

**MINISTÉRIO DA DEFESA  
EXÉRCITO BRASILEIRO  
DEPARTAMENTO DE CIÊNCIA E TECNOLOGIA  
INSTITUTO MILITAR DE ENGENHARIA  
( Real Academia de Artilharia, Fortificação e Desenho, 1792 )  
CURSO DE GRADUAÇÃO EM ENGENHARIA CARTOGRÁFICA**

**HUMBERTO DIEGO AGUIAR ALVES**

**IMPLEMENTAÇÃO DE SISTEMA PARA CÁLCULOS  
GEODÉSICOS EM AMBIENTE COMPUTACIONAL LIVRE:  
AJUSTAMENTO DE REDE DE NIVELAMENTO GEOMÉTRICO**

**Rio de Janeiro**

**2016**

**INSTITUTO MILITAR DE ENGENHARIA**

**CAP HUMBERTO DIEGO AGUIAR ALVES**

**IMPLEMENTAÇÃO DE SISTEMA PARA CÁLCULOS  
GEODÉSICOS EM AMBIENTE COMPUTACIONAL LIVRE:  
AJUSTAMENTO DE REDE DE NIVELAMENTO GEOMÉTRICO**

Projeto de Fim de Curso (PFC) apresentado ao Curso de Graduação em Engenharia Cartográfica do Instituto Militar de Engenharia, para a obtenção de grau na Verificação Especial (VF) de 2016.

Orientadores:

Prof. Leonardo Castro de Oliveira

Prof<sup>a</sup>. Heloísa Alves Silva Marques

**Rio de Janeiro**

**2016**

**INSTITUTO MILITAR DE ENGENHARIA**

**CAP HUMBERTO DIEGO AGUIAR ALVES**

**IMPLEMENTAÇÃO DE SISTEMA PARA CÁLCULOS  
GEODÉSICOS EM AMBIENTE COMPUTACIONAL LIVRE:  
AJUSTAMENTO DE REDE DE NIVELAMENTO GEOMÉTRICO**

Projeto de Fim de Curso apresentado ao Curso de Graduação em Engenharia Cartográfica do Instituto Militar de Engenharia, como requisito parcial para a obtenção de grau na Verificação Especial (VF).

Orientadores: Prof. Leonardo Castro de Oliveira  
Prof<sup>a</sup>. Heloísa Alves Silva Marques

Avaliado em 2016 pela seguinte Banca Examinadora:

---

Leonardo Castro de Oliveira D.E.

---

Luiz Felipe Coutinho Ferreira da Silva D.E.

---

Francisco Roberto da Rocha Gomes Ms.C.

**Rio de Janeiro**

**2016**

# INSTITUTO MILITAR DE ENGENHARIA

Praça General Tibúrcio, 80 – Praia Vermelha

Rio de Janeiro – RJ CEP: 22290-270

Este exemplar é de propriedade do Instituto Militar de Engenharia, que poderá incluí-lo em base de dados, armazenar em computador, microfilmear ou adotar qualquer forma de arquivamento.

É permitida a menção, reprodução parcial ou integral e a transmissão entre bibliotecas deste trabalho, sem modificação de seu texto, em qualquer meio que esteja ou venha a ser fixado, para pesquisa acadêmica, comentários e citações, desde que sem finalidade comercial e que seja feita a referência bibliográfica completa.

Os conceitos expressos neste trabalho são de responsabilidade do(s) autor(es) e do(s) orientador(es).

XXX	Alves, Humberto Diego Aguiar
XXXX	<p>IMPLEMENTAÇÃO DE SISTEMA PARA CÁLCULOS GEODÉSICOS EM AMBIENTE COMPUTACIONAL LIVRE: AJUSTAMENTO DE REDE DE NIVELAMENTO GEOMÉTRICO/ Humberto Diego Aguiar Alves; orientado por Leonardo Castro de Oliveira e Heloísa Alves Silva Marques– Rio de Janeiro: Instituto Militar de Engenharia, 2016.</p> <p>65p. : il.</p> <p>Projeto de Fim de Curso (PROFIC) – Instituto Militar de Engenharia, Rio de Janeiro, 2016.</p> <p>1. Curso de Engenharia Cartográfica – Projeto de Fim de Curso. 2. Ajustamento altimétrico. 3. Implementação. 4. Ambientes computacionais livres. I. Oliveira, Leonardo Castro; Marques, Heloísa Alves Silva. II. Título. III. Instituto Militar de Engenharia.</p>

# Sumário

<b>SUMÁRIO .....</b>	<b>5</b>
<b>LISTA DE ILUSTRAÇÕES.....</b>	<b>7</b>
<b>LISTA DE TABELAS .....</b>	<b>8</b>
<b>LISTA DE ABREVIATURAS E SIGLAS .....</b>	<b>9</b>
<b>1. INTRODUÇÃO .....</b>	<b>12</b>
1.1 CONTEXTUALIZAÇÃO .....	12
1.2 OBJETIVO.....	14
1.3 JUSTIFICATIVA.....	14
1.4 ORGANIZAÇÃO .....	15
<b>2. FUNDAMENTAÇÃO TEÓRICA.....</b>	<b>16</b>
2.1 AJUSTAMENTO MMQ DE REDES ALTIMÉTRICAS .....	16
2.2 CONTROLE DE QUALIDADE DO AJUSTAMENTO .....	17
2.2.1 <i>Análise global do ajustamento (teste Chi quadrado)</i> .....	19
2.2.2 <i>Identificação local de erros grosseiros (teste Baarda)</i> .....	20
2.3 PYTHON ADAPTADO A AJUSTAMENTO.....	21
2.4 DESENVOLVIMENTO WEB.....	25
<b>3. APRESENTAÇÃO DA PLATAFORMA .....</b>	<b>28</b>
3.1 ENTRADA DE DADOS .....	28
3.2 FLUXOGRAMA DO PROCESSAMENTO .....	30
3.3 INTERFACE COM O USUÁRIO .....	32
3.4 RELATÓRIO ON-LINE .....	34
3.5 MENSAGENS DE ERRO .....	37
3.6 AJUDA .....	38
3.7 HOSPEDAGEM.....	40
3.8 O PRODUTO NAS FERRAMENTAS DE BUSCA .....	40
<b>4. ESPECIFICAÇÕES TÉCNICAS .....</b>	<b>42</b>
4.1 DESCRIÇÃO DO PRODUTO.....	42
4.2 QUALIDADE DOS DADOS DO PRODUTO.....	43
<b>5. CONCLUSÃO.....</b>	<b>44</b>
<b>REFERÊNCIAS BIBLIOGRÁFICAS .....</b>	<b>45</b>
<b>APÊNDICE A - IBGE, QUADRO (VI), ESPECIFICAÇÕES PARA NIVELAMENTO GEOMÉTRICO .....</b>	<b>46</b>
<b>APÊNDICE B - ELEMENTOS DA LINGUAGEM PYTHON .....</b>	<b>47</b>
<b>APÊNDICE C - EXEMPLO DE AJUSTAMENTO MMQ .....</b>	<b>50</b>
<b>APÊNDICE D - EXEMPLO DE REDE DE NIVELAMENTO – FONTE: GEMAE 2009 .....</b>	<b>52</b>
<b>APÊNDICE E - EXEMPLO DE MODELAGEM DO PESO DAS OBSERVAÇÕES DE DESNÍVEL.....</b>	<b>54</b>

APÊNDICE F - INTRODUÇÃO AOS MÓDULOS MATH, NUMPY, PANDAS E MATPLOTLIB.PYPILOT DA LINGUAGEM PYTHON. ....	56
APÊNDICE G - QUESTÕES PARA EXEMPLOS E DEMONSTRAÇÕES .....	62
APÊNDICE H – DEMONSTRAÇÃO DA EQUAÇÃO DE SOLUÇÃO MMQ PARA PROBLEMAS DE ALTIMETRIA .....	63

## LISTA DE ILUSTRAÇÕES

Figura 1 - Representação gráfica da aceitação no teste Chi quadrado .....	20
Figura 2 - Exemplo de código Python incorporado. ....	26
Figura 3 - Ilustração do funcionamento model-view-controller .....	27
Figura 4 - Exemplo de arquivo texto aceito na plataforma .....	29
Figura 5 - Fluxograma do processamento em Python .....	31
Figura 6 - Página de execução do ajustamento altimétrico.....	32
Figura 7 -Tabela das observações ajustadas .....	35
Figura 8 - Tabela dos parâmetros ajustados.....	35
Figura 9 - Resultado do teste chi quadrado .....	36
Figura 10 - Resultado do teste Baarda .....	36
Figura 11 - Mensagens de erro dentro do formulário.....	37
Figura 12 - Mensagem de erro amigável .....	38
Figura 13 - Ajuda da ferramenta .....	39
Figura 14 - Sitemap em formato .txt.....	41
Figura 15 - Informando o Google Search engine sobre o seu sitemap .....	41

## LISTA DE TABELAS

Tabela 1 - Classificação dos levantamentos geodésicos de Altimetria .....	13
Tabela 2 - Comparação de modelagens de problemas .....	16
Tabela 3 - Exemplos de Interpretadores Python .....	22
Tabela 4 - Módulos externos utilizados em ajustamento .....	22
Tabela 5 - IDE's bastante utilizados em Python .....	24
Tabela 6 - Tempo de execução do ajustamento .....	43



## LISTA DE ABREVIATURAS E SIGLAS

IBGE	Instituto Brasileiro de Geografia e Estatística
SRC	Sistema de Referência de Coordenadas
MMQ	Método dos Mínimos Quadrados
HTML	<i>HyperText Markup Language</i>
MVC	Matriz Variância Covariância
PSF	<i>Python Software Foundation</i>
BSD	<i>Berkeley Software Distribution</i>
EPL	<i>Eclipse Public License</i>
GPL	<i>General Public Licence</i>
LGPL	<i>Lesser General Public Licence</i>
IDE	<i>Integrated Development Environment</i>
MIT	<i>Massachusetts Institute of Technology</i>
DLL	<i>Dynamic-link library</i>

## RESUMO

Nenhuma medição é exata. Devido a necessidade de se realizar uma grande quantidade de medições para se obter uma média representativa do real valor da grandeza observada, que o ajustamento é uma ferramenta importante entre os pesquisadores (especialistas em medições). Especialmente o ajustamento por mínimos quadrados (MMQ) vem ganhando espaço por ser o método de análise e ajustamento de dados mais rigoroso devido a sua abordagem probabilística dos erros, a sua capacidade de considerar a ponderação de cada observação e ainda pelo rompimento da barreira do custo computacional elevado. Neste trabalho será aplicada a modelagem paramétrica do MMQ para resolver problemas de ajustamento de redes altimétricas, bem como os testes de qualidade para detecção e identificação de erros. Sob a premissa de apenas se utilizar de ambientes computacionais livres, o produto usa a linguagem Python para realização dos cálculos, o webframework de código aberto Web2py para incorporação dos resultados obtidos em Hyper Text Markup Language- HTML, além de módulos Python abertos de uso científico, incluirá o software, exemplos, documentação de apoio e a metainformação, tudo estará disponível on-line e gratuito.

