO₂ para análise de C, H₂O para análise de H e NO para análise de N, os quais são eluídos e separados por uma coluna cromatográfica.

As determinações de C foram feitas em duplicata, utilizando o equipamento CHNS/O da Perkin-Elmer (2400 Series II CHNS/O Elemental Analyzer) pertencente a Embrapa Instrumentação. Pesou-se cerca de 10 mg das amostras de solo inteiro já peneiradas a 100 mesh. Tais amostras foram pesadas diretamente em cápsulas de estanho consumíveis,

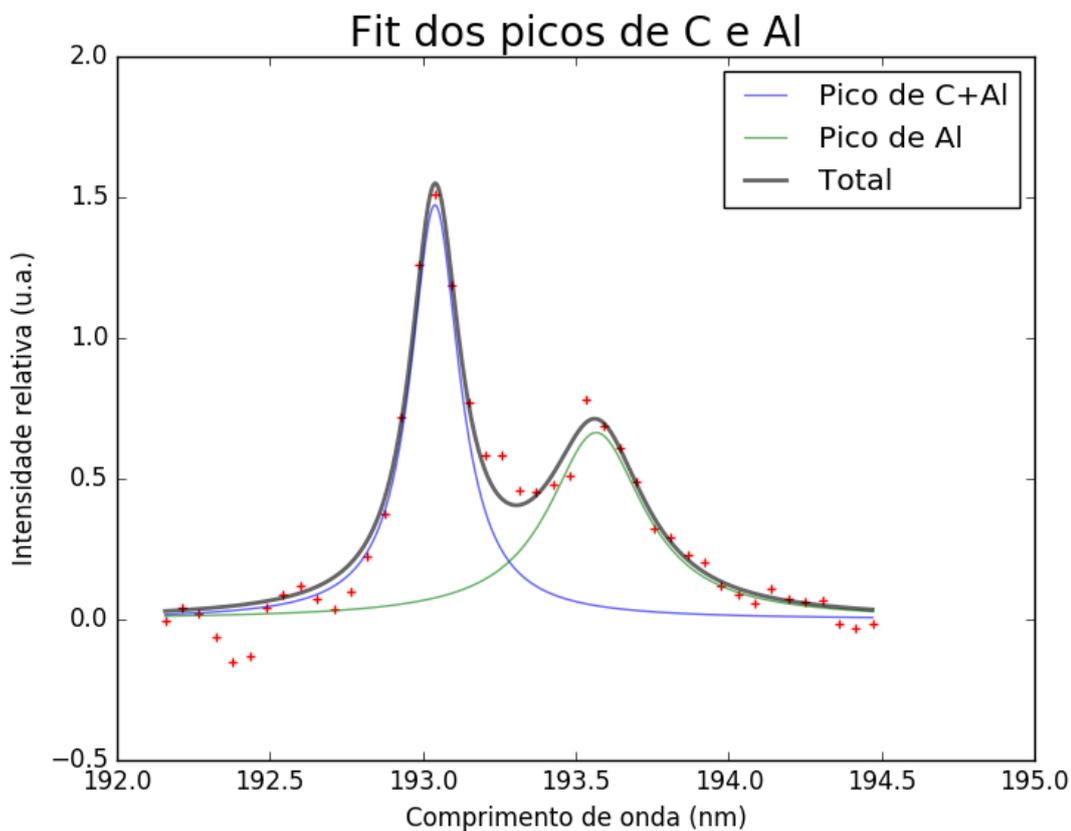
utilizando uma microbalança Perkin-Elmer, modelo AD6, que é conectada ao CHNS/O para aquisição direta dos valores de massa.

Em seguida, as cápsulas de estanho foram fechadas manualmente e introduzidas no forno do analisador do equipamento. O estanho presente na cápsula é utilizado para acelerar o processo de combustão. Os resultados da análise elementar são baseados em um valor de padrão conhecido, a acetanilida, um padrão orgânico de composição elementar conhecida. (XAVIER, 2014)

6.3.2.2. Resultados das amostras do grupo de análise

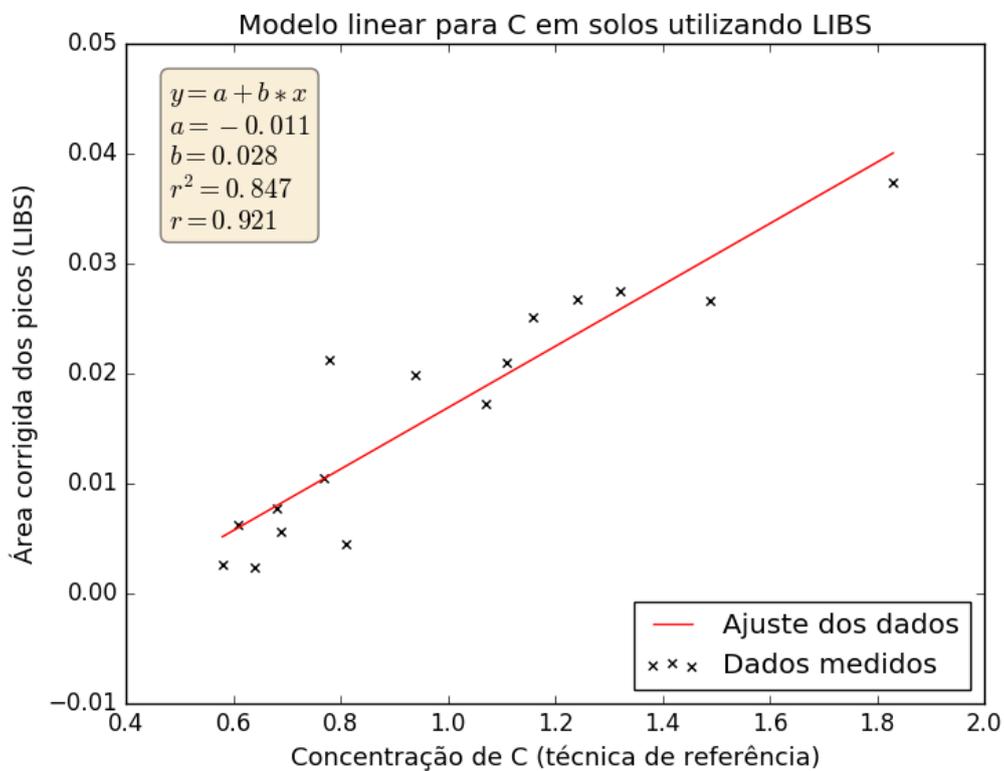
Os espectros LIBS obtidos para as pastilhas de solo apresentaram as linhas de C fortemente interferidas pelas linhas de emissão de Alumínio e Ferro. Dessa forma, não foi possível utilizar diretamente a linha em 247,8 nm; a linha 193,03 necessita de um modelo de correção como o descrito no tópico 2.2.1. No artigo que explica o método de correção utilizou funções gaussianas para o ajuste da curva, no entanto, para o conjunto de amostras de solo utilizado nesse experimento, a função Lorentziana apresentou o melhor ajuste de curva. A Figura 65 mostra o resultado desse ajuste.

Figura 65 – Fit utilizando uma curva Lorentziana para os picos de C e Al



O modelo de calibração foi gerado através da área do pico de C calculado para cada amostra como função do valor de concentração obtido com a técnica de referência, CHNS. O trabalho publicado por NICOLODELI *et al* utilizou um modelo exponencial, mas para o conjunto de amostras descrito aqui não resultou em melhoras significativas, o que motivou a manter o modelo linear (Figura 66), devido sua simplicidade e menor custo computacional.

Figura 66 – Modelo linear para a concentração de C em função da área do pico para as amostras de solo



CAPÍTULO 7 – CONCLUSÃO

Essa dissertação mostra que é possível a criação de um LIBS portátil utilizando soluções essencialmente simples em sua concepção. O trabalho conseguiu atingir todos os objetivos específicos com os quais tinha se proposto a resolver, abordando diversos aspectos da portabilização da técnica LIBS, desde a escolha de um minicomputador Raspberry Pi para o projeto, o desenvolvimento de um circuito eletrônico para o controle de disparo do laser e a análise de dados.

Além disso, todas as soluções foram integradas graças a escolha da linguagem Python e a grande gama de bibliotecas open source disponíveis para uso. Foram desenvolvidos dois programas: o *Python Portable LIBS*, que engloba o controle de disparo do laser e a aquisição do sinal dos espectrômetros StellarNet; e o programa *Python LIBS Analyser* em sua versão gráfica, que possui diversas opções para o operador manipular os seus resultados e construir modelos de calibração de um modo simplificado. Também, a criação do Driver Linux para os espectrômetros StellarNet EP2000-LR integra o esforço em se criar um protótipo funcional e uso fácil e intuitivo para o operador.

O protótipo foi projetado levando em consideração diversos aspectos, incluindo a determinação das melhores posições para os equipamentos dentro da caixa. Ao longo do processo de prototipagem, foram trabalhados cada um itens que compõem um LIBS, como um sistema para ajuste da lente de focalização do laser, um sistema para coleta com fibra óptica e uma base para o porta amostra, capaz de rotacionar com uma precisão de milímetros, permitindo facilmente medições com 30 tiros em cada face.

Por ainda estar em fase de fabricação, não foi possível validar o software de análise com os espectros obtidos pelo próprio sistema. Mas o mesmo teve um ótimo desempenho com os espectros do LIBS de bancada da Ocean Optics, o LIBS2500Plus, obtendo para o conjunto de amostras controladas, de KBr misturada com concentrações conhecidas de carbono, os valores de $R=0,96$ para o pico de carbono em 193,03 nm e $R=0,99$ para o pico em 247,8 nm. Para o conjunto de solos, embora não tão alto, obteve-se um ótimo resultado também, $R=0,92$ para o pico de carbono em 193,03 nm corrigido a influência do Alumínio; esse resultado se torna mais expressivo ainda dado o fato do conjunto de amostras

utilizadas ser extremamente heterogêneo, incluindo no mesmo grupo solos médio arenosos e argilosos.

