6
Referências bibliográficas

[1] IHS Technology – Information Analytics Expertise "Global Wireless Subscriptions Reach 5 Billion", 17 de Setembro de 2010. Disponível em: < http://www.isuppli.com/Mobile-and-WirelessCommunications/News/Pages/Global-Wireless-Subscriptions-Reach-5-Billion.aspx >. Acesso em: 5 mar. 2013.

[2] Blogger- "La población mundial al 2012", 6 de janeiro de 2012. Disponível em:<http://soca36.blogspot.com.br/2012/01/la-poblacion-mundial-al-2012.html> Acesso em: Acesso em: 5 mar. 2013.

[3] Commentary Jana Research Technology "Mobile Phone subscribers Worldwide", 25 de Setembro de 2012. Disponível em:<http://www.jana.com/blog/whats-in-a-number-how-jana-expanded-its-reach-to-3-48-billion-mobile-phone-users/> Acesso em: Acesso em: 6 mar. 2013.

[4] Relatório Anual da Anatel 2012 "Telefonia Móvel, Evolução da Planta". Disponível: <http://www.anatel.gov.br/Portal/verificaDocumentos/documento.asp?numeroPublicacao=297390&pub=original&filtro=1&documentoPath=297390.pdf> Acesso em: Acesso em: 12 mar. 2013.

[5] Wikipédia a enciclopédia livre "Redes que proporcionam acesso sem fio banda larga". Disponível em:<http://pt.wikipedia.org/wiki/Banda_larga> Acesso em: Acesso em: 12 mar. 2013.

[6] Digital Agenda for Europe The importance of Radio Spectrum for wireless technologies". Disponível em:<http://ec.europa.eu/digital-agenda/en/what-radio-spectrum-policy> Acesso em: Acesso em: 13 mar. 2013.

[7] ScienceDirect A. F. Bruce, "Cognitive Radio Technology" Capítulo 1 - History and Background of Cognitive Radio Technology in Cognitive Radio Technology. Disponível em:<http://www.sciencedirect.com/science/book/9780750679527> Acesso em: Acesso em: 15 mar. 2013.

[8] Louis E. Frenzel Jr. "Fundamentos de Comunicação Eletrônica" - Volume 1 - Página 19, Mais espaço na faixa superior. Disponível em: <http://books.google.com.br/books?id=kUKaKXcrYhkC&pg=PA19&dq=o+espectro+foi+quase+totalmente+ocupado&hl=ptBR&sa=X&ei=OjENUpScD47K9gT0xYC4Aw&ved=0CC8Q6AEwAA#v=onepage&q=o%20espectro%20foi%20quase%20totalmente%20ocupado&f=false> Acesso em: Acesso em: 15 mar. 2013.

[9] Wikipédia a enciclopédia livre "Cognitive Radio History". Disponível em:< http://en.wikipedia.org/wiki/Cognitive_radio > Acesso em: 16 mar. 2013.

[10] IEEEXplore, Simon Haykin "Cognitive Radio: Brain – Empowered Wireless Communications. IEEE jornal on selected áreas in communications, vol.23, No." Fevereiro 2005. Disponível em:< http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1391031&tag=1> Acesso em: 20 mar. 2013.

[11] K. –C. Chen, Y. –J. Institute of Communication Engineering. National Taiwan University Taipei, Taiwan "Cognitive radio: Cognitive Radio Network Architecture: Part I – General Structure". Disponível em:< http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1391031&tag=1> Acesso em: 20 mar. 2013.

[12] Frank H. P. Fitzek and Marcos D. katz "Cognitive Wireless Networks: Concepts, Methodologies and Visions Inspiring the Age of Enlightenment of Wireless Communications". Disponível em:<http://books.google.com.br/books?id=PnlUctqJnbcC&pg=PA373&dq=cognitive+radio+spectrum+sensing&hl=ptBR&sa=X&ei=Rrk4UvHuLYTo2gWu4oCIBw&ved=0CD8Q6AEwAg#v=onepage&q=cognitive%20radio%20spectrum%20sensing&f=false> Acesso em: 22 mar. 2013.