Palavras Chave: python, software livre, sistema web, medidas, ajustamento, mínimos quadrados, nivelamento.

## **ABSTRACT**

No measure is exact. Due to the necessity of making a large number of measurements to obtain a representative mean of the real value of a parameter, the adjustment is an important tool among surveyors (measurement specialists). Especially the adjustment by least squares method has been gaining space because it is the method of analysis and adjustment more rigorous due to its probabilistic approach of errors, its capacity of considering proper observations weighting and still by rupture with the high cost computational barrier. This work will apply the parametric modeling of least squares method to solve adjustment of leveling problems, as well as the goodness fit test for detection and identification of errors. Under the premise of only use free computing environments the product uses the Python language for the calculations, the opensource webframework Web2py for embedding the results in HTML, besides of open Python modules for scientific uses, will include the software, examples, support documentation and the meta-information, all of this will be available online and for free.

Key words: python, open-source, web platform, measure, adjustment, least squares, leveling.

# 1. INTRODUÇÃO

## 1.1 Contextualização

A origem das informações utilizadas nos diversos ramos da atividade humana é a coleta de dados. Esta, envolta de problemas por vezes grosseiros, por vezes inerentes é metodicamente regulada em benefício daquela. Além de não existir medição exata, erros de falhas humanas, modelagem inadequada do fenômeno e ainda erros de razões completamente aleatórias permeiam qualquer observação com os mais precisos instrumentos.

A solução básica para encontrar um valor aceitável para uma medição é fazer um grande número de observações, assumir que essas observações possuem uma distribuição normal de probabilidade e tomar a média como o valor mais provável.

Quando se aumenta a complexidade do problema, introduzindo novas observações e parâmetros que também devem ser ajustados, a “média” passa a assumir uma forma mais genérica, procurou-se como valor representativo um valor tal que o somatório dos erros de cada observação fosse mínimo. Este valor não é o mais adequado haja vista que os erros podem ser para mais ou para menos (positivos e negativos) e o somatório poderia ser nulo mesmo com erros altíssimos. Assim, o mais correto é aceitar como representativos os valores tais que o somatório dos erros ao quadrado seja mínimo e é justamente por essa abordagem que o ajustamento pelo Método dos Mínimos Quadrados (MMQ) segue.

A aceitação ou não dos resultados do ajustamento é apoiada, fundamentalmente, na ordem de grandeza dos resíduos finais em relação às necessidades de precisão do trabalho como um todo. Outros indicadores utilizados são as inferências estatísticas do teste de hipóteses entre as variâncias unitárias *priori* e *posteriori* (teste  $\chi^2$ ) e as inferências do teste de hipóteses da correção normalizada (Baarda).

Nos casos de ajustamento de redes altimétricas, deve-se considerar as Normas Gerais para levantamentos geodésicos do IBGE que especifica protocolos na fase de coleta de dados, divididos de acordo com a precisão do

nivelamento. Um extrato do Quadro (I) - Classificação dos levantamentos geodésicos se encontra na Tabela abaixo.

**Tabela 1 - Classificação dos levantamentos geodésicos de Altimetria**

ALTIMETRIA					
EXATIDÃO	Conforme as aplicações, sendo julgada caso a caso, mas devendo o erro padrão ser inferior a $2\text{mm} \sqrt{k}$ para cada duas RN após o ajustamento.	Melhor que $2\text{mm} \sqrt{k}$	Melhor que $3\text{mm} \sqrt{k}$	Melhor que $4\text{mm} \sqrt{k}$	Melhor que $6\text{mm} \sqrt{k}$
DESENVOLVIMENTO	A estrutura será desenvolvida caso a caso de acordo com as finalidades de cada projeto. Basicamente em circuitos e acompanhada de medições gravimétricas (nivelamento geopotencial).	Em circuitos com até 400km de perímetro e estações materializadas, afastadas de no máximo 3 km. Nas áreas metropolitanas dar-se-á preferência ao desenvolvimento em circuitos, em função da urbanização, com estações materializadas e espaçadas de, preferencialmente, 1 km.	Em circuitos com até 200km de perímetro e estações materializadas, afastadas de no máximo 3 km. Nas áreas metropolitanas dar-se-á preferência ao desenvolvimento em circuitos, com estações materializadas e afastadas de, preferencialmente, 1 km.	Em circuitos ou linhas, em função da área a ser atendida, com estações espaçadas de, no máximo, 3 km.	Em circuitos ou linhas, em função dos objetivos a serem atingidos pelos trabalhos.
EXEMPLOS DE UTILIZAÇÃO	Avaliação de movimentos da crosta terrestre; conexões de Sistemas Geodésicos; estudos e definição de parâmetros para os Sistemas Geodésicos; determinação de valores geopotenciais.	Elaboração de cartas gerais; apoio e controle das obras de engenharia e estudos científicos em geral.	Elaboração de cartas gerais; controle de obras de engenharia.	Elaboração de cartas gerais; controle de obras de engenharia.	Levantamentos e parcelamentos de áreas de pequeno valor; pequenas obras; estudos de drenagem e gradientes em áreas de topografia movimentada; elaboração de cartas gerais.
	CIENTÍFICO	FUNDAMENTAL (OU DE 1ª ORDEM)	PARA ÁREAS MAIS DESENVOLVIDAS (OU DE 2ª ORDEM)	PARA ÁREAS MENOS DESENVOLVIDAS (OU DE 3ª ORDEM)	LOCAL
DE ALTA PRECISÃO ÂMBITO NACIONAL			DE PRECISÃO ÂMBITO REGIONAL		PARA FINS TOPOGRÁFICOS

Fonte: Especificações e Normas Gerais para Levantamentos Geodésicos em território brasileiro - IBGE

Esse conjunto de regras impactam diretamente sobre o ajustamento, tornando-o aceitável para os diversos métodos de detecção de erros na medida que a padronização da prática elimina os erros grosseiros, sendo assim um fator de planejamento altamente relevante para qualquer interessado em realizar trabalhos de levantamentos altimétricos (nivelamento) e inclusive levantamentos gravimétricos e planimétricos.

## **1.2 Objetivo**

O objetivo é implementar uma plataforma na linguagem Python, no modo on-line, que realize ajustamento por MMQ, empregando o modelo paramétrico, de redes de NIVELAMENTO.

## **1.3 Justificativa**

Esse projeto visa fornecer a ferramenta de ajustamento altimétrico on-line do IME, assim como outras instituições já o fazem para diversos segmentos das geociências e cartografia.

Um exemplo é o software de ajustamento paramétrico de redes altimétricas Ajunível. Feito no ambiente Builder C++ e DLLs do Fortran da Universidade Estadual Paulista (MARQUES; MONICO, 2005)

No entanto, a escolha do software ser on-line e não de ser executável como o Ajunível, tem vantagens relacionadas ao acompanhamento do uso da ferramenta. Descobrir quem está utilizando a ferramenta, se os fins destes usuários são profissionais ou educativos, etc. Pode-se gerar relatórios de usos para o desenvolvedor e a visualização das oportunidades de melhoria se torna mais fácil.

Esta tendência de prover ferramentas gratuitas on-line é muito comum nas instituições de ensino tanto nacionais quanto internacionais, sendo uma forma eficaz e eficiente dos acadêmicos prestarem contas à comunidade e ainda interagir com ela.

Corrigir os *bugs* fica mais fácil, uma vez que as edições são feitas diretamente nos arquivos instalados no servidor, assim, em qualquer hora e local que tenha acesso à rede, inclusive em qualquer dispositivo, não só de um notebook que tenha o compilador Python instalado, pode-se editar a ferramenta.

## 1.4 Organização

No segundo capítulo será apresentado os fundamentos do MMQ, do método paramétrico e do teste de hipóteses com atenção especial para ajustamentos altimétricos.

Ainda no capítulo 2 deste trabalho, uma explicação geral sobre Python. Introdução às licenças se faz necessária para a escolha dos módulos (bibliotecas) a serem utilizados, uma vez que não se deseja vínculos comerciais com seus respectivos autores. No caso, referenciar o uso dos módulos no corpo do trabalho é a única forma de vínculo possível.

A explicação dos módulos Python utilizados (tratamento de matrizes, operação matricial, construção de gráficos e fórmulas estatísticas) e a explicação específica de suas funções (métodos) mais importantes para o ajustamento e como eles se encaixam nesse trabalho será abordado no capítulo 3.

A parte sobre a construção de sites abordando domínio, hospedagem, webframeworks, HTML básico e código Python *embedded* (incorporado) em HTML será abordado de maneira prática também no capítulo 3.

O capítulo 4 é a apresentação do produto em si, com suas possibilidades, características e resultados para dados simulados.

## 2. FUNDAMENTAÇÃO TEÓRICA

### 2.1 Ajustamento MMQ de redes altimétricas

Problemas de ajustamento altimétrico podem ser resolvidos tanto pela modelagem das equações de condição quanto pela paramétrica. Porém, esta é mais direta e atende ao ajustamento tanto dos parâmetros quanto das observações, enquanto aquela ajusta apenas observações e necessita de uma pré análise do usuário para montar suas matrizes necessárias ao ajustamento, o que demanda um desenvolvimento complexo se for deixado por conta de algum algoritmo. O método condicional força um trabalho extra desnecessário uma vez que o resultado do ajustamento por ambos é o mesmo.