Mesmo havendo equipamentos LIBS portáteis sendo fabricados em diversos países, poucos focam na medição de C em amostras de solo, demonstrando sua relevância no contexto internacional. No cenário nacional, tal relevância se acentua, como um equipamento pioneiro. Espera-se que os conhecimentos aqui tratados venham a ser um motivador para outros grupos de pesquisa ao redor do Brasil, fomentando cada vez mais o interesse nacional em desenvolver tecnologia avançada para o campo, seja no âmbito ambiental, quando focando no aumento de produtividade, buscando métodos mais precisos de se levantar mapas de nutrientes nas lavouras.

CAPÍTULO 8 – REFERÊNCIAS BIBLIOGRÁFICAS

- CENTRO DE ESTUDOS AVANÇADOS EM ECONOMIA APLICADA (CPEA); Relatório do PIB do Agronegócio, 2016. Disponível em: <<http://www.cepea.esalq.usp.br/br/pib-do-agronegocio-brasileiro.aspx>>. Acesso em: 3 fev. 2017.
- ASSOCIAÇÃO NACIONAL PARA DIFUSÃO DE ADUBOS (ANDA); Estatísticas – Principais indicadores do setor de fertilizantes, 2017. Disponível em: <http://www.anda.org.br/estatistica/Principais_Indicadores_2016.pdf>. Acesso em: 3 fev. 2017.
- XAVIER, A. A. P.; Avaliação do acúmulo e emissão de carbono do solo sob sistemas produtivos de pastagem, Dissertação de Mestrado, Biblioteca de Teses e Dissertações, USP, São Carlos, 2014. Disponível em: <<http://www.teses.usp.br/teses/disponiveis/75/75135/tde-10122014-101014/pt-br.php>>. Acesso em 3 fev. 2017
- CENTRO DE CIÊNCIA DO SISTEMA TERRESTRE (CCST); Contribuição do grupo de trabalho I ao quarto relatório de avaliação do painel intergovernamental sobre mudança do clima - sumário para os formuladores de políticas. Genebra, 2007. Disponível em: <<https://www.ipcc.ch/pdf/reports-nonUN-translations/portuguese/ar4-wg1-spm.pdf>>. Acesso em: 3 fev. 2017.
- CLIMATE CHANGE; Working Group I Contribution to the IPCC Fifth Assessment Report. Zähringerstrasse, 2013. Disponível em: <http://www.climatechange2013.org/images/report/WG1AR5_ALL_FINAL.pdf>. Acesso em: 3 fev. 2017.
- SEGNINI, A.; SANTOS, L. M. D.; SILVA, W. T. L. D.; MARTIN-NETO, L.; BORATO, C. E.; MELO, W. J. D.; BOLONHEZI, D.; Estudo comparativo de métodos para a determinação da concentração de carbono em solos com altos teores de Fe (Latossolos). Química Nova, v. 31, 2008, p. 94.
- SCHMIDT, S. M.; SORAUF, K. J.; MILLER, K. E.; Spark-Induced breakdown spectroscopy and multivariate analysis applied to the measurement of total carbon in soil, Massachusetts, Estados Unidos, 2012, p. 7.
- DUARTE, J. B.; Princípios sobre delineamentos em experimentação agrícola, Goiânia, Goiás. Trabalho de conclusão de Curso de Especialização em Estatística, DEI-IMF/UFG, 1996.
- CORRÊA, A. e ZUIN, V. G.; Química Verde - Fundamentos e Aplicações. Editora: EDUFSCAR, 2009.
- HERNÁNDEZ, R. C.; Aplicaciones de la técnica LIBS (Laser induced breakdown spectroscopy) a la caracterización de materiales, Encuentro Internacional de Investigadores en Material y tecnología do Plasma, Bucaramanga, Colombia, 2011, p. 22.
- FERREIRA, E. C.; GOMES NETO, J. A.; MILORI, D. M. B. P.; FERREIRA, E. J.; ANZANO, J. M.; Laser-induced breakdown spectroscopy: Extending its application to soil pH measurements. Spectrochimica Acta Part B 110, 2015, p. 96–99.

- MARAGONI, B. S.; SILVA, K. S. G.; NICOLODELLI, G.; SENESIO, G. S.; CABRAL, J. S.; VILLAS-BOAS, P. R.; SILVA, C. S.; TEIXEIRA, P. C.; TEIXEIRA, P. C.; NOGUEIRA, A. R. de A.; BENITES, V. de M.; MILORI, D. M. B. P.; Phosphorus quantification in fertilizers using laser induced breakdown spectroscopy (LIBS): a methodology of analysis to correct physical matrix effects. *Analytical Methods*, 2015, 8, p. 78.
- CERVANTES, C.; Quantificação e correção de matriz física de potássio em fertilizantes utilizando espectroscopia de emissão ótica com plasma induzido por laser (LIBS). *Anais da VIII Jornada Científica Embrapa São Carlos*, 2016, p. 21.
- NICOLODELLI, G.; MARANGONI, B. S.; CABRAL, J. S.; VILLAS-BOAS, P. R.; SENESI, G.S.; SANTOS, C. H. dos; ROMANO, R. A.; SEGNINI, A.; LUCAS, Y.; MONTES, C. R.; MILORI, D. M. B. P.; Quantification of total carbon in soil using laser-induced breakdown spectroscopy: a method to correct interference lines. *Applied Optics*, Nova York, v. 53, n. 10, p. 2170-2176, 2014.
- VILLAS-BOAS, P. R.; Laser-induced breakdown spectroscopy to determine soil texture: A fast analytical technique. *Geoderma* 263, 2016, p. 195-202.
- OSORIO, L. M.; CABRERA, L. V. P.; GARCÍA, M. A. A.; REYES, T. F.; RAVELO, I.; Portable LIBS system for determining the composition of multilayer structures on objects of cultural value. *Journal of Physics: Conference Series*, 274, 2011, 012093.
- RAKOVSKÝ, J.; ČERMÁK, P.; MUSSET, O.; VEIS, P.; A review of the development of portable laser induced breakdown spectroscopy and its applications. *Spectrochimica Acta Part B* 101, 2014, p. 269.
- CAMARGOS, S. L.; Análise química do Solo: Amostragem de solos e preparo de amostras para análise. In: *Fertilidade dos Solos*. Universidade Federal de Mato Grosso. Faculdade de Agronomia e Medicina Veterinária. Departamento de Solos e Engenharia Rural. Disciplina Solos II. 2005.
- BAUDELET&SMITH.; The first years of laser-induced breakdown spectroscopy. *J. Anal. At. Spectrom.* 28, 2013, p. 624-629.
- SACCHI, C. A.; *Journal of the Optical Society of America B*. 8, 1991, 337.
- NOLL, R.; *Laser-Induced Breakdown Spectroscopy – Fundamentals and Applications*. Springer-Verlag Berlin Heidelberg, 2012.
- FERREIRA, E. C.; MILORI, D. M. B. P.; FERREIRA, E. J.; DOS SANTOS, L. M.; MARTIN-NETO, L.; NOGUEIRA, A. R. A.; Evaluation of laser induced breakdown spectroscopy for multielemental determination in soils under sewage sludge application. *Talanta*, v. 85, n. 1, 2011, p. 435.
- NICOLODELLI, G. ; SENESI, G. S. ; ROMANO, R. A. ; DE OLIVEIRA PERAZZOLI, I. L. ; MILORI, D. M. B. P.; Signal Enhancement in Collinear Double-Pulse Laser-Induced Breakdown Spectroscopy Applied to Different Soils. *Spectrochimica Acta. Part B, Atomic Spectroscopy* , v. 1, 2015, p. 1.

- CREMERS, D. A.; EBINGER, M. H.; BRESHEARS, D. D.; UNKEFER, P. J.; KAMMERDIENER, S. A.; FERRIS, M. J.; CATLETT, K. M.; BROWN, J. R.; Measuring total soil carbon with laser-induced breakdown spectroscopy (LIBS), *J. Environ. Quality* 30, 2001, p. 2202.
- EBINGER, M. H.; NORFLEET, M. L.; BRESHEARS, D. D.; CREMERS, D. A.; FERRIS, M. J.; UNKEFER, P. J.; LAMB, M. S.; GODDARD, K. L.; MEYER, C. W.; Extending the applicability of laser-induced breakdown spectroscopy for total soil carbon measurement, *Soil Sci. Soc. Am. J.* 67, 2003, p. 1616.
- DA SILVA, R. M.; Estudo de uma metodologia para estimar a quantidade de carbono em amostras de solo utilizando espectroscopia de emissão óptica com plasma induzido por laser (LIBS), Dissertação de Mestrado, Biblioteca de Teses e Dissertações, USP, São Carlos, 2009. Disponível em: <<http://www.teses.usp.br/teses/disponiveis/75/75132/tde-25082009-094505/pt-br.php>>. Acesso em 3 fev. 2017
- BELKOV, M. V.; BURAKOV, V. S.; DE GIACOMO, A.; KIRIS, V. V.; RAIKOV, S. N.; TARASENKO, N. V.; Comparison of two laser-induced breakdown spectroscopy techniques for total carbon measurement in soils,” *Spectrochim. Acta, Part B* 64, 2009, p. 899.
- LEWIS, D. E.; MARTINEZ, J.; AKPOVO, C. A.; JOHNSON, L.; CHAUHAN, A.; EDINGTON, M. D.; Discrimination of bacteria from Jamaican bauxite soils using laser-induced break-down spectroscopy, *Anal. Bioanal. Chem.* 401, 2011, p. 2225.
- DIAZ, D; HAHN, D. W.; MOLINAT, A.; Evaluation of laser-induced breakdown spectroscopy (LIBS) as a measurement technique for evaluation of total elemental concentration in soils, *Appl. Spectrosc.* 66, 2012, p. 99.
- AYYALASOMAYAJULA, K. K.; FANG, Y. Y.; SINGH, J. P.; MCINTYRE, D. L.; JAIN, J.; Application of laser-induced breakdown spectroscopy for total carbon quantification in soil samples, *Appl. Opt.* 51, 2012, B149–B154.
- HONDROGIANNIS, E.; ANDERSEN D.; MIZIOLEK A.W.; The evaluation of a new technology for gunshot residue (GSR) analysis in the field, *SPIE Def. Secur. Sens*, 2013, p. 8726.
- MIZIOLEK A.W.; Progress in fieldable laser-induced breakdown spectroscopy (LIBS). M.A. Druy, R.A. Crocombe (Eds.), *SPIE Defense, Secur. Sens, International Society for Optics and Photonics*, 2012, p. 837402-13.
- GROZEVA, M.; PENKOVA, P.; In Situ LIBS Analysis of Valuable Museum Objects: The Mogilanska Tumulus Ritual Knemida, The Unknown Face of the Artwork, *İstanbul Kültür University Publication No 180*, 2012, p. 103-107.
- RANULFI, A. C.; Utilização de técnicas espectroscópicas no estudo e caracterização de doenças em citros: HLB (greening) e cancro cítrico. Dissertação de mestrado. Instituto de Física de São Carlos, 2014A.