[13] I. Hu, Zhen. II. Li, Husheng "Cognitive Radio Communications and Networking"Disponível:<http://books.google.com.br/books?id=9rb-=frontcover&dq=cognitive+radio+spectrum+sensing&hl=pt-BR&sa=X&ei=hgI3UrvEZT88QSsk4GYDA&ved=0CC8Q6AEwAA#v=onepage&q&f=false> Acesso em: 22 mar. 2013.

[14] Rádios Cognitivos "Compartilhamento do Espectro" Disponível em:<http://www.gta.ufrj.br/ensino/eel879/trabalhos_vf_2008_2/coutinho/CompartilhamentodoEspectro.html> Acesso em: 22 mar. 2013.

[15] X. Gao, G. Wu and T. Miki. "End-to-end QoS provisioning in mobile heterogeneous networks". IEEE Wireless Communications, vol.11, no.3, pp. 24-34, June 2004.

[16] Akyildiz, I. F., Wang, X., and Wang, W. 2005. "Wireless mesh networks: a survey". 4 de março 2005), pp 445-487. Disponível em:<http://www.sciencedirect.com/science/article/pii/S1389128604003457 > Acesso em: 23 mar. 2013.

[17] Yu-Yu Lin and Kwang-Cheng Chen, Fellow, IEEE "Distributed Spectrum Sharing in Cognitive Radio Networks - Game Theoretical View 2010". Disponíve em:<http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=5421750&queryText%3DDistributed+Spectrum+Sharing+in+Cognitive+Radio+Networks> Acesso em: 23 mar. 2013.

[18] YUCEK, T., ARSLAN, H., Communications Surveys Tutorials, "A survey of spectrum sensing algorithms for cognitive radio applications", IEEE, v. 11, n. 1, pp. 116 –130, mar. 2009. ISSN: 1553-877X. 1 de setembro de 2009. Disponível em:<http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=5421750&queryText%3DDistributed+Spectrum+Sharing+in+Cognitive+Radio+Networks> Acesso em: 24 mar. 2013.

[19] P.Kolodzy "Next generation communications: Kickoff meeting". In: *Proceedings of the Defense Advanced Research Projects Agency (DARPA)*, 2001. Acesso em: 24 mar. 2013.

[20] Pedro Smith Coutinho, Porf. José Ferreira de Rezende. "Detecção de Energia para Radio Cognitivos Usando GNU Radio e USRP2" março de 2011. Disponível em:<http://www.gta.ufrj.br/ftp/gta/TechReports/Coutinho11/Coutinho11.pdf> Acesso em: 24 março 2013.

[21] Andson Marreiros Malieriro. "Hanoff de Espectro em redes Baseadas em Radio Cognitivo Utilizando Redes Neurais Artificiais" junho de 2011. Disponível em:<http://repositorio.ufpa.br/jspui/bitstream/2011/2691/1/Dissertacao_HandoffEspectroRedes.pdf> Acesso em: 24 mar. 2013.

[22] Y. Zeng and Y.-C. Liang, IEEE Trans. on Veh. Technology "Spectrum-sensing algorithms for cognitive radio based on statistical covariances,"vol. 58, no. 4, pp. 1804–1815, maio de 2009. Disponível em:< http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4610972&tag=1> Acesso em: 28 março 2013.

[23] YUCEK, T., ARSLAN, H. *IEEE* "MMSE Noise Plus Interference Power Estimation in Adaptive OFDM Systems", Transactions on, v. 56, n. 6, pp. 3857 –3863, nov. 2007. ISSN: 0018-9545. 6 de novembro de 2007. Disponível em:< http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=04356997> Acesso em: 28 de março de 2013.

[24] SHANKAR, N., CORDEIRO, C., CHALLAPALI, K. "Spectrum agile radios: utilization and sensing architectures". In: New Frontiers in Dynamic Spectrum Access Networks, 2005. DySPAN 2005. 2005 First IEEE International *Symposium on*, pp. 160 –169, março 2005. Disponível em:< http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1542631> Acesso em: 30 de março de 2013.