Na abordagem condicionada do problema, condições são forçadas às observações e aos resíduos das mesmas. Na modelagem funcional paramétrica as observações são expressas em função de parâmetros que são desconhecidos e não são medidos diretamente, conforme a tabela a seguir:

**Tabela 2 - Comparação de modelagens de problemas**

Abordagem condicional	Abordagem paramétrica
$f(La) = f(Lb + V) = B.V + W = 0$	$f(Xa) = A.Xa = La = Lb + V$

A solução para problemas de altimetria se dá pela equação abaixo com a demonstração no APÊNDICE H - Demonstração da equação de solução MMQ para problemas de altimetria.

$$Xa = (A^T.P.A)^{-1}.A^T.P.Lb \quad (1)$$

Onde, Xa é o vetor de incógnitas ajustado, Lb é o vetor de observações acrescentado dos termos independentes do sistema, A é a matriz dos coeficientes e P é a matriz da modelagem dos pesos.

Conforme GHILANI & WOLF (2007) as vantagens da abordagem MMQ sobre outros métodos é:

- O ajustamento mais rigoroso;
- Pode ser aplicado mais facilmente;
- Possibilita rigorosa análise pós-ajustamento;



- Pode ser usada no planejamento de coleta de dados;
- O ajustamento mais rigorosamente embasado na teoria matemática da probabilidade

O vetor de parâmetros ajustados no MMQ pode ser encarado como a média generalizada (GEMAEL 2009), ou seja, se o modelo funcional --conjunto de equações que define as condições de ajustamento -- for adequado, os erros assumem uma distribuição Normal de probabilidade em que a média é o valor mais provável (pico), o que implica dizer que:

*O MMQ conduz aos parâmetros mais prováveis, que por sua vez conduz às observações mais prováveis.  
(GHILANI & WOLF, 2007)*

Exemplos de ajustamento por MMQ genérico e outro de rede altimétrica constam nos APÊNDICES C e D respectivamente.

## 2.2 Controle de qualidade do ajustamento

Em busca de estatísticas que acusem a qualidade do ajustamento, tem-se a necessidade de obter as matrizes de variância e covariância de  $Xa$ ,  $La$  e  $V$ . Dessas matrizes podem-se extrair os respectivos desvios-padrões de cada parâmetro ajustado, cada observação ajustada e cada resíduo bastando tirar a raiz quadrada dos elementos da diagonal.

A MVC dos parâmetros ajustados é proveniente da aplicação da lei de propagação da covariância em:

$$Xa = (A^T \cdot P \cdot A)^{-1} \cdot A^T \cdot P \cdot Lb \quad (2)$$

Resultando em:

$$\Sigma_{Xa} = \sigma_0^2 (A^T \cdot P \cdot A)^{-1} \quad (3)$$

A MVC das observações ajustadas é proveniente da aplicação da lei de propagação da variância e covariância em:

$$La = A \cdot Xa \quad (4)$$

Resultando em:

$$\Sigma_{La} = A \cdot (\Sigma_{Xa}) \cdot A^T \quad (5)$$

A MVC dos resíduos é proveniente da aplicação do operador MVC nos 2 lados da equação (1) referente a V:

$$V = La - Lb \quad (6)$$

$$\Sigma_V = \Sigma_{La} - \Sigma_{Lb} \quad (7)$$

Interessante notar como a escolha da variância a priori influencia nas MVCs de  $Xa$ ,  $La$  e  $V$  bem como qualquer fator multiplicativo  $k$  que seja incluído na modelagem de  $\Sigma_{Lb}$ .

A modelagem de  $\Sigma_{Lb}$  pode ser apoiada nas normas gerais para os levantamentos geodésicos do IBGE que apresenta que a qualidade dos trabalhos deverá ser controlada através das diferenças entre o nivelamento e o contranivelamento, observando-se os valores limites de:

- $3mm \times \sqrt{k}$  , para os levantamentos de alta precisão;
- $6mm \times \sqrt{k}$  , para os de precisão em áreas mais desenvolvidas;
- $8mm \times \sqrt{k}$  , para os de precisão em áreas menos desenvolvidas;
- $12mm \times \sqrt{k}$  , para os levantamentos locais.

Sendo  $k$  = distância nivelada em quilômetros. Segue no APÊNDICE A o extrato do Quadro VI – da referida norma relativo a especificações para nivelamento geométrico.

Se este controle for mantido na execução então os valores estipulados para sua respectiva classe (Tabela 1) serão alcançados após o ajustamento.

Por exemplo, se a modelagem de  $\Sigma_{Lb}$  do exemplo de GEMAEL 2009 do APÊNDICE D for de acordo com o quadro de classificação para levantamentos altimétricos do IBGE (APÊNDICE A), com todos os requisitos necessários para ser classificado como de precisão para áreas mais desenvolvidas, de diferença máxima entre o nivelamento e o contranivelamento de uma linha melhor que  $6mm \times \sqrt{k}$  ,  $k$  em quilômetros. Dando uma variância de  $36 \times k \text{ mm}^2 = 0,036 \times k \text{ m}^2$ . A matriz peso ficaria:

$$P = \text{diag} \left( \frac{1}{0,036 \times 4}; \frac{1}{0,036 \times 2}; \frac{1}{0,036 \times 2}; \frac{1}{0,036 \times 4}; \frac{1}{0,036 \times 2}; \frac{1}{0,036 \times 4} \right)$$

$$= diag \begin{bmatrix} 6,9444 \\ 13,8888 \\ 13,8888 \\ 6,9444 \\ 13,8888 \\ 6,9444 \end{bmatrix}$$

Resultando em um aumento de aproximadamente 2700% no valor de  $\hat{\sigma}_0^2$  em comparação com o valor sem considerar o quadro do IBGE.

### 2.2.1 Análise global do ajustamento (teste Chi quadrado)

Para verificar se o fator de variância a posteriori  $\hat{\sigma}_0^2$  pode ser igualado ao fator de variância a priori ( $\sigma_0^2 = 1$ ) com certo grau de confiança para uma certa redundância (gl), é realizado o teste Chi quadrado ( $\chi^2$ ).

O teste  $\chi^2$  consiste em tentar dizer sim a “O valor da minha variância a posteriori  $\hat{\sigma}_0^2$  é aceitável para ainda ser encarada como igual ao fator de variância a priori  $\sigma_0^2$ ?”, ou seja, deve-se:

Aceitar a seguinte hipótese nula:

$$H_0 : \hat{\sigma}_0^2 = \sigma^2 \quad (8)$$

E rejeitar a seguinte hipótese alternativa:

$$H_1 : \hat{\sigma}_0^2 \neq \sigma^2 \quad (9)$$

O grau de significância  $\alpha$  (ou grau de confiança  $1 - \alpha$ ) delimita o intervalo de confiança com que se pode fazer a afirmação.

Geralmente adota-se um nível de significância  $\alpha = 5\% \rightarrow 1 - \alpha = 95\%$ . Assim, para responder a pergunta anterior é necessário responder: “Quais os valores mínimo e máximo de  $\hat{\sigma}_0^2$  para os quais se encontram 95% das amostras?”

A declaração acima em linguagem de probabilidade é:

$$P(\sigma^2_{\min} < \sigma^2 < \sigma^2_{\max}) = 95 \quad (10)$$

Desenvolvendo a equação acima pode-se chegar a equação que marca a aceitação da hipótese nula:

$$\chi^2(0,025; gl) > \frac{gl \cdot S^2}{\sigma^2} > \chi^2(9,725; gl) \quad (11)$$

A figura 1 abaixo ilustra o significado da aceitação,

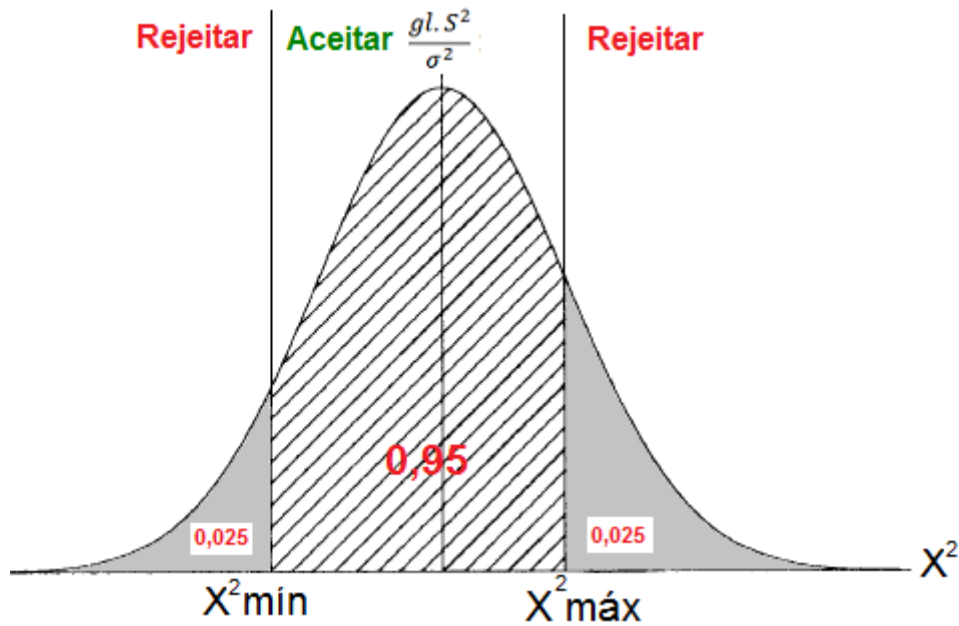


Figura 1 - Representação gráfica da aceitação no teste Chi quadrado

### 2.2.2 Identificação local de erros grosseiros (teste Baarda)

Após a detecção de erro, a identificação ou localização da possível fonte causadora é necessária. O método de Baarda se utiliza da correção normalizada que é assim definida para uma MVC das observações diagonal (Camargo, 2000):

$$w_i = \frac{v_i}{\sigma_{v_i}} \quad (12)$$

Com  $1 \leq i \leq n$ ,  $w$  assume a correção normalizada, e  $v_i$  o resíduo da  $i$ -ésima observação e  $\sigma_{v_i}$  o desvio padrão deste resíduo. Neste caso, a MVC dos resíduos é calculada com o fator de variância a priori. O teste será aceito caso:

$$-\sqrt{X^2_{(\alpha 0,1)}} < w_i < \sqrt{X^2_{(\alpha 0,1)}} \quad (13)$$

Sendo, o nível de significância  $\alpha = n \times \alpha 0$ .

Caso contrário, a observação  $i$  pode conter erro.

Pode ocorrer de mais de uma observação ser identificada com erro grosseiro. Nesse caso, a observação mais provável de conter o erro será aquela com maior valor de  $w_i$ . Sendo identificada a observação com erro grosseiro é possível realizar a adaptação do ajustamento, por exemplo,

excluindo essa observação e realizando o ajustamento novamente. Outras formas de adaptação podem ser vistas em Teunissen (1998).