RANULFI, A. C.; BEBEACHIBULI, A. ; ROMANO, R. A. ; FERREIRA, E. J. ; VILLAS-BOAS, P. R. ; MILORI, D. M. B. P.; Avaliação da técnica de espectroscopia de emissão óptica com plasma induzido por laser (LIBS) no estudo de doenças em citros: HLB (greening) e cancro cítrico. In: Simpósio Nacional de Instrumentação Agropecuária, 2014, São Carlos. Anais do SIAGRO Ciência, Inovação e Mercado 2014, 2014B.

KIGRE INC.; MK-367 Datasheet, 2009. Disponível em:
<http://www.kigre.com/products/mk367_datasheet.pdf>. Acesso em: 3 fev. 2017.

STELLARNET INC.; Porta-LIBS-2000 and Plasma Monitor Configurations, 2015.
Disponível em <<http://www.stellarnet.us/systems/porta-libs-2000-and-plasma-monitor-configurations/>>. Acesso em: 3 fev. 2017.

SILVA, K. S. G. ; FRAGALLI, A. ; CAMPOS, M. ; VILLAS BOAS, P. R. ; BECKER, M. ; CRESTANA, S. ; MILORI, D. M. B. P. . Desenvolvimento de sistema para controle remoto de um LIBS portátil. In: Simpósio Nacional de Instrumentação Agropecuária, 2014, São Carlos. Anais do Simpósio Nacional de Instrumentação Agropecuária 2014, 2014.

REHM, B.; Estrella device driver for the StellarNet EPP2000 series of spectrometers, 2013.
Disponível em: <<https://github.com/shugaa/estrella>>. Acesso em: 3 fev. 2017.

KESHAVA, N.; Distance Metrics and Band Selection in Hyperspectral Processing With Applications to Material Identification and Spectral Libraries. IEEE Transactions on Geoscience and Remote Sensing, Volume: 42, Issue: 7, 2004, p. 1552 – 1565.

APÊNDICE A – CÓDIGO FONTE DA INTERFACE MODO TEXTO

Arquivo pylibsanalyser-1.18.py

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
#
import sys; sys.dont_write_bytecode = True # não gera arquivos pyc
#
# pylibsanalyser.py
#
# Copyright 2017 Kleydson Stenio
#
# This program is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2 of the License, or
# (at your option) any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this program; if not, write to the Free Software
# Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston,
# MA 02110-1301, USA.
#

# variáveis globais - pastas utilizadas pelo programa (troque caso deseje
mudar os nomes)
global inputs, outputs, spectra, FLAG
inputs, outputs, spectra = 'inputs','outputs','spectra'
outer_data = 'inputs.xls'

# variáveis para a importação de picos
'''
delimiter_s -> tipo de delimitador espectro LIBS
rows -> total de linhas a serem puladas do espectro (cabeçalho)
outlier_c -> critério de exclusão de outliers
type_fillpeaks -> define se usará o fp1 (todo espectro) ou o fp2 (por
espectrômetro)
bool_outliers -> booleano que indica se será feita ou não a remoção de
outliers
bool_savgol -> booleano que indica se será passado ou não o filtro
Savitzky-Golay
bool_fillpeaks -> booleano que indica se será passado ou não o filtro 4S
Peak Filling
bool_graph -> booleano que define se será ou salvo o espectro total da
amostras
'''
delimiter = '\t'
rows = 8
outlier_c = 0.9916
type_fillpeaks = ['fp1',0]
bool_outlier = True
bool_savgol = False
bool_fillpeaks = True
bool_graph = True
```

```

# variáveis para isolamento de picos
'''
elements -> lista com o(s) elemento(s) a serem isolados
x_lower -> lista com o(s) comprimento(s) de onda do início do(s) pico(s)
x_upper -> lista com o(s) comprimento(s) de onda do final do(s) pico(s)
bool_area_norm -> booleano que define se será ou não normalizado o pico
pela área da reta
'''
elements = ['C1', 'C2', 'Ti']
x_lower = [192.146, 246, 399.4]
x_upper = [194.481, 249, 400.5]
bool_area_norm = False

# variáveis para fit de picos
alpha = 1.3269 # Lorentz 2 (alterado o range) | 0.842 (Gauss) | 1.187
(Lorentz) | 1.2288 (Bi-Lorentz)
xc = [[193.04, 193.53]]
curve = ['Lorentz']
poly = [0]
fit_graph = True

# variáveis para o modelo
column_number = 5
c_alpha = 0.95
nitrogen = True
bool_graph_model=False

def main(args):
    # define a raiz e a versão do programa
    root, ver = args[0], args[1]

    # faz print do nome do programa
    print('\n Python LIBS Analyser %s\n' %ver)

    # verifica se as pastas existem; caso não, as cria
    FLAG = check_folders(root, inputs, outputs, spectra)
    if FLAG != 0:
        print('\n\n Erro na verificação das pastas.')
        return 1

    # verifica se existe um arquivo de excel com os dados, caso não, usa as
    # variáveis definidas acima
    if os.path.isfile(root+'/' +outer_data) == True:
        data = pandas.read_excel(root+'/' +outer_data, sheetname=0)
        elements = list(data.Elemento)
        peaknumber = list(data.NumPicos)
        xc = list(data.CentroPico)
        x_lower = list(data.RangeInf)
        x_upper = list(data.RangeSup)
        curve = list(data.Curva)
        poly = list(data.Polinomio)

    # pergunta as opções para o usuário
    print('\n O que deseja fazer? \n')
    option = 0
    vals_s = ['Todos os passos', 'Apenas importar espectros', 'Apenas isolar
    picos', 'Apenas fazer fit dos picos', 'Apenas criar modelo', 'Importar e
    isolar', 'Isolar em diante', 'Fit em diante', 'Limpar todas as pastas', 'Fitar
    alternativo (SDV)', 'SAIR']
    vals_n = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]

```

```

vals = pandas.DataFrame({'Opções': vals_s, 'Valores': vals_n})
print(vals.to_string(index=False))
while option <= (vals_n[0]-1) or option > vals_n[-1]:
    bool_option = False
    try:
        option = int(raw_input('\n Insira a opção desejada: '))
    except ValueError:
        print('\n Valor inválido, a opção escolhida deve ser um
inteiro.\n')
        bool_option = True
    if (option <= (vals_n[0]-1) or option > vals_n[-1]) and bool_option
== False:
        print('\n Valor inválido, escolha um número entre %s e %s\n'
%(vals_n[0],vals_n[-1]))

# executa rotina de importação
if option in (1,2,6):
    outputs_clear(root, outputs, 'total')
    outputs_clear(root, outputs, 'pictures')
    outputs_clear(root, outputs, 'mean')
    print('\n Importando:')
    FLAG = importdata(root, outputs, spectra, delimiter, rows,
outlier_c, type_fillpeaks, bool_outlier, bool_savgol, bool_fillpeaks,
bool_graph)
    if FLAG != 0:
        if FLAG == 1:
            print('\n\n Matriz de medidas nula (reduza o critério de
outliers).')
        elif FLAG == 2:
            print('\n\n Mais de 50% dos espectros foram eliminados
(reduza o critério de outliers).')
        print('\n Saindo do programa.\n')
        return 1

# executa rotina de isolamento dos picos
if option in (1,3,6,7):
    outputs_clear(root, outputs, 'isolated')
    print('\n Isolando:')
    FLAG = isodata(root, outputs, x_lower, x_upper, elements,
bool_area_norm)
    if FLAG != 0:
        print('\n\n Erro no isolamento dos picos.')
        return 1

# executa rotina de fit
if option in (1,4,7,8):
    outputs_clear(root, outputs, 'elements')
    print('\n Fitando:')
    FLAG = fitdata(root, outputs, curve, poly, xc, alpha, elements,
peaknumber, fit_graph)
    if FLAG != 0:
        if FLAG == 1:
            print('\n\n Erro no fit dos picos. Tente reexecutar o
isolamento dos picos.')
        elif FLAG == 2:
            print('\n\n Erro no fit. Não é possível utilizar correção
polinomial para o método Trapezoidal.')
        elif FLAG == 3:
            print('\n\n Erro no fit. Não é possível utilizar mais de um
pico para o método Trapezoidal.')
        return 1

```

```

# executa rotina de fit alternativo
if option == 10:
    outputs_clear(root, outputs, 'elements')
    print('\n Fitando de modo alternativo (SDV):')
    FLAG = fitdata_sdv(root, outputs, curve, poly, xc, alpha, elements,
peaknumber, fit_graph)
    if FLAG != 0:
        if FLAG == 1:
            print('\n\n Erro no fit dos picos. Tente reexecutar o
isolamento dos picos.')
        elif FLAG == 2:
            print('\n\n Erro no fit. Não é possível utilizar correção
polinomial para o método Trapezoidal.')
        elif FLAG == 3:
            print('\n\n Erro no fit. Não é possível utilizar mais de um
pico para o método Trapezoidal.')
        return 1