[25] ZHAO, Q., GEIRHOFER, S., TONG, L., et al. "Optimal Dynamic Spectrum Access via Periodic Channel Sensing". In: *Wireless Communications and Networking Conference, 2007.WCNC 2007. IEEE*, pp. 33 –37, março 2007. Disponível em:<http://scholar.google.com.br/scholar?q=Optimal+Dynamic+Spectrum+Access+via+Periodic+Channel+Sensing&hl=ptBR&as_sdt=0&as_vis=1&oi=scholart&sa=X&ei=CR0WU7mEDc75kQeI0YHICQ&ved=0CCoQgQMwAA> Acesso em: 30 de março de 2013.

[26] CABRIC, D., MISHRA, S., BRODERSEN, R. "Implementation issues in spectrum sensing for cognitive radios". In: Signals, Systems and Computers, 2004. Conference Record of the Thirty-Eighth Asilomar Conference on, Disponível:<http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1399240> Acesso em: 30 de março de 2013.

[27] Khambekar, N., Dong, L., Chaudhary, V. "Utilizing OFDM Guard Interval for Spectrum Sensing". In: *Wireless Communications and* Networking Conference, 2007.WCNC 2007. IEEE, pp. 38 –42, março 2007. doi: 10.1109/WCNC.2007.13. Disponível em:<http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1399240 > Acesso em: 2 de abril de 2013.

[28] Peter Steenkiste, Douglas Sicker, Gary Minden, Dipankar Raychaudhuri "Future Directions in Cognitive Radio Network Research", NSF Workshop Report, 9 de março de 2009. Disponível em:<https://www.cs.cmu.edu/~prs/NSF_CRN_Report_Final.pdf> Acesso em: 2 de abril de 2013.

[29] KARAGIANNIDIS, G., LIOUMPAS, A. "An Improved Approximation for the Gaussian Q-Function", *Communications Letters, IEEE*, v. 11, n. 8, pp. 644 –646, agosto 2007. ISSN: 1089-7798. Disponível em:<https://translate.google.com.br/?hl=pt-BR&tab=wT#es/pt/marzo%20%0AJunio%0Anoviembre%20%0Aabril%20%0Aagosto> Acesso em: 2 de abril de 2013.

[30] Y.-C. Liang, Y. Zeng, E. Peh, A.T. Hoang, "Sensing-throughput tradeoff for cognitive radio networks", IEEE Transactions on Wireless Communications 7 de abril de 2008. Disponível em:< http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4489760&tag=1> Acesso em: 3 de abril de 2013.

[31] LEHTOMAKI, J. J., JUNTTI, M., SAARNISAARI, H., et al. "Threshold setting strategies for a quantized total power radiometer", IEEE Signal Processing Letters, nov. 2005. Disponível em:<http://www.ee.oulu.fi/~jannel/PDFs/lehtomaki_SPL05QUAN.pdf> Acesso em: 3 de abril de 2013.

[32] Bruno A. Olhausen "Aliasing", PSC129 – Sensory Processes, October 10,2000.

[33] IEEE std. 802.22/D1.0, "Draft Standard for Wireless Regional Area Networks
Part 22: Cognitive Wireless RAN Medium Access Control (MAC) specifications: Policies and procedures for operation in the TV bands," abril 2008. Disponível:<https://standards.ieee.org/develop/project/802.22a.html> Acesso em: 4 de abril de 2013.

[34] J. Notor, "The evolution of spectrum sharing in the IEEE 802.22 WRAN standards process," fevereiro 2006. [Online]. Disponível em:< http://www.eecs.berkeley.edu/_dtse/3r notor.ppt> Acesso em: 4 de abril de 2013.

[35] J. Unnikrishnan and V. Veeravalli, "Cooperative sensing for primary detection in cognitive radio" IEEE J. Sel. Topics in Sig. Proc., vol. 2, no. 1, pp. 18–27, fevereiro. 2008. Disponível em:< http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4453896> Acesso em: 10 de abril de 2013.