O teste pode ser repetido, até que todas as observações sejam aceitas no teste até o limite de se manter a redundância do sistema, cabendo ao calculista a decisão final sobre as observações com indicação de erro grosseiro.

### **2.3 Python adaptado a ajustamento**

De acordo com a Python Software Foundation (PSF), Python é uma linguagem voltada a objeto, interpretada, de altíssimo nível, modular, multiplataforma e, o mais importante, de implementação livre. É permitida a distribuição, comercial ou não, tanto da linguagem quanto de aplicações desenvolvidas nela, bastando cumprir a exigência de manter o aviso de créditos da PSF.

Python possui as versões 2.x e 3.x com pequenas diferenças de sintaxe e grandes diferenças em suporte. A versão 3.x possui maior suporte em caracteres unicode porém, por ser novo, algumas bibliotecas antigas consolidadas ainda não se adaptaram, impondo aos desenvolvedores menos experientes a versão 2.x.

A linguagem é tipicamente usada em aplicações web ou com o uso de ferramentas como o freeze ou Py2Exe é possível distribuir aplicações Python stand-alone, sem que o usuário tenha que instalar o interpretador Python separadamente.

Por ser uma linguagem interpretada, o interpretador Python escolhido lê linha-a-linha e converte em código de baixo nível à medida que vai executando o programa, sendo portanto mais dinâmico. O modo de escrita fica escrever-testar-corriger-escrever-testar-distribuir ao invés de escrever-compilar-testar-corriger-compilar-testar-distribuir.

Os interpretadores Python são acompanhados por bibliotecas padrão e um ambiente simples de desenvolvimento e edição de texto. Existem uma variada gama de alternativas ao interpretador da PSF, cada qual com um foco específico (jogos, web, banco de dados, científico, etc.).

A biblioteca padrão básica inclui módulos para processamento de texto e expressões regulares, protocolos de rede, acesso aos serviços do sistema operacional, criptografia, interface gráfica, etc. A modularidade da linguagem permite sua expansão de capacidades. Existe uma grande variedade de módulos adicionais criados sob as mais variadas licenças que podem ser instaladas em qualquer interpretador. Alguns interpretadores importantes estão na tabela abaixo:

**Tabela 3 - Exemplos de Interpretadores Python**

<b>Interpretador</b>	<b>Descrição</b>	<b>Licença</b>
<b>Python</b>	Interpretador básico da PSF	Própria - PSF
<b>PyPy</b>	Rápida interpretação do código	MIT
<b>Anaconda</b>	Inclui pacotes de desenvolvimento científico, matemático, engenharia e análise de dados	Freemium
<b>Jython</b>	Traduz o código Python em código java, permitindo importar classes JAVA como módulos Python.	PSF
<b>CPython</b>	Uma alternativa Python que suporta pacotes em C.	PSF
<b>Skulpt</b>	Implementação in-browser de Python. Sem necessidade de suporte do servidor ou plugins.	MIT

Por ser de implementação livre, a busca de funcionalidades avançadas requer um trabalho de pesquisa intenso, haja vista que existem inúmeras bibliotecas que cumprem com os mesmos objetivos, além de não existir um repositório oficial muito menos um completo.

Por exemplo, como módulos de construção gráfica pode-se utilizar wxPython, PyGTK, PyQt (usado para plugins em QGIS), PythonCard, Tkinter, PyGlade, PMW, Kiwi, ncurses, OpenGL, PIL e VPython.

Para ajustamento, são importantes as bibliotecas da tabela a seguir:

**Tabela 4 - Módulos externos utilizados em ajustamento**

<b>Nome</b>	<b>Aplicação</b>	<b>Licença</b>
<b>Pandas</b>	Tratamento de matrizes	BSD
<b>Numpy</b>	Operação matricial	BSD
<b>Matplotlib.pyplot</b>	Construção de gráficos	BSD
<b>Scipy.stats</b>	Fórmulas estatísticas	Própria (similar a BSD)

As bibliotecas (pacotes ou módulos) podem ser importados e chamados. Tomando a biblioteca *math* (especializada em funções matemáticas) e sua função cosseno como exemplo, a importação se daria da seguinte forma:

```
>>> import math
>>> math.cos(x)
```

Também podem ser acessados através de um pseudônimo (alias):

```
>>> import math as mt
>>> mt.cos(x)
```

Ou pode-se somente importar a função, classe ou método que se deseja utilizar:

```
>>> from math import cos
>>> cos(x)
```

Neste caso, o referido passa a ser chamado diretamente, assim como as demais funções do *core* Python já o faziam.

A observação da licença é um traço importante pois não se deseja vínculos comerciais, a menos de um comentário de crédito. As mais comuns são:

- *General Public License* (GPL) autoriza gratuitamente a usar, desenvolver e distribuir o objeto da licença, porém a aplicação desenvolvida também deve ter licença GPL, ou seja, sem fins comerciais.
- *Lesser General Public License* (LGPL) autoriza gratuitamente a usar, desenvolver e distribuir o objeto da licença, e a aplicação desenvolvida não precisa ter licença GPL, ou seja, pode ter fins comerciais.
- *Berkeley Software Distribution* (BSD) autoriza praticamente a fazer qualquer coisa (copiar, modificar, mesclar, publicar, distribuir, sublicensiar, vender cópias, etc.) apenas isentando os criadores caso os usuários tenham algum tipo de problema.
- *Massachusetts Institute of Technology* (MIT) é similar a BSD, mas exclui uma cláusula que proíbe o uso do nome dos detentores do crédito em promoção da aplicação desenvolvida, ou seja, é mais permissiva.
- *Freemium*, o objeto da licença é gratuito com recursos adicionais pagos.

- Proprietária é uma licença que exige pagamento completo ou de parte do objeto (uso do nome da marca, uso da aplicação, distribuição mesclada com outra aplicação, etc. ).

Todas, exceto a proprietária, requerem os devidos créditos incluídos em porções do código criado, ou seja, escrever “Copyright (c) <ano> <desenvolvedor>” em partes do código.

Por fim, sobre os ambientes de desenvolvimento integrado (*integrated development environment* - IDE), são softwares que vão além de um editor de texto normal, apoiando o desenvolvedor com realce e cor de sintaxe, autocompletamento inteligente, renomeia uma variável em todos os seus usos, debugging (checar dados inválidos ou bugs), cronômetro da aplicação, guide user interface (GUI), etc. As IDE's que suportam Python mais famosas são descritas na tabela 4 abaixo:

**Tabela 5 - IDE's bastante utilizados em Python**

<b>Nome</b>	<b>Descrição</b>	<b>Licença</b>
<b>Eclipse + Pydev</b>	Multilinguagem. Além das características básicas, possui integração com o webframework Django, dicas de código e console interativo.	Eclipse Public License (EPL)
<b>Pycharm</b>	Provê análise de código, autocompletamento inteligente, dicas de código, fácil troca de interpretadores e instalação de módulos, um debugger gráfico. Suporte aos webframeworks Django, Web2Py e Flask.	Edição Professional: Proprietária (paga) Edição Educacional: Apache License
<b>Eric</b>	Além das características básicas, possui designer de GUI's integrado com o Qt Designer, debugger gráfico, sistema integrado de documentação dos códigos fonte e diagramas da aplicação. Suporte ao webframework Django.	GPL
<b>Spyder</b>	Editor de sintaxe com realce, coloração e autocompletamento. Suporte a múltiplos consoles (incluindo IPython) e capacidade de explorar as variáveis através de um GUI.	MIT

Mais informações como palavras de uso reservado, funções embutidas (built-in), operadores, variáveis e laços com exemplos de sintaxe se encontram no APÊNDICE B.



## 2.4 Desenvolvimento web

De acordo com o portal w3schools.com, um site na Internet tem um endereço virtual chamado de domínio e todos os seus arquivos e dados são armazenados (hospedados) em um computador remoto chamado de servidor (*Host*). Um internauta, ao clicar em um link, envia uma solicitação de dados de determinado endereço a determinado servidor, este por sua vez envia uma resposta com os dados solicitados.

O navegador (Browser) é o software no computador do internauta responsável por organizar os dados recebidos na tela. Ele pode ainda encaminhar os dados para alguma pasta local, no caso de dados não renderizáveis como arquivos executáveis (.exe), arquivos compactados (.zip), arquivos de vídeos fora de sua estrutura de marcação, etc. .

Os dados a serem renderizados pelo navegador possuem toda sua estrutura organizada por uma linguagem de **marcação** chamada de HTML que classifica o conteúdo em títulos, cabeçalhos, rodapés, links, imagens, vídeo, tabelas, parágrafos, entre outras.

O estilo e o layout do conteúdo estruturado em HTML são feitos por uma linguagem de apoio chamada CSS (*Cascading Style Sheets*) que controla fontes, cores, margens, alturas, larguras, imagens de fundo, posicionamento entre outras propriedades.

Já o comportamento do conteúdo no navegador (*client-side*) é feito por uma linguagem de **programação** chamada de Java Script que controla animações, interações com o usuário e a dinamização do conteúdo.