# executa rotina de criação de modelo
if option in (1,5,7,8):
    try:
        data_model = pandas.read_excel(root+'/' +outer_data,sheetname=1)
    except IndexError:
        print('\n\n Erro na criação do modelo. Não existe segunda
planilha no arquivo "inputs.xls".')
        return 1
    outputs_clear(root, outputs, 'model')
    print('\n Criando modelo:')
    FLAG = modeldata(root, outputs, data_model, column_number, c_alpha,
nitrogen, bool_graph_model=False)
    if FLAG != 0:
        print('\n\n Erro na criação do modelo. Tente reexecutar
novamente.')
        return 1

# limpa todas as pastas
if option == 9:
    outputs_clear(root, outputs, 'total')
    outputs_clear(root, outputs, 'pictures')
    outputs_clear(root, outputs, 'mean')
    outputs_clear(root, outputs, 'isolated')
    outputs_clear(root, outputs, 'elements')
    outputs_clear(root, outputs, 'model')

# faz print de uma mensagem de sucesso e finalmente sai do programa
print('\n\n Todas as operações foram executadas com sucesso!')

return 0

if __name__ == '__main__':
    import os, numpy, pandas
    from pyfiles import *
    from glob import glob
    # versão do programa
    ver = glob('*.*.py')
    ver = ver[0].split(' ')[-1].replace('.py','')
    sys.exit(main([os.getcwd(),ver]))

```

APÊNDICE B – CÓDIGO FONTE DA INTERFACE GRÁFICA

Arquivo pylibsanalyser-1.18-GUI.py

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
import sys;
sys.dont_write_bytecode = True # não gera arquivos pyc
import os, pandas, interface, xlwt, xlrd
from pathlib import Path
from pyfiles.fitdata import fitdata
from pyfiles.isodata import isodata
from pyfiles.modeldata import modeldata
from PyQt5 import QtGui, QtWidgets
from pyfiles.importdata import importdata, outputs_clear, check_folders
#
# pylibsanalyser.py
#
# Copyright 2017 Kleydson Stenio <k@Miradesk>
#
# This program is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2 of the License, or
# (at your option) any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this program; if not, write to the Free Software
# Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston,
# MA 02110-1301, USA.
#
global FLAG

def check_value(object):
    if object:
        return object.text()

def click_check_box(bool, object):
    if bool:
        if not object.isChecked():
            object.click()

class App(QtWidgets.QMainWindow, interface.Ui_MainWindow):
    def __init__(self):
        super(self.__class__, self).__init__()
        self.setupUi(self)

        # declara variaveis
        self.inputs = 'inputs'
        self.outputs = 'outputs'
        self.root = os.getcwd()
```

```

# le o arquivo de configuracoes e checa as pastas
check_folders(self.root, self.inputs, self.outputs, spectra='')
self.load_config()

# conecta os sinais com os slots
self.TB_procurarPasta.clicked.connect(self.procura_pasta_espectro)
self.TB_procurarPasta_2.clicked.connect(self.procura_pasta_modelo)
self.B_adicionarLinha.clicked.connect(self.add_linha)
self.B_removerLinha.clicked.connect(self.remove_linha)
self.B_executarImport.clicked.connect(self.importar)
self.B_executarIso.clicked.connect(self.isolar)
self.B_executarFit.clicked.connect(self.fit_picos)
self.B_executarModelo.clicked.connect(self.modelo)

self.CB_outliers.stateChanged.connect(self.change_spin)
self.CB_correcaoC.stateChanged.connect(self.change_alpha)

self.change_spin()
self.change_alpha()

# adiciona Labels no grafico
self.GW_graficoFit.setLabel('left', 'Intensidade Relativa (u.a.)')
self.GW_graficoFit.setLabel('bottom', 'Comprimento de onda (nm)')

self.GW_graficoImport.setLabel('left', 'Intensidade Relativa
(u.a.) ')
self.GW_graficoImport.setLabel('bottom', 'Comprimento de onda
(nm) ')

self.GW_graficoIso.setLabel('left', 'Intensidade Relativa (u.a.)')
self.GW_graficoIso.setLabel('bottom', 'Comprimento de onda (nm)')

def change_spin(self):
    if self.CB_outliers.isChecked():
        self.DSB_criterio.setEnabled(True)
    else:
        self.DSB_criterio.setDisabled(True)

def change_alpha(self):
    if self.CB_correcaoC.isChecked():
        self.LE_valorAlpha.setEnabled(True)
    else:
        self.LE_valorAlpha.setDisabled(True)

def procura_pasta_espectro(self):
    # pede para o usuario escolher uma pasta para salvar o arquivo
    directory = QtGui.QFileDialog.getExistingDirectory(self, "Selecione
uma Pasta")

    if directory: # se o usuario escolher uma pasta: escreve no campo
        self.LE_pastaEspectros.setText(directory)

def procura_pasta_modelo(self):
    # pede para o usuario escolher uma pasta para salvar o arquivo
    directory = QtGui.QFileDialog.getOpenFileName(self, "Selecione um
arquivo")

    if directory: # se o usuario escolher uma pasta: escreve no campo
        self.LE_arquivoModelo.setText(directory[0])

def add_linha(self):

```

```

rowPosition = self.TABLE_iso.rowCount()
self.TABLE_iso.insertRow(rowPosition)
cb = QtWidgets.QComboBox()
cb.setGeometry(QtCore.QRect(480, 60, 91, 31))
cb.setObjectName("Combo_curva")
cb.addItem("Lorentziano")
cb.addItem("Lorentziano c/ pico fixo")
cb.addItem("Lorentziano bifurcado")
cb.addItem("Lorentziano bifurcado c/ pico fixo")
cb.addItem("Lorentziano bifurcado c/ pico fixo e seleção de fator
M")
cb.addItem("Gaussiano")
cb.addItem("Gaussiano c/ pico fixo")
cb.addItem("Trapezoidal")
self.TABLE_iso.setCellWidget(rowPosition, 6, cb)

def remove_linha(self):
rowSelected = self.TABLE_iso.currentRow()
self.TABLE_iso.removeRow(rowSelected)

def save_config(self):
config = dict()
writer = pandas.ExcelWriter('config.xls', engine='xlwt')

# importacao
config['espec_folder'] = self.LE_pastaEspectros.text()
config['delim'] = self.Combo_delim.currentText()
config['skip_rows'] = self.SB_pularC.value()
config['use_out'] = self.CB_outliers.isChecked()
config['value_out'] = self.DSB_criterio.value()
config['golay'] = self.CB_filtroSVG.isChecked()
config['fillpeaks'] = self.CB_fillpeaks.isChecked()

# isolamento and fit
rows = self.TABLE_iso.rowCount()
config['table_rows'] = rows
if rows >= 1:
df = pandas.DataFrame(columns=['element', 'npeaks', 'cpeak',
'lim_sup', 'lim_inf', 'poli', 'curve'])
for i in range(0, rows):
df.loc[i] = ''
df.loc[i]['element'] = check_value(self.TABLE_iso.item(i,
0))
df.loc[i]['npeaks'] = check_value(self.TABLE_iso.item(i,
1))
df.loc[i]['cpeak'] = check_value(self.TABLE_iso.item(i, 2))
df.loc[i]['lim_sup'] = check_value(self.TABLE_iso.item(i,
3))
df.loc[i]['lim_inf'] = check_value(self.TABLE_iso.item(i,
4))
df.loc[i]['poli'] = check_value(self.TABLE_iso.item(i, 5))
df.loc[i]['curve'] = self.TABLE_iso.cellWidget(i,
6).currentText()

# Convert the dataframe to an XlsxWriter Excel object.
df.to_excel(writer, 'table')

# modelo
config['model_file'] = self.LE_arquivoModelo.text()
config['read_column'] = self.LE_colunaLeitura.value()
config['c_correlation'] = self.CB_correcaoC.isChecked()

```

```

config['alpha_value'] = self.LE_valorAlpha.text()
config['n_correlation'] = self.CB_correcaoN.isChecked()

s = pandas.Series(config)
p = s.to_frame(name='values')
p.to_excel(writer, 'config')

writer.save()

def load_config(self):
    config = Path("config.xls")
    if not config.is_file():
        return

    config = pandas.read_excel("config.xls", "config").to_dict()
    config = config['values']

    # importacao
    self.LE_pastaEspectros.setText(config['espec_folder'])
    index = self.Combo_delim.findText(config['delim'],
QtCore.Qt.MatchFixedString)
    if index >= 0:
        self.Combo_delim.setCurrentIndex(index)
    self.SB_pularC.setValue(config['skip_rows'])
    click_check_box(config['use_out'], self.CB_outliers)
    self.DSB_criterio.setValue(config['value_out'])

    click_check_box(config['golay'], self.CB_filtroSVG)
    click_check_box(config['fillpeaks'], self.CB_fillpeaks)

    # isolamento e fit
    rows = config['table_rows']
    if rows >= 1:
        df = pandas.read_excel("config.xls", "table")
        for i in range(0, rows):
            self.add_table_row(i, df.loc[i])

    # modelo
    self.LE_arquivoModelo.setText(config['model_file'])
    self.LE_colunaLeitura.setValue(int(config['read_column']))
    if self.LE_valorAlpha.isEnabled():
        self.LE_valorAlpha.setText(str(config['alpha_value']))
    click_check_box(config['c_correlation'], self.CB_correcaoC)
    click_check_box(config['n_correlation'], self.CB_correcaoN)

def add_table_row(self, index, line):
    self.add_linha()
    for i in range(0, 6):
        item = QtWidgets.QTableWidgetItem(str(line[i]))
        self.TABLE_iso.setItem(index, i, item)
    combo_value = line[6]
    combo = self.TABLE_iso.cellWidget(index, 6)
    index = combo.findText(combo_value, QtCore.Qt.MatchFixedString)
    if index >= 0:
        combo.setCurrentIndex(index)

def get_cell(self, row, collum):
    item = self.TABLE_iso.item(row, collum)
    if item:
        return item.text()
    return ""