[36] J. G. Proakis and D. G. Manolakis, "Digital Signal Processing", 3rd ed.Prentice-Hall, 1996.

[37] H. Stark and J. W. Woods, "Probability and Random Processes with Applications to Signal Processing", 3rd ed. Prentice-Hall, 2002.

[38] C. W. Therrien, "Discrete Random Signals and Statistical Signal Processing. Prentice-Hall", 1992.

[39] Wikipédia a enciclopédia livre "The Cognitive Radio History". Disponível em: <http://en.wikipedia.org/wiki/Cognitive_radio>. Acesso em: 12 de abril de 2013.

[40] Francisco Alberto Sandoval Noreña. "Novos Receptores com Posto Reduzido e suas Aplicacoes em sistemas Baseados em DS-CDMA". Dissertação de mestrado, fevereiro de 2013. Disponível em:<http://www2.dbd.puc-rio.br/pergamum/biblioteca/php/mostrateses.php?open=1&arqtese=1112776_2013_Indice.html>. Acesso em: 22 de abril de 2013.

[41] Rodrigo C. de Lamare e Raimundo Sampaio-Neto. "Adaptive Reduced-Rank Processing Based on Joint and Interative Interpolation, Decimation and Filtering" Disponível em:<http://www-users.york.ac.uk/~rcdl500/adrrank_tsp_final.pdf>. Acesso em: 22 de abril de 2013.

[42] C. V. Rodríguez R., "Caracterização do Canal Rádio em Banda Larga na faixa de 3,5GHz em Ambiente Urbano", Doctorate Dissertation, Pontific Catholic University of Rio de Janeiro, Rio de Janeiro, Sept. 2009.

[43] L. A. R. S. Mello, C. V. Rodríguez R., "Propagation Meassurements at 3.5 GHz in a Dense Urban Area", The 4th European Conference on Antennas and Propagation EUCAP , Apr. 2010.

[44] C. V. Rodríguez R., L. A. R. S. Mello, M. Pontes, "Path Loss and Delay Spread Characteristics of Mobile WiMAX Channel in an Urban Area", MOMAG, Aug. 2010.

[45] S. J. Shellhammer, "Spectrum sensing in IEEE 802.22," in Proc., IAPR Wksp. Cognitive Info. Processing, Junho de 2008, pp. 1–6. Disponível em:< http://www.eurasip.org/Proceedings/Ext/CIP2008/papers/1569094657.pdf>. Acesso em: 22 de abril de 2013.

[46] T. Yucek and H. Arslan, "A survey of spectrum sensing algorithms for cognitive radio applications," IEEE Comm. Surveys & Tutorials, vol. 11,no. 1, pp. 116–130, Janeiro 2009. Disponível em:< http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4796930&tag=1>. Acesso em: 22 de abril de 2013.

[47] C. V. Rodríguez R., L. A. R. S. Mello, M. Pontes, "Experomental Evaluation of time Domain Energy Detection for Cognitive Radio Applications", IMOC, Oct. 29 2011-Nov. 1 2011.

[48] A. Sonnenschein and P. M. Fishman, "Radiometric detection of spreadspectrum signals in noise of uncertain power" IEEE Trans. on Aerospace and Electronic Systems, vol. 28, no. 3, pp. 654–660, julho 1992. Disponível em:<http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4796930&tag=1>. Acesso em: 25 de abril de 2013.

[49] R. Tandra and A. Sahai, "SNR walls for signal detection" IEEE J. Sel. Topics in Sig. Proc., vol. 2, no. 1, pp. 4–17, Feb. 2008. Disponível em:< http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4453895>. Acesso em: 25 de abril de 2013.

[50] H. Urkowitz, "Energy detection of unknown deterministic signals" IEEE, vol. 55, no. 4, pp. 523–531, Apr. 1967. Disponível em:< http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1447503>. Acesso em: 27 de abril de 2013.

[51] I.S.REED “On a Moment Theorem for Complex Gaussian Processes”, IRE Transaction on information Theory. Disponível em:< http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1057719&userType=inst>. Acesso em: 27 de abril de 2013.