Linguagens de marcação e estilo não são apropriadas quando se requer o funcionamento de algum programa no servidor (*server-side*), para isso, são comuns linguagens mais robustas (ASP, PHP, Python, Java, Java Script, Lua, Ruby, Pearl, etc.) que são compiladas pelo computador remoto que retorna dados processados, que em seguida são embutidos em trechos do código HTML que por fim retornam ao usuário. Ao visualizar o código de uma página, não é possível ver essa outra linguagem pois ela foi convertida em HTML ainda no servidor, o navegador do internauta já recebe os dados prontos para serem exibidos. Abaixo um exemplo de um código Python incorporado (*embedded*) no

HTML, a ser compilado pelo servidor, e como ele é recebido no browser do internauta.

```
<html>
  <head>
    <style> h2{color: red;}</style>
  </head>
  <body>
    <h2>{{import math
      response.write(math.sqrt(4*9))}}</h2>
    <h3>{{for i in ['Maria ', 'Joaquina ', 'de ', 'Amaral ', 'Pereira ', 'Goes']:
      response.write(i)
      pass}}</h3>
  </body>
</html>
```

---

```
<html>
  <head>
    <style> h2{color: red;}</style>
  </head>
  <body>
    <h2>6.0</h2>
    <h3>Maria Joaquina de Amaral Pereira
    Goes</h3>
  </body>
  <object id="889a6a5a-fa7b-b297-799a-
  1aab3a8fa2ba" width="0" height="0" type=
  "application/gas-events-bb"></object> == $0
</html>
```

**6.0**  
**Maria Joaquina de Amaral Pereira Goes**

Figura 2 - Exemplo de código Python incorporado. Acima, o código original. Abaixo, o código recebido pelo internauta e como ele foi renderizado pelo seu browser

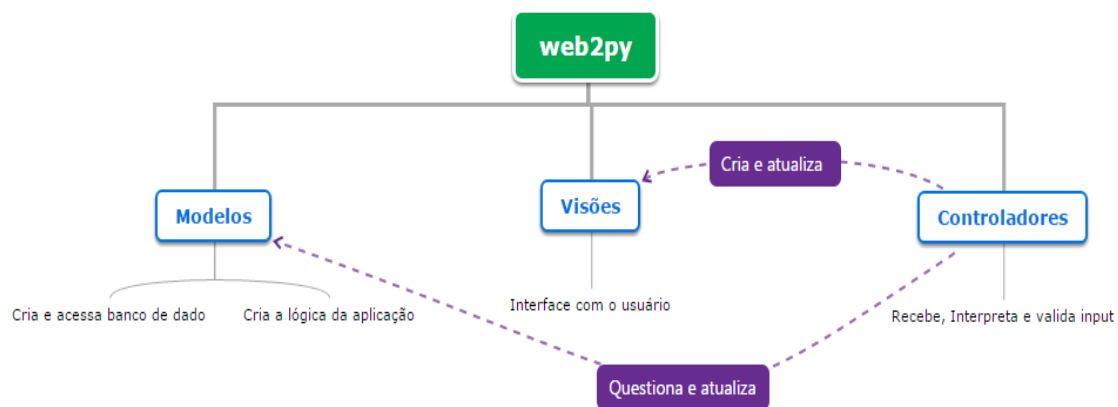
No caso, os elementos “<style>” e “</style>” marcam o início e o fim da linguagem CSS e os elementos “{{” e “}}” marcam o começo e o fim da linguagem Python.

Diferenciar se a linguagem foi compilada pelo servidor ou pelo navegador do internauta é importante para um desenvolvedor, uma vez que é medida de planejamento na escolha da linguagem. Deve-se balancear se é mais conveniente o navegador fazer o processamento em Java Script ou os dados serem enviados ao servidor que os processaria na linguagem mais apropriada para então retornar ao computador do internauta. Uma divisão equânime da carga computacional é o mais sensato.

Sendo assim, os cálculos robustos ou que requeiram acessar a base de dados remota geralmente são feitos pelo servidor e os cálculos leves são feitos no computador do internauta, compilados pelo seu navegador. Por exemplo, são compilados pelo servidor aplicações que controlam a transmissão de dados via streaming, aplicações com intensa entrada e saída de dados ou

aplicações que produzam um site complexo de uma página apenas e são compilados pelo navegador aplicações Java Script que recebem entrada do mouse ou teclado para alterar o conteúdo na tela.

Outro aspecto importante é a integração entre HTML e Python através da arquitetura *model-view-controller* que neste trabalho se concretiza com o uso do *webframework* Web2py. Esta escolha se deve a não haver necessidade de instalação desta ferramenta, ou seja, ela não depende do sistema operacional mas apenas do interpretador Python presente para começar a usá-la. Aqui cada projeto de site é chamado de aplicação e ainda permite-se diversas aplicações para um mesmo *webframework*. A figura 3 abaixo ilustra o funcionamento desta arquitetura.



**Figura 3 - Ilustração do funcionamento model-view-controller**

Bancos de dados e variáveis globais são criados na construção dos modelos, as visões são construídas com código misto HTML-Python interfaceando com os usuários e os controladores recebem os inputs e acessam bancos de dados ou modifica as partes das visões com trecho em Python.

### 3. APRESENTAÇÃO DA PLATAFORMA

#### 3.1 Entrada de dados

O software de ajustamento deve receber do usuário as informações do sistema a ser ajustado, são elas:

- Os parâmetros a serem ajustados;
- O vetor de tempos independentes;
- Os comprimentos das linhas;
- Organização do sistema.

Como não se sabe, ou não se deseja definir em um primeiro momento, o tamanho do sistema, optou-se pela entrada de dados em forma de um arquivo de texto sem formatação (.txt) de forma padronizada. O software deve, antes de tudo, analisar o arquivo enviado pelo usuário para fazer sua validação.

O padrão do arquivo texto pode variar de acordo com as opções que o usuário escolher no momento da execução do software, inicialmente consiste do sistema de equações reescrito em  $n$  linhas e 4 colunas, sendo  $n$  a quantidade de equações do sistema, entre uma coluna e a seguinte deve existir ao menos um espaço.

A 1ª coluna corresponde ao nível da estação de ré, a 2ª coluna ao nível da estação de vante, a 3ª coluna ao desnível observado entre as estações (vante-ré) e a 4ª coluna ao comprimento da linha da respectiva observação, as três primeiras em metros e a última em quilômetros que pode ter o mesmo valor em todas as linhas caso não se deseje a ponderação das observações. O indicativo de quais são os parâmetros a serem ajustados é a presença de algum caractere não convertível para número.

Para aceitar também decimais com vírgula, o sistema sempre converte os decimais da 3ª e 4ª coluna do arquivo texto para decimais com ponto e assim permanecem.

Por exemplo, o sistema abaixo cujo valor do desnível no ponto  $h_A=0$ :

$$\begin{cases} h_I - h_A = 6,16 \\ h_{II} - h_A = 12,57 \\ h_{II} - h_I = 6,41 \\ h_{III} - h_A = 1,09 \\ h_{II} - h_{III} = 11,58 \\ h_I - h_{III} = 5,07 \end{cases}$$

Deve ser convertido na seguinte entrada escrita em um arquivo .txt:

1	2	3	4	5
0	hI	6.16	4	
0	hII	12.57	2	
hI	hII	6.41	2	
0	hIII	1.09	4	
hIII	hII	11.58	2	
hIII	hI	5.07	4	

1ª col: estações de ré (-)  
2ª col: estações de vante (+)  
\*: espaços separam as colunas  
\*: sinais não são necessários  
\*: incógnitas possuem letras  
\*: Referências de nível possuem apenas números

3ª col: desníveis em [m]  
4ª col: distâncias em [Km]  
\*: separador decimal é (.)  
\*: não colocar espaços após o último caractere de cada linha  
\*: apenas números

Figura 4 - Exemplo de arquivo texto aceito na plataforma

Os parâmetros a serem ajustados serão reconhecidos pelo software se ele não conseguir convertê-los para números. Ou seja, a presença de uma letra ou pontuação que não seja vírgula caracteriza as incógnitas.

Não são convertíveis para numeral: Ha, P2, 2/3, 23!, etc.

São convertíveis para numeral: +23, -9, 23,4, etc.

A 4ª coluna, escrita com o mesmo valor, indica ao software que não se deseja a ponderação das observações.

Posteriormente verificou-se a possibilidade do usuário demandar usar uma modelagem própria dos pesos e não a oferecida em função dos comprimentos da linhas, a solução foi oferecer a opção de converter os dados da 4ª coluna diretamente em desvios padrões das observações em metros.

O sistema métrico adotado para exibição dos resultados foi o Sistema Internacional, ou seja, metros. Mas é claro que se o usuário passar as informações de entrada em pés, os resultados estarão em pés. Percebe-se que devido aos pesos serem relativos, um fator multiplicativo que altere todos os pesos não altera o resultado do ajustamento dos parâmetros e das observações, mas altera as análises pós-ajustamento, ou seja, as MVC's, teste  $\chi^2$  e teste Baarda.

Devido ao teste  $\chi^2$  em observações de desnível não ser aceito de imediato, foi oferecida a opção de deixar os desvios padrões das observações no modo automático, que basicamente testa um fator multiplicativo iterativamente para os pesos até a aceitação no referido teste.

### 3.2 Fluxograma do Processamento

O primeiro passo do processamento é manipular o arquivo texto enviado para o servidor, bem como as opções selecionadas, e armazená-las em variáveis do ambiente Python, qualquer erro nesta etapa provoca uma mensagem de erro em vez do relatório. Independente da validade do arquivo de texto, ele é deletado para não ocupar espaço na memória do HD do servidor.

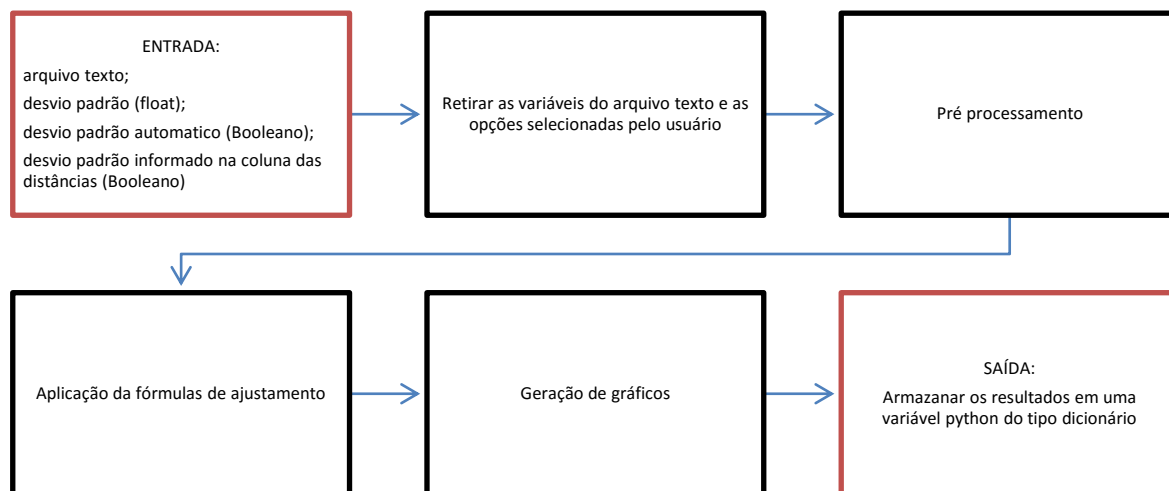
O Segundo passo é o pré-processamento das variáveis brutas, momento em que o software realiza pequenos ajustes como:

- Subtrai ou soma os níveis das estações de vante ou ré, respectivamente, dos valores de desnível, montando assim o vetor  $L_b$ ;
- Verifica a presença de injunções e sai do modo de desvio padrão automático;
- Encontra os parâmetros a serem ajustados e exclui suas duplicatas;
- Verifica a quantidade de parâmetros e equações e preenche a matriz de coeficientes  $A$  com 0,-1 ou 1 coerentemente;
- Monta a matriz de pesos  $P$  de acordo com as opções do usuário e com o arquivo de texto enviado;

O terceiro passo é a aplicação das fórmulas, este passo possui um laço iterativo que encontra o desvio padrão da observação que, de acordo com a modelagem dos erros das observações oferecida pelo software, implica na aceitação no teste  $\chi^2$ . Independente da aceitação neste teste, os cálculos prosseguem obtendo-se as MVC's.