```

```

def importar(self):
    # variáveis para a importação de picos
    options = {"TAB": "\t", "VÍRGULA": ",", "PONTO E VÍRGULA": ";"}
    spectra = self.LE_pastaEspectros.text()
    delim = self.Combo_delim.currentText() # delimiter = '\t'
    rows = int(self.SB_pularC.text()) # rows = 8
    outlier_c = float(self.DSB_criterio.text().replace(",", ".")) #
outlier_c = 0.9916

    bool_outlier = self.CB_outliers.isChecked() # bool_outlier = True
    bool_savgol = self.CB_filtroSVG.isChecked() # bool_savgol = False
    bool_fillpeaks = self.CB_fillpeaks.isChecked() # bool_fillpeaks =
True

    type_fillpeaks = ['fp1', 0]

    outputs_clear(self.root, self.outputs, 'total')
    outputs_clear(self.root, self.outputs, 'pictures')
    outputs_clear(self.root, self.outputs, 'mean')

    # print('\n Importando:')
    result = importdata(self.root, self.outputs, spectra,
options[delim], rows, outlier_c, type_fillpeaks,
                        bool_outlier,
                        bool_savgol, bool_fillpeaks, False)
    if isinstance(result, int):
        if result == 0:
            QtGui.QMessageBox.about(self, 'Erro', 'Matriz de medidas
nula (reduza o critério de outliers)')
        elif result == 1:
            QtGui.QMessageBox.about(self, 'Erro',
'Mais de 50% dos espectros foram
eliminados (reduza o critério de outliers).')
            # print('\n Saindo do programa.\n')
            return
    else:
        x = result[:, 0]
        y = numpy.mean(result[:, 1:], 1)
        self.GW_graficoImport.plot(x,y,pen="b")
        QtGui.QMessageBox.about(self, 'Sucesso', 'Importado com
sucesso!')

def isolar(self):
    outputs_clear(self.root, self.outputs, 'isolated')
    # variáveis para isolamento de picos
    bool_area_norm = False
    elements = []
    x_lower = []
    x_upper = []
    rows = self.TABLE_iso.rowCount()
    for i in range(0, rows):
        elements.append(self.TABLE_iso.item(i, 0).text())
        x_lower.append(float(self.TABLE_iso.item(i, 3).text()))
        x_upper.append(float(self.TABLE_iso.item(i, 4).text()))

    result = isodata(self.root, self.outputs, x_lower, x_upper,
elements, bool_area_norm)
    # if result != 0:

```

```

        #      QtGui.QMessageBox.about(self, 'Erro', 'Erro no isolamento dos
picos.')
```

```

        #      return 1

        QtGui.QMessageBox.about(self, 'Sucesso', 'Isolado com sucesso')
        self.GW_graficoIso.plot(result, pen="g")
        self.L_graficoIso.setText("Gráfico de isolamento (Pico de
"+elements[-1]+")")
        self.L_graficoIso.setGeometry(QtCore.QRect(600, 230, 671, 20))

    def fit_picos(self):
        outputs_clear(self.root, self.outputs, 'elements')
        options = {"Lorentziano": "Lorentz",
                  "Lorentziano c/ pico fixo": "Fixo-Lorentz",
                  "Lorentziano bifurcado": "Bi-Lorentz",
                  "Lorentziano bifurcado c/ pico fixo": "Fixo-Bi-Lorentz",
                  "Lorentziano bifurcado c/ pico fixo e seleção de fator
M": "Fixo-BiM-Lorentz",
                  "Gaussiano": "Gauss",
                  "Gaussiano c/ pico fixo": "Fixo-Gauss",
                  "Trapezoidal": "Trapezoidal" }

        elements = []
        n_peaks = []
        xc = list()
        poly = []
        curve = []
        rows = self.TABLE_iso.rowCount()
        for i in range(0, rows):
            elements.append(self.TABLE_iso.item(i, 0).text())
            n_peaks.append(int(self.TABLE_iso.item(i, 1).text()))
            xc.append(float(self.TABLE_iso.item(i, 2).text()))
            poly.append(int(self.TABLE_iso.item(i, 5).text()))
            c = self.TABLE_iso.cellWidget(i, 6).currentText()
            curve.append(options[c])

        alpha = 1.3269

        result = fitdata(self.root, self.outputs, curve, poly, xc, alpha,
elements, n_peaks, False)
        if isinstance(result, int):
            if result == 1:
                QtGui.QMessageBox.about(self, 'Erro', 'Erro no fit dos
picos. Tente reexecutar o isolamento dos picos.')
```

```

            elif result == 2:
                QtGui.QMessageBox.about(self, 'Erro',
                                     'Erro no fit. Não é possível
utilizar correção polinomial para o método Trapezoidal.')
```

```

            elif result == 3:
                QtGui.QMessageBox.about(self, 'Erro',
                                     'Erro no fit. Não é possível
utilizar mais de um pico para o método Trapezoidal.')
```

```

            return 1
        else:
            QtGui.QMessageBox.about(self, 'Sucesso', 'Realizado o fit com
sucesso')

            self.GW_graficoFit.plot(result, pen="r")
            self.L_graficoFrit.setText("Gráfico do Fit (Pico de
"+elements[0]+") ["+c+"]")
```

```

# ---- MODELO ---- #
def modelo(self):
    outputs_clear(self.root, self.outputs, 'model')
    # variáveis para o modelo

    column_number = self.LE_colunaLeitura.value() # column_number = 5
    c_correction = self.CB_correcaoC.isChecked()
    c_alpha = float(self.LE_valorAlpha.text()) # c_alpha = 0 # 0.95
    outer_data = self.LE_arquivoModelo.text() # outer_data =
'inputs.xls'

    nitrogen = self.CB_correcaoC.isChecked()

    try:
        data_model = pandas.read_excel(outer_data, sheetname=1)
    except IndexError:
        QtGui.QMessageBox.about(self, 'Erro',
                                'Erro na criação do modelo. Não existe
segunda planilha no arquivo "inputs.xls".')
        return 1

    a,b,r2, e = modeldata(self.root, self.outputs, data_model,
column_number, c_correction, c_alpha, nitrogen, False)

    self.LE_R.setText(str(round(r2, 3)))
    modelo = "y = "+str(a)+" + "+str(b)+"x"
    self.LE_modelo.setText(modelo)
    QtGui.QMessageBox.about(self, 'Sucesso', 'Realizado o modelo com
sucesso')
    self.L_modelo.setText("Modelo de Calibração de
"+data_model.columns[-1])

    def closeEvent(self, event):
        result = QtGui.QMessageBox.question(self, "Confirmar saída...",
                                            "Tem certeza que deseja sair ?",
                                            QtGui.QMessageBox.Yes |
QtGui.QMessageBox.No)
        event.ignore()
        if result == QtGui.QMessageBox.Yes:
            self.save_config()
            event.accept()

if __name__ == '__main__':
    import sys

    app = QtWidgets.QApplication(sys.argv)
    Form = App()
    Form.show()
    sys.exit(app.exec_())

```

Arquivo interface.py

```

# -*- coding: utf-8 -*-
import sys; sys.dont_write_bytecode = True # não gera arquivos pyc
# Form implementation generated from reading ui file
'pylibsanalyserV2.5.ui'
#
# Created by: PyQt5 UI code generator 5.5.1
#

```

```

# WARNING! All changes made in this file will be lost!
import pyqtgraph as pg
from PyQt5 import QtCore, QtGui, QtWidgets
from pyqtgraph import PlotWidget

class Ui_MainWindow(object):
    def setupUi(self, MainWindow):
        ## Switch to using white background and black foreground
        pg.setConfigOption('background', 'w')
        pg.setConfigOption('foreground', 'k')

        MainWindow.setObjectName("MainWindow")
        MainWindow.resize(1280, 719)
        self.centralwidget = QtWidgets.QWidget(MainWindow)
        self.centralwidget.setObjectName("centralwidget")
        self.L_graficoImport = QtWidgets.QLabel(self.centralwidget)
        self.L_graficoImport.setGeometry(QtCore.QRect(600, 10, 671, 20))
        self.L_graficoImport.setObjectName("L_graficoImport")
        self.Importacao = QtWidgets.QGroupBox(self.centralwidget)
        self.Importacao.setGeometry(QtCore.QRect(10, 10, 581, 191))
        self.Importacao.setStyleSheet("#Importacao { \n"
"    border: 1px solid gray; \n"
"    border-radius: 3px; \n"
"}")

        self.Importacao.setFlat(True)
        self.Importacao.setObjectName("Importacao")
        self.L_pastaEspec = QtWidgets.QLabel(self.Importacao)
        self.L_pastaEspec.setGeometry(QtCore.QRect(20, 40, 121, 31))
        self.L_pastaEspec.setObjectName("L_pastaEspec")
        self.L_delim = QtWidgets.QLabel(self.Importacao)
        self.L_delim.setGeometry(QtCore.QRect(20, 80, 111, 21))
        self.L_delim.setObjectName("L_delim")
        self.LE_pastaEspectros = QtWidgets.QLineEdit(self.Importacao)
        self.LE_pastaEspectros.setGeometry(QtCore.QRect(150, 40, 381, 31))
        self.LE_pastaEspectros.setObjectName("LE_pastaEspectros")
        self.CB_fillpeaks = QtWidgets.QCheckBox(self.Importacao)
        self.CB_fillpeaks.setGeometry(QtCore.QRect(150, 150, 91, 31))
        self.CB_fillpeaks.setObjectName("CB_fillpeaks")
        self.B_executarImport = QtWidgets.QPushButton(self.Importacao)
        self.B_executarImport.setGeometry(QtCore.QRect(500, 160, 71, 29))
        self.B_executarImport.setObjectName("B_executarImport")
        self.TB_procurarPasta = QtWidgets.QToolButton(self.Importacao)
        self.TB_procurarPasta.setGeometry(QtCore.QRect(540, 40, 24, 27))
        self.TB_procurarPasta.setObjectName("TB_procurarPasta")
        self.DSB_criterio = QtWidgets.QDoubleSpinBox(self.Importacao)
        self.DSB_criterio.setGeometry(QtCore.QRect(500, 119, 71, 27))
        self.DSB_criterio.setDecimals(4)
        self.DSB_criterio.setMaximum(1.0)
        self.DSB_criterio.setSingleStep(0.0001)
        self.DSB_criterio.setProperty("value", 1.0)
        self.DSB_criterio.setObjectName("DSB_criterio")
        self.CB_outliers = QtWidgets.QCheckBox(self.Importacao)
        self.CB_outliers.setGeometry(QtCore.QRect(370, 120, 71, 21))
        self.CB_outliers.setObjectName("CB_outliers")
        self.CB_filtroSVG = QtWidgets.QCheckBox(self.Importacao)
        self.CB_filtroSVG.setGeometry(QtCore.QRect(20, 151, 111, 31))
        self.CB_filtroSVG.setObjectName("CB_filtroSVG")
        self.L_pularC = QtWidgets.QLabel(self.Importacao)
        self.L_pularC.setGeometry(QtCore.QRect(20, 115, 99, 21))
        self.L_pularC.setObjectName("L_pularC")
        self.L_criterioOut = QtWidgets.QLabel(self.Importacao)