[52] Informação SDR “SDR – Software defined radio”. Disponível em:< http://www.ajmesquita.qsl.br/?page_id=344>. Acesso em: 30 de abril de 2013.

[53] BLOSSOM, E. “Exploring GNU Radio”. Disponível em:<http://gnuradio.org/redmine/projects/gnuradio/activity?from=2009-03-11>. Acesso em: 30 de abril de 2013.

[54] The free & open software radio ecosystem “GNU Radio”. Disponível em:<http://gnuradio.org/redmine/projects/gnuradio/wiki>. Acesso em: 30 de abril de 2013.

[55] A national Instruments Company “Ettus Research LLC”. Disponível em:<http://www.ettus.com/. Acessado em dez. 2010. >. Acesso em: 2 de maio de 2013.

[56] Ettus Research, A national Instruments Company “Universal software radio peripheral”. Disponível em:<http://www.ettus.com/home>. Acesso em: 2 de maio de 2013.

A
Modelo Analítico e Estatístico.

A.1
Para o Detector de Energia (ED)

Para a Hipótese H_0, apresentamos a obtenção da média $\mu_0 = \sigma_W^2$

A variável de teste estatístico para o detector de energia apresentado na equação (3.6) onde para a obtenção da média dentro a hipótese H_0 é representada da seguinte forma:

$$E[T(x)] = \frac{1}{N}\sum_{n=1}^{N} E[|w(n)|^2] \qquad (A.1)$$

Para a variância da variável complexa $E[|w(n)|^2] = \sigma_w^2$

$$T(x) = \sigma_w^2 \frac{1}{N}\sum_{n=1}^{N} 1 \qquad (A.2)$$

Pelo somatório obtemos:

$$T(x) = \sigma_w^2 \frac{1}{N} * \mathrm{N} \qquad (A.3)$$

Finalmente

$$T(x) = \sigma_w^2 \qquad (A.4)$$

Para a Hipótese H_0 apresenta a obtenção da variância $\sigma_0^2 = \frac{\sigma_w^4}{N}$

A variável de teste estatístico do detector de energia para a obtenção da variância, dentro da hipótese ($\boldsymbol{H_0}$), para uma variável real é representada da seguinte forma:

$$\sigma_w^2 = E[T(x)^2] - m_x^2 \tag{A.5}$$

Onde

$$E[T(x)^2] = \frac{1}{N^2}\sum_{n_1=1}^{N}\sum_{n_2=1}^{N} E[|w(n_1)|^2|w(n_2)|^2] \tag{A.6}$$

$$m_x^2 = (\sigma_w^2)^2 \text{ (mostrado no passo anterior)} \tag{A.7}$$

Da equação (A. 6) temos:

$$E[T(x)^2] = \frac{1}{N^2}\sum_{n_1=1}^{N}\sum_{n_2=1}^{N} E[|w(n_1)|^2|w(n_2)|^2]$$

Por propriedades sabemos que: $|w(n)|^2 = w(n)w^*(n)$ então temos:

$$E[T(x)^2] = \frac{1}{N^2}\sum_{n_1=1}^{N}\sum_{n_2=1}^{N} E[w(n_1)w^*(n_1)w(n_2)w^*(n_2)] \tag{A.8}$$

Aplicar o teorema da referencia [51]:

$$E[T(x)^2] = \frac{1}{N^2}\sum_{n_1=1}^{N}\sum_{n_2=1}^{N} E[w(n_1)w^*(n_1)]E[w(n_2)w^*(n_2)] + E[w(n_1)w^*(n_2)]E[w(n_2)w^*(n_1)] \tag{A.9}$$

Por conveniência, separamos a equação em duas partes:

Primeira parte:

$$\sum_{n_1=1}^{N}\sum_{n_2=1}^{N} E[w(n_1)w^*(n_1)]E[w(n_2)w^*(n_2)] \qquad (A.10)$$