O quarto e último passo executa funções extras de geração de gráficos para enriquecer o relatório.

Todos os cálculos feitos, uma variável Python do tipo dicionário contendo os resultados é passada adiante. Abaixo a figura 5 representando o fluxograma do processamento.



**Figura 5 - Fluxograma do processamento em Python**

Durante o ajustamento foram utilizadas bibliotecas Python externas de uso aberto como o numpy, scipy, matplotlib.pyplot e pandas, de modo que o interpretador Python instalado na máquina do servidor deve possuí-las, caso contrário um erro pode ser provocado. Devido a algumas destas bibliotecas bem como o webframework não terem migrado para Python 3.x toda a implementação do ajustamento foi feita em Python 2.x. Informações de como utilizar esses módulos constam no APÊNDICE F.

As 4 etapas do processamento são feitas por dois arquivos, um contendo o fluxo das etapas chamado de NivParamet.py, acrônimo de nivelamento paramétrico, e outro contendo funções com as fórmulas de ajustamento chamado de Mtools.py, acrônimo de minhas *tools* ou minhas ferramentas. Utilizando o web2py, bastam estes módulos estar na pasta respectiva para o sucesso na importação.

### 3.3 Interface com o usuário

Por entender que o software deve ser encarado como a primeira ferramenta do site, foi confeccionada uma página inicial simples com um link para a página de ajustamento altimétrico. O web2py se encarrega de fazer o reaproveitamento de layouts entre as páginas, no caso, um menu superior e um rodapé é usado em ambas, mas na página do ajustamento existem subpáginas que reaproveitam um menu lateral bem como o menu superior e o rodapé. Para se ter uma melhor visualização do que está por trás da ferramenta de ajustamento segue na figura 6 abaixo a concepção da interface de execução do processamento.



Figura 6 - Página de execução do ajustamento altimétrico

O menu superior e o rodapé são simples devido a facilidade de serem replicados fora do ambiente web2py, assim em projetos futuros incorporados ao site não será obrigatório utilizar o mesmo *webframework* ao custo de apenas reconstruir estas partes. O menu superior é uma variável Python construída, na montagem do modelo a partir de funções do web2py e inseridos na visão através de trechos de código Python *embedded*, (poderia ter sido feito em HTML pois são apenas links para outras páginas). Já o rodapé é apenas um botão no canto inferior direito, escrito em JavaScript na montagem das



visões, ao clicar no botão 'Share' o usuário é direcionado a compartilhar a página atual nas mídias sociais (Twitter, Facebook e Google +).

O menu lateral à esquerda está presente nas subpáginas da ferramenta de ajustamento, ele é feito em HTML mas alterado por código em JavaScript que permite escondê-lo quando a janela diminui para aquém de 768 pixels de largura, ou seja, ele não aparece, por exemplo, em navegação por celulares. Para a alternância entre esconder e aparecer, independente do tamanho da janela, foi confeccionado um botão com o símbolo '<>>' que acompanha a barra lateral no seu canto superior direito. Este botão se torna útil caso o usuário salve o relatório em PDF utilizando a ferramenta de impressão dos navegadores ao eliminar esta barra da janela. Outra funcionalidade é um indicador de processamentos com sucesso de cor verde ao lado do botão 'OTIMIZAR REDE', a cada processamento com sucesso a variável armazenada no banco de dados interno do site é incrementada.

Ao clicar em uma subpágina da ferramenta de ajustamento, uma requisição é enviada ao servidor que, por sua vez, passa ao controlador web2py, existe uma função em Python no arquivo controlador para cada visão requisitada que permite alterar o arquivo de sua respectiva visão. Para a maioria das subpáginas o controlador só retorna a variável global menu superior, mas na página de execução do ajustamento (Fig 6), o controlador retorna um formulário, aguarda inputs do usuário, executa o processamento em caso de validação e altera a sua respectiva visão de modo a exibir o relatório do ajustamento.

O formulário recebe 4 entradas, o arquivo texto, uma opção de modelagem dos desvios padrões das observações com alguns valores na função de auto-completamento (por padrão  $dp \text{ em } mm = 1 \times \sqrt{\text{comprimento da linha em Km}}$ ), uma entrada de um booleano que indica a intenção de deixar o desvio padrão no modo automático e uma entrada booleana que indica que o arquivo de texto possui na 4ª coluna os desvios padrões em metros de uma modelagem própria em vez dos comprimentos das linhas em quilômetros.

O espaço de comentários logo abaixo é um trecho de código em JavaScript que permite o login pelo facebook para a postagem. A escolha do comentário ser desta forma foi devido a facilidade de moderação, não necessidade de

armazenar usuários, não necessidade de armazenar os comentários e ser de fácil implementação.

Os estilos são derivados de arquivos css disponibilizados gratuitamente na internet. A empresa Twitter disponibiliza uma folha de estilos chamada de bootstrap.css e seu respectivo JavaScript bootstrap.js que neste trabalho foram aproveitados nas tabelas do relatório. O Web2py possui uma sua folha de estilo própria que foi aproveitada na estilização do menu superior e do formulário na página de execução do ajustamento. O estilo do menu lateral proveio de um template de site pronto de [www.free-css.com/free-css-templates](http://www.free-css.com/free-css-templates) modificado para combinar com o menu superior. Existem códigos css declarados na seção <head> do arquivo de visão e também declarados dentro da seção a qual se destinam formatar.

### **3.4 Relatório on-line**

Entre o formulário e o espaço de comentários é exibido o relatório. Seu código se encontra dentro um laço condicional que só é acionado caso o processamento seja realizado com sucesso. Em casos de processamentos com erros anteriormente previstos, no lugar do relatório surge uma mensagem de erro amigável.

O relatório é composto por quatro tabelas e um gráfico constantes nas figuras 7, 8, 9 e 10 abaixo. A 1ª tabela é uma concatenação do arquivo de entrada com o desvio padrão da observação, observação ajustada  $L_a$  e resíduo  $V$ .

## Ajustamento Paramétrico: 6 equações e 3 incógnitas

Observações de entrada + Processamento : ▽

Estação(ré)	Visada(vante):	Desnível(m):	Distância(Km):	Desvio Padrão da observação(m) = $21.0 \cdot \text{RAI} / Z(\text{dist}[\text{Km}]) / 1000$ :	Desnível ajustado(m):	Resíduos(m):
0	hl	6.16	4	0.04	6.16	-3.5527136788e-15
0	hll	12.57	2	0.0282842712475	12.59	0.02
hl	hll	6.41	2	0.0282842712475	6.43	0.02
0	hlll	1.09	4	0.04	1.05	-0.04
hlll	hll	11.58	2	0.0282842712475	11.54	-0.04
hlll	hl	5.07	4	0.04	5.11	0.04

**Figura 7 -Tabela das observações ajustadas**

No caso, o desvio padrão modelado seguiu a fórmula  $dp \text{ em mm} = k \times \sqrt{\text{comprimento da linha em Km}}$ , onde  $k=21,0$  atingiu aceitação no teste  $\chi^2$ .

A 2ª tabela informa os parâmetros ajustados ao lado dos seus respectivos desvios padrões, para isso efetua-se a raiz quadrada dos elementos da diagonal principal de  $\Sigma_{\chi^2}$ . Independente da aceitação no teste  $\chi^2$  e consequentemente da equiparação entre as variâncias a priori e posteriori ( $\hat{\sigma}_0^2 = \sigma^2$ ), a fórmula utilizada é com a variância a posteriori, de modo que a não aceitação da hipótese nula pode acarretar em perda de significado desta informação.

Parâmetros Ajustados: ▽

Ponto	Alt. Ajustada(m):	Desv Padrão(m):
hl	6.16	0.0326598632371
hlll	1.05	0.0326598632371
hll	12.59	0.0282842712475

**Figura 8 - Tabela dos parâmetros ajustados**

A 3ª tabela possui dados sobre o teste  $\chi^2$  e é acompanhada de um gráfico ilustrativo, a linha vertical preta representa o  $\chi^2$  calculado, se ela estiver em cima das regiões em vermelho ou simplesmente não aparecendo, significa hipótese nula rejeitada.