```

```

self.L_criterioOut.setGeometry(QRect(450, 119, 41, 21))
self.L_criterioOut.setObjectName("L_criterioOut")
self.Combo_delim = QtWidgets.QComboBox(self.Importacao)
self.Combo_delim.setGeometry(QRect(150, 80, 81, 25))
self.Combo_delim.setObjectName("Combo_delim")
self.Combo_delim.addItem("")
self.Combo_delim.addItem("")
self.Combo_delim.addItem("")
self.SB_pularC = QtWidgets.QSpinBox(self.Importacao)
self.SB_pularC.setGeometry(QRect(150, 110, 51, 27))
self.SB_pularC.setObjectName("SB_pularC")
self.L_tratamentoOut = QtWidgets.QLabel(self.Importacao)
self.L_tratamentoOut.setGeometry(QRect(450, 90, 121, 17))
self.L_tratamentoOut.setObjectName("L_tratamentoOut")
self.L_r = QtWidgets.QLabel(self.centralwidget)
self.L_r.setGeometry(QRect(910, 660, 56, 17))
self.L_r.setObjectName("L_r")
self.LE_modelo = QtWidgets.QLineEdit(self.centralwidget)
self.LE_modelo.setGeometry(QRect(600, 680, 291, 27))
self.LE_modelo.setObjectName("LE_modelo")
self.GW_graficoFit = PlotWidget(self.centralwidget)
self.GW_graficoFit.setGeometry(QRect(600, 470, 671, 191))
self.GW_graficoFit.setObjectName("GW_graficoFit")
self.LE_R = QtWidgets.QLineEdit(self.centralwidget)
self.LE_R.setGeometry(QRect(910, 680, 71, 27))
self.LE_R.setObjectName("LE_R")
self.L_graficoFrit = QtWidgets.QLabel(self.centralwidget)
self.L_graficoFrit.setGeometry(QRect(600, 451, 331, 20))
self.L_graficoFrit.setWordWrap(False)
self.L_graficoFrit.setObjectName("L_graficoFrit")
self.L_graficoIso = QtWidgets.QLabel(self.centralwidget)
self.L_graficoIso.setGeometry(QRect(600, 230, 331, 20))
self.L_graficoIso.setObjectName("L_graficoIso")
self.L_modelo = QtWidgets.QLabel(self.centralwidget)
self.L_modelo.setGeometry(QRect(600, 657, 161, 21))
self.L_modelo.setObjectName("L_modelo")
self.Modelo = QtWidgets.QGroupBox(self.centralwidget)
self.Modelo.setGeometry(QRect(10, 540, 581, 171))
self.Modelo.setStyleSheet("#Modelo { \n"
" border: 1px solid gray; \n"
" border-radius: 3px; \n"
" }")

self.Modelo.setObjectName("Modelo")
self.LE_arquivoModelo = QtWidgets.QLineEdit(self.Modelo)
self.LE_arquivoModelo.setGeometry(QRect(120, 40, 401, 31))
self.LE_arquivoModelo.setObjectName("LE_arquivoModelo")
self.LE_valorAlpha = QtWidgets.QLineEdit(self.Modelo)
self.LE_valorAlpha.setGeometry(QRect(500, 80, 61, 29))
self.LE_valorAlpha.setObjectName("LE_valorAlpha")
self.CB_correcaoN = QtWidgets.QCheckBox(self.Modelo)
self.CB_correcaoN.setGeometry(QRect(210, 110, 131, 31))
self.CB_correcaoN.setObjectName("CB_correcaoN")
self.B_executarModelo = QtWidgets.QPushButton(self.Modelo)
self.B_executarModelo.setGeometry(QRect(506, 140, 71, 29))
self.B_executarModelo.setObjectName("B_executarModelo")
self.CB_correcaoC = QtWidgets.QCheckBox(self.Modelo)
self.CB_correcaoC.setGeometry(QRect(210, 80, 161, 31))
self.CB_correcaoC.setObjectName("CB_correcaoC")
self.L_arquivoModelo = QtWidgets.QLabel(self.Modelo)
self.L_arquivoModelo.setGeometry(QRect(10, 40, 101, 31))
self.L_arquivoModelo.setObjectName("L_arquivoModelo")

```

```

self.L_colunaLeitura = QtWidgets.QLabel(self.Modelo)
self.L_colunaLeitura.setGeometry(QtCore.QRect(10, 80, 97, 31))
self.L_colunaLeitura.setObjectName("L_colunaLeitura")
self.L_valorAlpha = QtWidgets.QLabel(self.Modelo)
self.L_valorAlpha.setGeometry(QtCore.QRect(370, 80, 121, 31))
self.L_valorAlpha.setObjectName("L_valorAlpha")
self.TB_procurarPasta_2 = QtWidgets.QToolButton(self.Modelo)
self.TB_procurarPasta_2.setGeometry(QtCore.QRect(540, 40, 24, 27))
self.TB_procurarPasta_2.setObjectName("TB_procurarPasta_2")
self.LE_colunaLeitura = QtWidgets.QSpinBox(self.Modelo)
self.LE_colunaLeitura.setGeometry(QtCore.QRect(120, 80, 51, 27))
self.LE_colunaLeitura.setObjectName("LE_colunaLeitura")
self.Isolamento = QtWidgets.QGroupBox(self.centralwidget)
self.Isolamento.setGeometry(QtCore.QRect(10, 210, 581, 321))
self.Isolamento.setStyleSheet("#Isolamento { \n"
" border: 1px solid gray; \n"
" border-radius: 3px; \n"
" }")

self.Isolamento.setObjectName("Isolamento")
self.B_executarIso = QtWidgets.QPushButton(self.Isolamento)
self.B_executarIso.setGeometry(QtCore.QRect(490, 290, 87, 29))
self.B_executarIso.setObjectName("B_executarIso")
self.TABLE_iso = QtWidgets.QTableWidget(self.Isolamento)
self.TABLE_iso.setGeometry(QtCore.QRect(10, 30, 561, 201))

self.TABLE_iso.setSizeAdjustPolicy(QtWidgets.QAbstractScrollArea.AdjustToCo
ntents)

self.TABLE_iso.setAlternatingRowColors(True)
self.TABLE_iso.setColumnCount(7)
self.TABLE_iso.setObjectName("TABLE_iso")
self.TABLE_iso.setRowCount(0)
item = QtWidgets.QTableWidgetItem()
font = QtGui.QFont()
font.setKerning(True)
item.setFont(font)
self.TABLE_iso.setHorizontalHeaderItem(0, item)
item = QtWidgets.QTableWidgetItem()
self.TABLE_iso.setHorizontalHeaderItem(1, item)
item = QtWidgets.QTableWidgetItem()
self.TABLE_iso.setHorizontalHeaderItem(2, item)
item = QtWidgets.QTableWidgetItem()
self.TABLE_iso.setHorizontalHeaderItem(3, item)
item = QtWidgets.QTableWidgetItem()
self.TABLE_iso.setHorizontalHeaderItem(4, item)
item = QtWidgets.QTableWidgetItem()
self.TABLE_iso.setHorizontalHeaderItem(5, item)
item = QtWidgets.QTableWidgetItem()
self.TABLE_iso.setHorizontalHeaderItem(6, item)
self.TABLE_iso.horizontalHeader().setDefaultSectionSize(93)
self.B_adicionarLinha = QtWidgets.QPushButton(self.Isolamento)
self.B_adicionarLinha.setGeometry(QtCore.QRect(60, 240, 71, 29))
self.B_adicionarLinha.setObjectName("B_adicionarLinha")
self.B_removerLinha = QtWidgets.QPushButton(self.Isolamento)
self.B_removerLinha.setGeometry(QtCore.QRect(130, 240, 71, 29))
self.B_removerLinha.setObjectName("B_removerLinha")
self.B_executarFit = QtWidgets.QPushButton(self.Isolamento)
self.B_executarFit.setGeometry(QtCore.QRect(400, 290, 87, 29))
self.B_executarFit.setObjectName("B_executarFit")
self.L_linha = QtWidgets.QLabel(self.Isolamento)
self.L_linha.setGeometry(QtCore.QRect(10, 240, 41, 31))
self.L_linha.setObjectName("L_linha")