Segunda parte:

$$\sum_{n_1=1}^{N}\sum_{n_2=1}^{N} E[w(n_1)w^*(n_2)]E[w(n_2)w^*(n_1)] \qquad (A.11)$$

Usamos a propriedade $E[w(n)\,w^*(n)] = \sigma_w^2$, para o cálculo da primeira parte $(A.10)$.

$$\sum_{n_1=1}^{N}\sum_{n_2=1}^{N} \sigma_w^2\sigma_w^2 = N^2\sigma_w^4 \qquad (A.12)$$

Agora, para o cálculo da segunda parte $(A.11)$, se for $n_1 \neq n_2 = 0$ temos:

$$\sum_{n_{1,2}}^{N} \sigma_w^2\sigma_w^2 = 2N\sigma_w^4 \qquad (A.13)$$

Substituindo as equações (A.12) e (A.13) na equação $(A.6)$, temos então:

$$E[T(x)^2] = \frac{1}{N^2} * (N^2\sigma_w^4 + 2N\sigma_w^4) \qquad (A.14)$$

$$E[T(x)^2] = \sigma_w^4 * \frac{N+2}{N} \qquad (A.15)$$

Finalmente, substituímos em (A. 5) o resultado de (A.11) e (A. 4)

$$\sigma_w^2 = E[T(x)^2] - m_x^2 \quad \text{(A. 16)}$$

$$\sigma_w^2 = \sigma_w^4 * \frac{N+2}{N} - \sigma_w^4 \quad \text{(A. 17)}$$

$$\sigma_w^2 = \frac{2\sigma_w^4}{N} \quad \text{(A. 18)}$$

Relação Sinal Ruído

$$\gamma = hs + w$$

Para SNR temos

$$\gamma = \frac{E|hs|^2}{E|n^2|} = |h|^2 \frac{\sigma_s^2}{\sigma_w^2} \quad \text{(A. 19)}$$

Para a Hipótese H_1, apresentamos a obtenção da média $\mu_0 = \sigma_w^2(\gamma+1)$.

A variável de teste estatístico para o detector de energia apresentada na equação (3.3), onde a obtenção da média dentro da hipótese H_1 é representada da seguinte forma:

$$T(x) = \frac{1}{N}\sum_{n=1}^{N}[|h(n)s(n) + w(n)|^2] \quad \text{(A. 20)}$$

Por propriedades, sabemos que: $|w(n)|^2 = w(n)w^*(n)$, então, temos:

$$T(x) = \frac{1}{N}\sum_{n=1}^{N}\big(h^*(n)s^*(n) + w^*(n)\big)\big(h(n)s(n) + w(n)\big) \quad \text{(A. 21)}$$

Aplicando o teorema da referencia [51]:

$$T(x) = \frac{1}{N}\sum_{n=1}^{N}[|h^*(n)|^2 s^*(n)s(n)] + [h^*(h)s^*(n)w(n)] + [h(n)s(n)w(n)] + [w^*(n)w(n)] \quad (A.22)$$

Sabemos que $E[w(n)] = 0$ *e* $E[w^*(n)] = 0$ temos:

$$T(x) = \frac{1}{N}\sum_{n=1}^{N} E[|h^*(n)|^2 s^*(n)s(n)] + E[w^*(n)w(n)] \quad (A.23)$$

Agora sabemos que h(n) = h, então.

$$T(x) = \frac{1}{N} * N * \ |h^2|\sigma_s^2 + \sigma_w^2 \quad (A.24)$$

$$T(x) = \sigma_w^2 \ \left(\frac{|h^2|\sigma_s^2}{\sigma_w^2} + 1\right) \quad (A.25)$$

Sabemos o valor de γ da equação (13)

$$\gamma = \frac{|h^2|\sigma_s^2}{\sigma_w^2} \quad (A.26)$$

Substituindo na equação (A.25), obtemos:

$$T(x) = \sigma_w^2 \ \ (\gamma + 1) \quad (A.27)$$

Função $\boldsymbol{Q}(.)$

A função $Q(.)$ é definida, para uma distribuição normal com media zero e variância um [29]:

$$Q(x) = \frac{1}{\sqrt{2\pi}}\int_x^{\infty} \exp\left(-\frac{z^2}{2}\right) dz \quad (A.28)$$

O cálculo de $Q\left(\frac{x-\mu}{\sigma}\right)$ é feito para uma distribuição normal com média μ e variância σ^2 quaisquer.