Análise da qualidade do ajustamento - Teste estatístico para detecção de erros ( $X^2$ ):  $\tau$

Variância a priori( $m^2$ ) ( $\sigma^2_{pri}$ ):	Variância a posteriori( $m^2$ ) ( $\sigma^2_{pos}$ ):	MIN = $X^2(3, 2.5\%)$ :	Calculado = $\sigma^2_{pos} * (\text{Graus de Liberdade}) / \sigma^2_{pri}$ :	MAX = $X^2(3, 97.5\%)$ :
1	1.66666666667	0.215795282624	5.0	9.3484036045

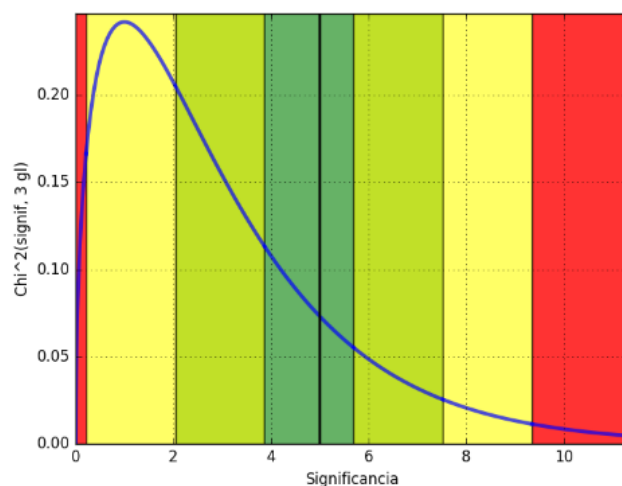


Figura 9 - Resultado do teste chi quadrado

O gráfico é feito em Python com a biblioteca matplotlib.pyplot e salvo sempre com o mesmo nome em formato de imagem .png, o que sobrescreve o arquivo anterior e evita o consumo de memória do HD do servidor

Análise da qualidade do ajustamento - Teste estatístico para Identificação das Observações com Erro Grosseiro (Baarda):  $\tau$

Obs	Estação(ré)	Visada(vante):	Desnível(m):	Distância(Km):	Resíduos(m):	Desv Padrão dos Resíduos(m):	Correção normalizada:	Aceitação por Baarda - $\pm 2.08615015398$ :
1	0	hl	6.16	4	-3.5527136788e-15	0.04	-8.881784197e-14	Não existe evidências estatísticas para a rejeição
2	0	hll	12.57	2	0.02	0.0230940107676	0.866025403784	Não existe evidências estatísticas para a rejeição
3	hl	hll	6.41	2	0.02	0.0230940107676	0.866025403784	Não existe evidências estatísticas para a rejeição
4	0	hlII	1.09	4	-0.04	0.04	-1.0	Não existe evidências estatísticas para a rejeição
5	hlII	hll	11.58	2	-0.04	0.0230940107676	-1.73205080757	Não existe evidências estatísticas para a rejeição
6	hlII	hl	5.07	4	0.04	0.04	1.0	Não existe evidências estatísticas para a rejeição

Figura 10 - Resultado do teste Baarda

A 4ª e última tabela representa o teste para a identificação de observações com erros grosseiros (Baarda). Ela se dá pela concatenação do arquivo de entrada com os resíduos  $V$ , com o desvio padrão dos resíduos obtidos a partir raiz quadrada dos elementos da diagonal principal de  $\Sigma_V$ , com a correção

normalizada e com uma coluna indicadora da aceitação do método. Esta coluna exibe 'Não existe evidências estatísticas para a rejeição' caso a correção normalizada se enquadre na aceitação e 'Existe evidências estatísticas para a rejeição' caso contrário.

Ao final, um botão de imprimir destaca o relatório do restante da página para o usuário poder salvar em formato PDF.

### 3.5 Mensagens de erro

A maior parte dos erros do uso da ferramenta são devido a não estruturação correta do arquivo .txt enviado. Outra fonte de erro é o excesso de erros sistemáticos e grosseiros no processo de medição que, acumulados, podem tornar o sistema de equações a serem ajustadas em um problema mal condicionado. Isto implica em um determinante da matriz dos coeficientes (A) próximo a zero, que por sua vez implica em dificuldades de se encontrar sua matriz inversa necessária nas fórmulas de ajustamento. Foram criadas medidas preventivas de erro no próprio formulário, estes bloqueios são:

- Não é possível 'Enviar' o formulário sem nenhum arquivo selecionado;
- Não é possível enviar arquivos com mais de 1048576 caracteres o que dá aproximadamente 1Mb;
- Não é possível escolher desvios padrões fora do intervalo 0,0001 e 999.

Na figura abaixo se tem as mensagens geradas:

<p>Arquivo</p> <p><input type="button" value="Escolher arquivo"/> Nenhum arquivo selecionado</p> <p>Enter from 0 to 1.04858e+06 characters</p> <p>Desvio Padrão = <math>X_{mm} * \text{RAIZ}(\text{dist}[\text{Km}])</math></p> <p><input type="text" value="1.00"/></p> <p><input checked="" type="checkbox"/> Desvio Padrão automático</p>	<p>Arquivo</p> <p><input type="button" value="Escolher arquivo"/> Nenhum arquivo selecionado</p> <p>Sem arquivo !</p> <p>Desvio Padrão = <math>X_{mm} * \text{RAIZ}(\text{dist}[\text{Km}])</math></p> <p><input type="text" value="999"/></p> <p>0.0001 &lt; X &lt; 100 !</p> <p><input checked="" type="checkbox"/> Desvio Padrão automático</p>
--	--

**Figura 11 - Mensagens de erro dentro do formulário. À esquerda erro devido ao tamanho excessivo do arquivo enviado. À direita erros devido a desvio padrão fora de intervalo e sem arquivo**

Se o envio foi autorizado, começa o processamento conforme descrito anteriormente. Utilizando-se do laço try/except de Python, foi implementado

uma proteção para evitar mensagens de erro fora do site, o objetivo não é apenas ser amigável mas também proteger o sistema. Qualquer tipo de erro dentro do laço try encaminha o prosseguimento do processamento para um laço alternativo except mais simples e que apenas muda as variáveis passadas para o arquivo html da sua respectiva visão, o que permite aparecer a mensagem de erro da figura 12 abaixo.



Figura 12 - Mensagem de erro amigável

O link 'clique aqui' dentro da mensagem de erro encaminha o navegante à página 'Exemplos e Demos' onde ele pode verificar os modelos de questões de ajustamento altimétrico adaptadas ao uso da ferramenta.

### 3.6 Ajuda

Ao lado do título do formulário foi criado o botão de ajuda da ferramenta, consistindo de perguntas e respostas que guiam o usuário no entendimento da plataforma, ela segue na figura 13, a seguir.

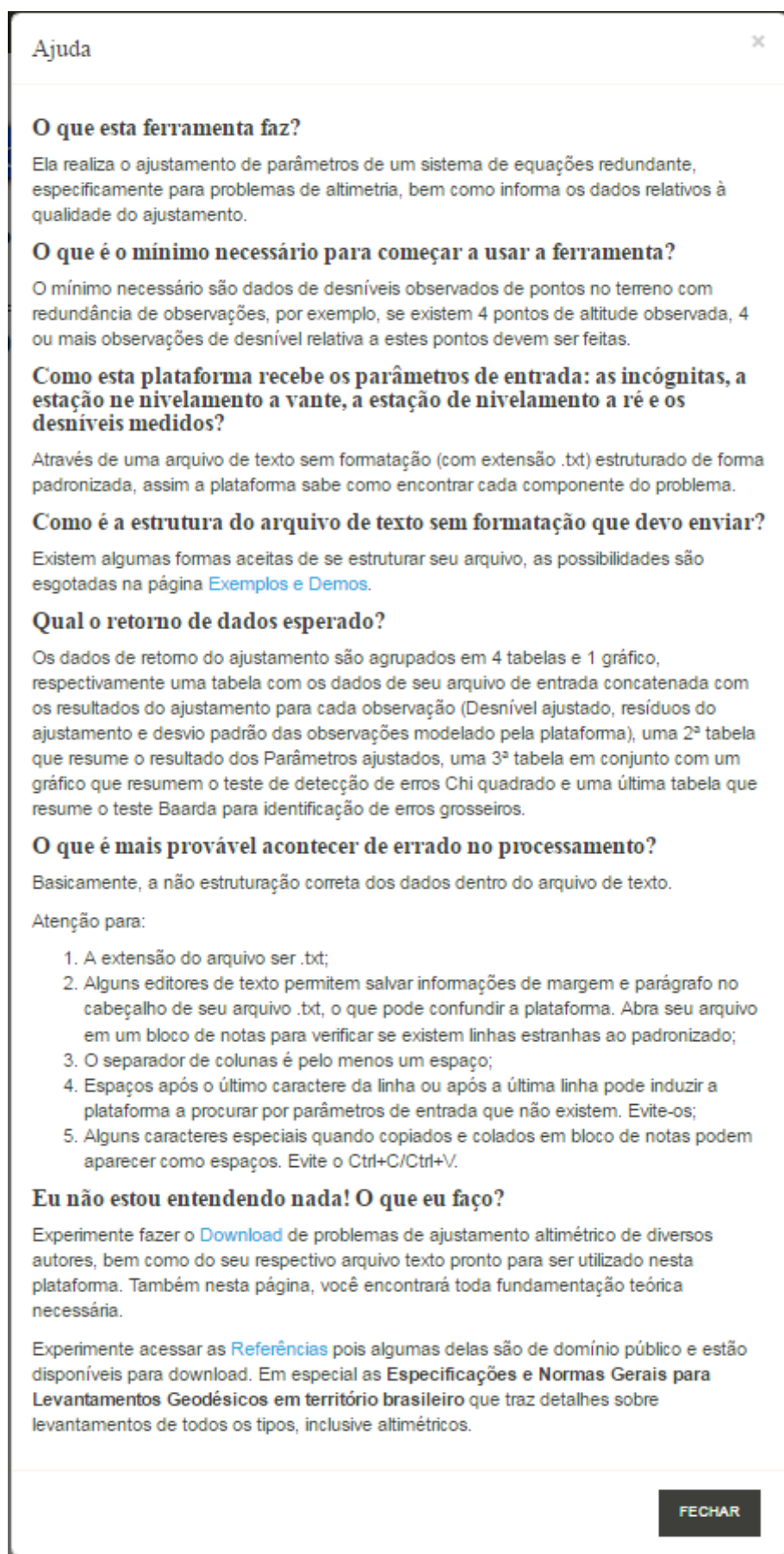


Figura 13 - Ajuda da ferramenta



### 3.7 Hospedagem

A hospedagem escolhida, neste primeiro momento, foi a oferecida pela empresa [PythonAnywhere.com](http://PythonAnywhere.com), os motivos que a ensejaram foram a gratuidade, o suporte Python abrangente o que inclui as bibliotecas utilizadas por este trabalho e o fornecimento de um domínio também gratuito.

As desvantagens da gratuidade são espaço no HD do servidor reduzido a 50 Mb, não permitir mais de uma aplicação e não permitir eventos agendados aplicados ao site. Isto é um fator limitante para o crescimento do projeto de site como um todo, mas é aceitável para este projeto apenas.

Pode-se considerar a disponibilização de um computador do Instituto ligado à internet em tempo integral e inteiramente dedicado à função de servidor. O web2py permite utilizar, como software de servidor, a opção embutida (*Rocket*) ou um software servidor de terceiros como o Apache.