```

```

self.GW_graficoIso = PlotWidget(self.centralwidget, "w")
self.GW_graficoIso.setGeometry(QRect(600, 250, 671, 191))
self.GW_graficoIso.setObjectName("GW_graficoIso")
self.GW_graficoImport = PlotWidget(self.centralwidget, "w")
self.GW_graficoImport.setGeometry(QRect(600, 30, 671, 191))
self.GW_graficoImport.setObjectName("GW_graficoImport")
MainWindow.setCentralWidget(self.centralwidget)

self.retranslateUi(MainWindow)
QtCore.QMetaObject.connectSlotsByName(MainWindow)

def retranslateUi(self, MainWindow):
    _translate = QtCore.QCoreApplication.translate
    MainWindow.setWindowTitle(_translate("MainWindow",
"PyLIBSAnalyser"))
    self.L_graficoImport.setText(_translate("MainWindow", "Gráfico da
Importação"))
    self.Importacao.setTitle(_translate("MainWindow", "Importação"))
    self.L_pastaEspec.setText(_translate("MainWindow", "Pasta com
espectros"))
    self.L_delim.setText(_translate("MainWindow", "Delimitador"))
    self.CB_fillpeaks.setText(_translate("MainWindow", "Fillpeaks"))
    self.B_executarImport.setText(_translate("MainWindow", "Executar"))
    self.TB_procurarPasta.setText(_translate("MainWindow", "..."))
    self.CB_outliers.setText(_translate("MainWindow", "Utilizar"))
    self.CB_filtroSVG.setText(_translate("MainWindow", "Filtro SV
Golay"))
    self.L_pularC.setText(_translate("MainWindow", "Pular Cabeçalho"))
    self.L_criterioOut.setText(_translate("MainWindow", "Critério"))
    self.Combo_delim.setItemText(0, _translate("MainWindow", "TAB"))
    self.Combo_delim.setItemText(1, _translate("MainWindow",
"VÍRGULA"))
    self.Combo_delim.setItemText(2, _translate("MainWindow", "PONTO E
VÍRGULA"))
    self.L_tratamentoOut.setText(_translate("MainWindow", "Tratamento
Outliers"))
    self.L_r.setText(_translate("MainWindow", "R² "))
    self.LE_modelo.setText(_translate("MainWindow", "y = 0.1x+ 0.3
(exemplo)"))
    self.LE_R.setText(_translate("MainWindow", "0.85"))
    self.L_graficoFrit.setText(_translate("MainWindow", "Gráfico de
Fit"))
    self.L_graficoIso.setText(_translate("MainWindow", "Gráfico de
Isolamento"))
    self.L_modelo.setText(_translate("MainWindow", "Modelo"))
    self.Modelo.setTitle(_translate("MainWindow", "Modelo"))
    self.CB_correcaoN.setText(_translate("MainWindow", "Correção por
N"))
    self.B_executarModelo.setText(_translate("MainWindow", "Executar"))
    self.CB_correcaoC.setText(_translate("MainWindow", "Correção de C
por Al"))
    self.L_arquivoModelo.setText(_translate("MainWindow", "Local
arquivo"))
    self.L_colunaLeitura.setText(_translate("MainWindow", "Coluna de
leitura"))
    self.L_valorAlpha.setText(_translate("MainWindow", "Valor do alpha
(C+Al)"))
    self.TB_procurarPasta_2.setText(_translate("MainWindow", "..."))
    self.Isolamento.setTitle(_translate("MainWindow", "Isolamento"))
    self.B_executarIso.setText(_translate("MainWindow", "Executar
ISO"))

```

```
item = self.TABLE_iso.horizontalHeaderItem(0)
item.setText(_translate("MainWindow", "Elemento"))
item = self.TABLE_iso.horizontalHeaderItem(1)
item.setText(_translate("MainWindow", "N°Picos"))
item = self.TABLE_iso.horizontalHeaderItem(2)
item.setText(_translate("MainWindow", "Centro Pico"))
item = self.TABLE_iso.horizontalHeaderItem(3)
item.setText(_translate("MainWindow", "Limite Inferior"))
item = self.TABLE_iso.horizontalHeaderItem(4)
item.setText(_translate("MainWindow", "Limite Superior"))
item = self.TABLE_iso.horizontalHeaderItem(5)
item.setText(_translate("MainWindow", "Polinômio"))
item = self.TABLE_iso.horizontalHeaderItem(6)
item.setText(_translate("MainWindow", "Curva"))
self.B_adicionarLinha.setText(_translate("MainWindow",
"Adicionar"))
self.B_removerLinha.setText(_translate("MainWindow", "Remover"))
self.B_executarFit.setText(_translate("MainWindow", "Executar
FIT"))
self.L_linha.setText(_translate("MainWindow", "Linha:"))
```

APÊNDICE C – CÓDIGO FONTE DO DRIVER PYTHON/LINUX PARA OS ESPECTRÔMETROS STELLARNET EPPUSB2000-LR

Arquivo estrella_python_controller.py

```
#!/usr/bin/env python
#
# Copyright (c) 2017, Kleydson Stenio
# All rights reserved.
#
# Redistribution and use in source and binary forms, with or without
# modification, are permitted provided that the following conditions are
# met:
#
# * Redistributions of source code must retain the above copyright notice,
# this
# list of conditions and the following disclaimer.
# * Redistributions in binary form must reproduce the above copyright
# notice,
# this list of conditions and the following disclaimer in the
# documentation
# and/or other materials provided with the distribution.
# * Neither the name of the author nor the names of its contributors may be
# used to endorse or promote products derived from this software without
# specific prior written permission.
#
# THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS
# IS"
# AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
# IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
# PURPOSE ARE
# DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
# LIABLE
# FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
# CONSEQUENTIAL
# DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
# OR
# SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
# HOWEVER
# CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
# LIABILITY,
# OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE
# USE
# OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
#
#
# *****
#
# Python Controller, main program.
#
from estrella_classes import *
from sys import exit
from os import getcwd
import matplotlib.pyplot as plt

def estrella_controller():
```

```

print '\n','Initializing Estrella Python Driver','\n'

# saves the root of the program
root = getcwd()

# import main libraries to control the spectrometer (change the
location to match yours)
dll, estrella =
import_libraries('/opt/libdll/lib/libdll.so','/opt/estrella/lib/libestrella
.so')

# create x-axis (must pass the location of file with calibration
parameters, change if needed)
wavelength = create_xaxis(root + '/calibration_parameters.txt')

# create the device list and the estrella session
try:
    devices, esession = estrella_begin(dll, estrella)
except TypeError:
    print 'Error detected, exiting.','\n'
    exit(1)

# setup estrella session
setup = estrella_setup(esession, devices, dll, estrella,
ESTR_XSMOOTH_NONE, ESTR_XTMODE_NORMAL, ESTR_TEMPComp_OFF, ESTRELLA_RATE,
ESTR_XRES_HIGH)
if setup != 0:
    print 'Error detected, exiting.','\n'
    exit(1)

# do the normal scan
counts = estrella_nscan(devices, esession, dll, estrella)
if counts == 1:
    print 'Error detected, exiting.','\n'
    exit(1)

# plot the result using Matplotlib
plt.plot(wavelength,counts)
plt.show()

# close the session
estrella_end(devices, esession, dll, estrella)

print '\n','All done without errors.','\n'

return 0

if __name__ == '__main__':
    estrella_controller()

```

Arquivo enumerations.py

```

#!/usr/bin/env python
#
# Copyright (c) 2017, Kleydson Stenio
# All rights reserved.
#
# Redistribution and use in source and binary forms, with or without

```

```

# modification, are permitted provided that the following conditions are
met:
#
# * Redistributions of source code must retain the above copyright notice,
this
# list of conditions and the following disclaimer.
# * Redistributions in binary form must reproduce the above copyright
notice,
# this list of conditions and the following disclaimer in the
documentation
# and/or other materials provided with the distribution.
# * Neither the name of the author nor the names of its contributors may be
# used to endorse or promote products derived from this software without
# specific prior written permission.
#
# THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS
IS"
# AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
# IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE ARE
# DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
LIABLE
# FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
CONSEQUENTIAL
# DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
OR
# SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
HOWEVER
# CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
LIABILITY,
# OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE
USE
# OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
#
#
*****
****
#
# Python Controller, enumerations.
#

from ctypes import c_int

#####
# Enumerations for the libraries in python shape #
#####

# maximum path string length
ESTRELLA_PATH_MAX = c_int(256)

# spectrometer rate
ESTRELLA_RATE = c_int(50)

# library error codes
ESTROK = c_int(0)
ESTRERR = c_int(1)
ESTRINV = c_int(2)
ESTRNOMEM = c_int(3)
ESTRTIMEOUT = c_int(4)
ESTRNOTIMPL = c_int(5)
ESTRALREADY = c_int(6)

```

```

# values for enumeration 'estr_xtmode_t'
ESTR_XTMODE_NORMAL = c_int(0)
ESTR_XTMODE_TRIGGER = c_int(1)
ESTR_XTMODE_TYPES = c_int(2)

# values for enumeration 'estr_xtrate_t'
ESTR_XRES_LOW = c_int(0)
ESTR_XRES_MEDIUM = c_int(1)
ESTR_XRES_HIGH = c_int(2)
ESTR_XRES_TYPES = c_int(3)

# values for enumeration 'estr_xsmooth_t'
ESTR_XSMOOTH_NONE = c_int(0)
ESTR_XSMOOTH_5PX = c_int(1)
ESTR_XSMOOTH_9PX = c_int(2)
ESTR_XSMOOTH_17PX = c_int(3)
ESTR_XSMOOTH_33PX = c_int(4)
ESTR_XSMOOTH_TYPES = c_int(5)

# values for enumeration 'estr_tempcomp_t'
ESTR_TEMPCOMP_OFF = c_int(0)
ESTR_TEMPCOMP_ON = c_int(1)
ESTR_TEMPCOMP_TYPES = c_int(2)

# values for enumeration 'estrella_devicetype_t'
ESTRELLA_DEV_USB = c_int(0)
ESTRELLA_DEV_LPT = c_int(1)