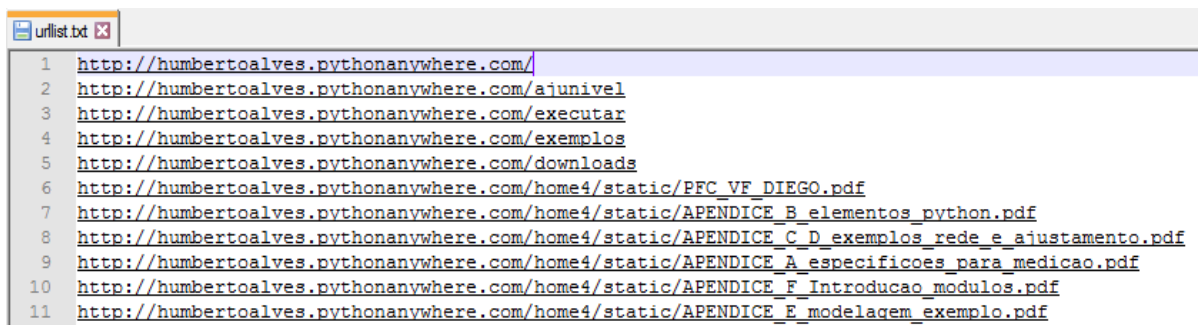
### 3.8 O Produto nas ferramentas de busca

Para o produto aparecer na ferramenta de busca do Google ele foi cadastrado no *Google Search Console*, a confirmação de propriedade se deu inserindo uma tag HTML com um código gerado dentro da seção “<head></head>” da página inicial do site. A tag tem esta característica:

```
<meta name="google-site-verification" content="código_gerado" />
```

Quando o site é novo o Google recomenda construir um arquivo *sitemap* para permitir a indexação das demais partes do site, nada mais é que um arquivo de texto sem formatação com todas as páginas e conteúdo que o site possui, segue um exemplo na figura 13 abaixo.





```
1 http://humbertoalves.pythonanywhere.com/
2 http://humbertoalves.pythonanywhere.com/ajunivel
3 http://humbertoalves.pythonanywhere.com/executar
4 http://humbertoalves.pythonanywhere.com/exemplos
5 http://humbertoalves.pythonanywhere.com/downloads
6 http://humbertoalves.pythonanywhere.com/home4/static/PFC_VF_DIEGO.pdf
7 http://humbertoalves.pythonanywhere.com/home4/static/APENDICE_B_elementos_python.pdf
8 http://humbertoalves.pythonanywhere.com/home4/static/APENDICE_C_D_exemplos_rede_e_ajustamento.pdf
9 http://humbertoalves.pythonanywhere.com/home4/static/APENDICE_A_especificacoes_para_medicao.pdf
10 http://humbertoalves.pythonanywhere.com/home4/static/APENDICE_F_Introducao_modulos.pdf
11 http://humbertoalves.pythonanywhere.com/home4/static/APENDICE_E_modelagem_exemplo.pdf
```

Figura 14 - Sitemap em formato .txt

Para finalizar, este arquivo *sitemap* deve ser submetido ao Google conforme a figura 15 abaixo, a indexação da página se dá em algumas horas.

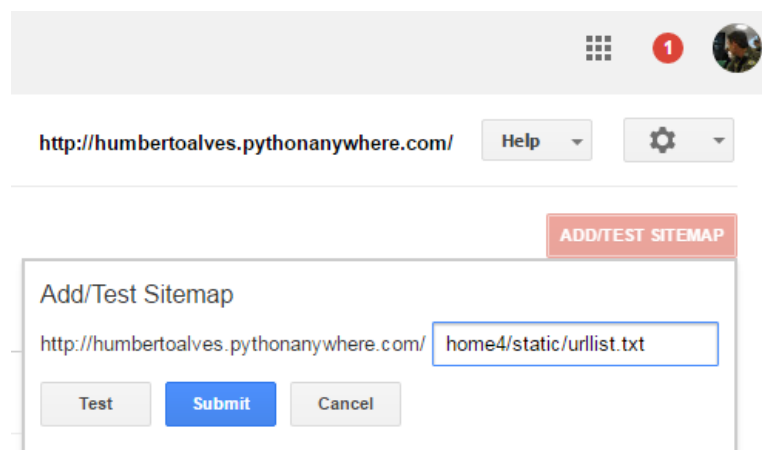


Figura 15 - Informando o Google Search engine sobre o seu sitemap

E por fim, a descrição do site é acrescentada na seção “<head></head>” com uma tag HTML neste formato:

```
<meta name="description" content="Descrição do site"/>
```

## 4. ESPECIFICAÇÕES TÉCNICAS

### 4.1 Descrição do Produto

O produto deste projeto corresponde a um site na internet cujo objetivo é realizar um ajustamento de uma rede de nivelamento e exibir o relatório on-line. Ele traz meios de interagir com os usuários possuindo um viés educativo ao permitir a exploração de partes do conteúdo deste PFC e dos apêndices como forma de ajuda ao uso da ferramenta e de aprendizado em Python, ajustamento e construção de sites.

O produto gerado pode servir de maneira educacional ou profissional para nivelamento de redes independente delas estarem em consonância com as normas do IBGE.

Detalhes sobre o produto:

- a) Título do site: se6alunos
- b) Título da ferramenta: Aju Nív Geo
- c) Domínio da home page: [se6alunos.pythonanywhere.com](http://se6alunos.pythonanywhere.com)
- d) Data da elaboração:
  - i. 1ª versão: 25/05/2016
  - ii. 2ª versão: 01/07/2016
  - iii. 3ª versão: 16/09/2016
- e) Responsável pela elaboração:
  - i. Nome: Humberto Diego Aguiar Alves
  - ii. Organização: Instituto Militar de Engenharia
- f) Tamanho do site:
  - i. Aplicação: 7,14MB dos quais
    - 1) HTML: 140KB
    - 2) Python: 31,02KB de controladores + 12,4KB de modelos + 35KB de módulos próprios
    - 3) CSS: 266KB
    - 4) Javascript: 421KB
    - 5) Imagens: 2,02MB
    - 6) Fontes: 210KB
    - 7) Arquivos para download: 3,95MB

- ii. Aplicação + webframework: 32,7MB
- g) Requisitos de software no servidor:
  - i. Interpretador: Python 2.7
  - ii. Bibliotecas: Python-numpy, Python-scipy, Python-tk, Python-matplotlib.pyplot, Python-pandas
  - iii. Webframework: Web2py
- h) Mapa do site:
  - i. Home page
  - ii. Aju Niv Geo
    - 1) Introdução
    - 2) Otimizar rede
    - 3) Exemplos e Demos
    - 4) Downloads
    - 5) Referências

## 4.2 Qualidade dos Dados do Produto

A precisão dos cálculos com aritméticas de pontos flutuantes segue o erro de representação da linguagem Python que trata variáveis flutuantes como um *float* de precisão dupla da norma internacional IEEE-754, ou seja, um número desse tipo tem precisão de 53 bits na sua representação binária (*Python v2.7.2 documentation*).

O tempo de execução foi medido inserindo um contador de tempo no início e no fim dos trechos de códigos, os problemas A, B, C e D são explicados no APÊNDICE G – Problemas de ajustamento altimétrico.

Tabela 6 - Tempo de execução do ajustamento

Problema	Tempos de execução em processador icore7 2,1Ghz Memória RAM 2GB		
<b>A</b>	1,529s	1,519s	1,474s
<b>B</b>	1,542s	1,531s	1,495s
<b>C</b>	1,491s	1,490s	1,505s
<b>D</b>	1,587s	1,56s	1,541s

## **5. CONCLUSÃO**

Foi confeccionada uma plataforma on-line para a realização de ajustamento com implementação da funcionalidade na linguagem Python e implementação da interface na linguagens JavaScript, HTML e CSS. Sendo útil no apoio ao ensino da matéria de ajustamento deste Instituto.

Os conhecimentos de programação e internet, fora da grade curricular da engenharia cartográfica, se destacaram pela gama de material disponível para consulta, sob as mais variadas formas, desde simples documentação em texto passando por canais de vídeos no Youtube até aplicativos de celular. Os mais variados métodos de didática são encontrados: exemplos, práticas e até mesmo jogos de programação, de forma que as dificuldades iniciais podem ser facilmente superadas por qualquer aluno ou professor.

Percebe-se que o desenvolvimento web está ao acesso de todos sendo não só uma ferramenta comercial, mas também de apoio didático a qualquer tipo de conhecimento, ou seja, esta ferramenta também será útil para a implementação de projetos de final de curso futuros e até de outros trabalhos acadêmicos.

## REFERÊNCIAS BIBLIOGRÁFICAS

GEMAEL, Camil. **Introdução ao ajustamento de observações: Aplicações geodésicas**. Curitiba. UFPR, 1994. Reimpressão 2004.

GHILANI, Charles D.; WOLF, Paul R. **Adjustment Computations: Spatial Data Analysis**. 4.ed. JOHN WILEY & SONS, 2007. 611p.

DALMOLIN, Quintino. **Ajustamento por mínimos quadrados**. Curitiba. UFPR, 2002. 169p

IBGE. **Especificações e Normas Gerais para Levantamentos Geodésicos em território brasileiro**. RESOLUÇÃO - PR no 22, de 21-07-83. Disponível em: <<ftp://geofp.ibge.gov.br/documentos/geodesia/pdf/bservico1602.pdf>>. Acesso em: 03/04/2016.

**Numpy reference 1.11.0: Written by the NumPy community**. 2015. Disponível em <<https://docs.scipy.org/doc/numpy-dev/numpy-ref.pdf>>. Acesso em 10/05/2016.

LAY, David C. **Linear Algebra and Its Applications**. 5th Edition, University of Maryland. Pearson. 2016. 579p

KIUSALAAS, Jaan . **Numerical Methods in Engineering With Python 3**. 3ª Ed. Pennsylvania State University.- Cambridge University Press. 2013. 432p.

DI PIERRO, Massimo. **Web2py complete reference manual**. 5th Ed. 2013. Library of Congress Cataloging-in-Publication. Disponível em <[https://dl.dropboxusercontent.com/u/18065445/web2py/web2py\\_manual\\_5th.pdf](https://dl.dropboxusercontent.com/u/18065445/web2py/web2py_manual_5th.pdf)> Acesso em 10/05/2016.

REINGART, Mariano; et al. **Web2py Application Development Cookbook**. Packt Publishing. 2012.

**Portal w3schools - The world's largest web developer site**. Disponível em <<http://www.w3schools.com/default.asp>> Acesso em 10/05/2016.

**Portal Python software Foundation**. Disponível em <<https://www.python.org/>> Acesso em 10/05/2016.

Camargo, Paulo de Oliveira. **Notas de aulas do Curso de Graduação em Engenharia Cartográfica**, FCT/UNESP. 2000. 222p.