```

Arquivo functions.py

```

#!/usr/bin/env python
#
# Copyright (c) 2017, Kleydson Stenio
# All rights reserved.
#
# Redistribution and use in source and binary forms, with or without
# modification, are permitted provided that the following conditions are
# met:
#
# * Redistributions of source code must retain the above copyright notice,
# this
# list of conditions and the following disclaimer.
# * Redistributions in binary form must reproduce the above copyright
# notice,
# this list of conditions and the following disclaimer in the
# documentation
# and/or other materials provided with the distribution.
# * Neither the name of the author nor the names of its contributors may be
# used to endorse or promote products derived from this software without
# specific prior written permission.
#
# THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS
# IS"
# AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
# IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
# PURPOSE ARE
# DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
# LIABLE

```

```

# FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
CONSEQUENTIAL
# DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
OR
# SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
HOWEVER
# CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
LIABILITY,
# OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE
USE
# OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
#
#
*****
****
#
# Python Controller, needed functions (if you need more, add here).
#

from ctypes import cdll, byref, c_uint, c_int, c_void_p, c_float
from numpy import empty, frombuffer, float32
from pandas import read_csv
from enumerations import *
from structs import *

#####
# Functions for the controller to work properly #
#####

def import_libraries(local_dll, local_estrella):
    libdll = cdll.LoadLibrary(local_dll)
    libestrella = cdll.LoadLibrary(local_estrella)
    return libdll, libestrella

def create_xaxis(parameters_location):
    xaxis = empty(2051)
    calibration = read_csv(parameters_location, sep=' ', skiprows=3)
    for i in range(2051):
        xaxis[i] = (calibration.Parameters[1]/4.0)*(float(i*i)) +
(calibration.Parameters[0]/2.0)*(float(i)) + calibration.Parameters[2]
    return xaxis

def create_yaxis(data):
    yaxis = frombuffer(data, float32)
    return yaxis

def estrella_begin(libdll, libestrella):

    # create a device list for the libraries
    device = c_void_p(None)
    numdevices = c_uint(0)
    devices = dll_list_t()
    libdll.dll_init(byref(devices))

    # create and try to set up an estrella session
    ession = estrella_session_t()
    rc = libestrella.estrella_find_devices(byref(devices))
    if rc != 0:
        print 'It was not possible to look for USB devices.\n(Are you
root?) ', '\n'
        libdll.dll_clear(byref(devices))

```

```

        return 1

rc = libdll.dll_count(byref(devices), byref(numdevices))
if ((rc != 1) or (numdevices == 0)):
    print 'No device found.', '\n'
    libdll.dll_clear(byref(devices))
    return 1

rc = libdll.dll_get(byref(devices), byref(device), None, c_int(0))
if (rc != 1):
    print 'It was not possible to access the EPP2000 device.', '\n'
    libdll.dll_clear(byref(devices))
    return 1

rc = libestrella.estrella_init(byref(ession), device)
if (rc != 0):
    print 'It was not possible to create the session.', '\n'
    libestrella.estrella_close(byref(ession))
    libdll.dll_clear(byref(devices))
    return 1

return devices,ession

def estrella_end(devices, ession, libdll, libestrella):
    libestrella.estrella_close(byref(ession))
    libdll.dll_clear(byref(devices))
    return 0

def estrella_setup(ession, devices, libdll, libestrella, XSMOOTH, XTMODE,
TEMPCOMP, RATE, RESOLUTION):

    rc = libestrella.estrella_update(byref(ession), c_int(1), XSMOOTH,
TEMPCOMP)
    if (rc != 0):
        print 'It was not possible to configure the data.', '\n'
        libestrella.estrella_close(byref(ession))
        libdll.dll_clear(byref(devices))
        return 1

rc = libestrella.estrella_rate(byref(ession), RATE, RESOLUTION);
if (rc != 0):
    print 'It was not possible to configure the rates.', '\n'
    libestrella.estrella_close(byref(ession))
    libdll.dll_clear(byref(devices))
    return 1

rc = libestrella.estrella_mode(byref(ession), XTMODE)
if (rc != 0):
    print 'It was not possible to configure operation mode.', '\n'
    libestrella.estrella_close(byref(ession))
    libdll.dll_clear(byref(devices))
    return 1

return 0

def estrella_nscan(devices, ession, libdll, libestrella):

# create a C like float array and do the normal scan
data = (c_float * 2051)()
rc = libestrella.estrella_scan(byref(ession), byref(data))
if (rc != 0):

```

```

    print 'The scan failed.', '\n'
    libestrella.estrella_close(byref(ession))
    libdll.dll_clear(byref(devices))
    return 1

# convert the scan result into a Numpy array
counts = create_yaxis(data)

return counts

def estrella_ascan(devices, ession, libdll, libestrella):
    rc = libestrella.estrella_async_scan(byref(ession))
    if (rc != 0):
        print 'Async scan failed.', '\n'
        libestrella.estrella_close(byref(ession))
        libdll.dll_clear(byref(devices))
        return 1
    return 0

def estrella_aresult(devices, ession, libdll, libestrella):

    # create a C like float array retrieve the async result
    data = (c_float * 2051) ()
    rc = libestrella.estrella_async_result(byref(ession), byref(data))
    if (rc != 0):
        print 'It was not possible to retrieve the data from scan.', '\n'
        libestrella.estrella_close(byref(ession))
        libdll.dll_clear(byref(devices))
        return 1

    # convert the scan result into a Numpy array
    counts = create_yaxis(data)

    return counts

```

Arquivo structs.py

```

#!/usr/bin/env python
#
# Copyright (c) 2017, Kleydson Stenio
# All rights reserved.
#
# Redistribution and use in source and binary forms, with or without
# modification, are permitted provided that the following conditions are
# met:
#
# * Redistributions of source code must retain the above copyright notice,
#   this
#   list of conditions and the following disclaimer.
# * Redistributions in binary form must reproduce the above copyright
#   notice,
#   this list of conditions and the following disclaimer in the
#   documentation
#   and/or other materials provided with the distribution.
# * Neither the name of the author nor the names of its contributors may be
#   used to endorse or promote products derived from this software without
#   specific prior written permission.
#
# THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS
# IS"

```

```

# AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
# IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE ARE
# DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
LIABLE
# FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
CONSEQUENTIAL
# DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
OR
# SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
HOWEVER
# CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
LIABILITY,
# OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE
USE
# OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
#
#
*****
****
#
# Python Controller, structures.
#

from ctypes import c_ubyte, c_ushort, c_uint, c_int, c_char, c_char_p,
c_void_p, c_size_t, Structure, Union, POINTER

#####
# Specific enumerations for the Classes #
#####

estr_xtmode_t = c_int
estr_xtrate_t = c_int
estr_xsmooth_t = c_int
estr_tempcomp_t = c_int
estrella_devicetype_t = c_int
estr_lock_t = c_int

#####
# Structs for LIBDLL in Classes (python shape) #
#####

class dll_item(Structure):
    pass
dll_item._fields_=[("data",c_void_p),
                   ("prev",POINTER(POINTER(dll_item))),
                   ("next",POINTER(POINTER(dll_item))),
                   ("datasize",c_size_t)]

class dll_list_t(Structure):
    _fields_ = [("count",c_uint),
               ("first",POINTER(POINTER(dll_item))),
               ("last",POINTER(POINTER(dll_item)))]

#####
# Structs for ESTRELLA in Classes (python shape) #
#####

class estrella_usbdev_t(Structure):
    pass
estrella_usbdev_t._fields_ = [('bus', c_char * 256),

```

```

        ('devnum', c_ubyte),
        ('vendorid', c_ushort),
        ('productid', c_ushort),
        ('manufacturer', c_char * 128),
        ('product', c_char * 128),
        ('serialnumber', c_char * 32),]

class estrella_dev_t_u(Union):
    pass
estrella_dev_t_u._fields_ = [('usb', estrella_usbdev_t)]

class estrella_dev_t(Structure):
    pass
estrella_dev_t._fields_ = [('devicetype', estrella_devicetype_t),
                           ('spec', estrella_dev_t_u)]

class usb_dev_handle(Structure):
    pass
usb_dev_handle._fields_ = []

class estrella_session_t_u(Union):
    pass
estrella_session_t_u._fields_ = [('usb_dev_handle',
                                  POINTER(usb_dev_handle))]

class estrella_session_t(Structure):
    pass
estrella_session_t._fields_ = [('rate', c_int),
                                ('scanstoavg', c_int),
                                ('xtmode', estr_xtmode_t),
                                ('xtrate', estr_xtrate_t),
                                ('xsmooth', estr_xsmooth_t),
                                ('tempcomp', estr_tempcomp_t),
                                ('dev', estrella_dev_t),
                                ('spec', estrella_session_t_u),
                                ('lock', estr_lock_t)]

#####
# Structs for ESTRELLA USB Classes (python shape) #
#####

class estrella_usb_request_t(Structure):
    _fields_ = [("requesttype",c_uint),
                ("request",c_uint),
                ("value",c_uint),
                ("index",c_uint),
                ("size",c_uint),
                ("data",c_char_p)]

```

Arquivo _init.py

```

#!/usr/bin/env python
#
# Copyright (c) 2017, Kleydson Stenio
# All rights reserved.
#
# Redistribution and use in source and binary forms, with or without
# modification, are permitted provided that the following conditions are
# met:
#

```

```
# * Redistributions of source code must retain the above copyright notice,
this
# list of conditions and the following disclaimer.
# * Redistributions in binary form must reproduce the above copyright
notice,
# this list of conditions and the following disclaimer in the
documentation
# and/or other materials provided with the distribution.
# * Neither the name of the author nor the names of its contributors may be
# used to endorse or promote products derived from this software without
# specific prior written permission.
#
# THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS
IS"
# AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
# IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE ARE
# DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
LIABLE
# FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
CONSEQUENTIAL
# DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
OR
# SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
HOWEVER
# CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
LIABILITY,
# OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE
USE
# OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
#
#
*****
****
#
# Python Controller, estrella_classes init file.
#

from enumerations import *
from functions import *
from structs import *
```