Limiar

$$\bar{\tau} = \frac{\sigma_w^2}{\sqrt{N}} Q^{-1}(\bar{P}_f) \qquad (A.29)$$

Probabilidade de Falso Alarme

$$P_f(\tau) = Prob\{T(x) > \tau; H_o\} = Q\left(\left(\frac{\tau}{\sigma_w^2} - 1\right)\sqrt{N}\right) \qquad (A.30)$$

Probabilidade de Detecção

$$P_d(\tau) = Prob\{T(x) > \tau; H_1\} = Q\left(\left(\frac{\tau}{\sigma_w^2} - \gamma - 1\right)\sqrt{\frac{N}{2\gamma+1}}\right) \qquad (A.31)$$

A.2
Para o Detector do valor absoluto de covariância (CAV)

Matriz Toeplitz

Em álgebra linear, uma matriz de Toeplitz, nomeado em homenagem a Otto Toeplitz, é uma matriz quadrada com toda a sua diagonal da esquerda para a direita. Apresenta a seguinte estrutura:

$$T = \begin{pmatrix} a & b & c & d & k \\ f & a & b & c & d \\ g & f & a & b & c \\ h & g & f & a & b \\ j & h & g & f & a \end{pmatrix}$$

Em termos matemáticos:

$$\forall \quad a_{i,j} \in T \rightarrow a_{i,j} = a_{i+1,j+1}$$

B
Rádio Definido por *Software* (*SDR – Software Defined Radio*)

Um sistema de rádio definido por *software*, ou SDR, é um sistema de comunicação sem-fio onde componentes normalmente implementados em *hardware* (por exemplo, misturadores, filtros, amplificadores, moduladores / demoduladores, detectores, etc.) são implementadas por meio de *software* em um computador pessoal ou dispositivo de computação incorporado. Embora o conceito de SDR não seja novo, a capacidade de rápida evolução da eletrônica digital torna muitos processos práticos, que costumavam ser apenas teoricamente possíveis.

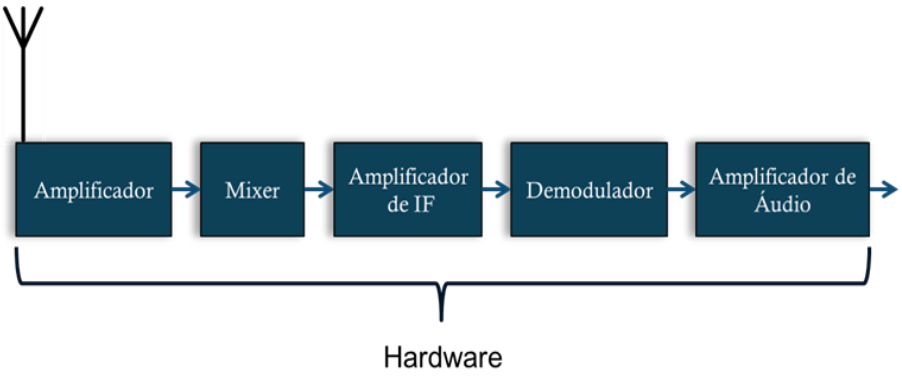

Figura B.1: Componentes do *Hardware*.

Um sistema de SDR básico pode ser constituído por um computador pessoal equipado com uma placa de som ou um conversor analógico-digital, acoplado a um dispositivo que gere RF como é mostrado na Figura B.2 Operações significativas de processamento de sinal são realizadas pelo processador em vez de através de componentes de *hardware* específicos. Isto permite criar um rádio que pode receber e transmitir protocolos diferentes (por vezes referidos como formas de onda) com base, unicamente, no *software* utilizado.

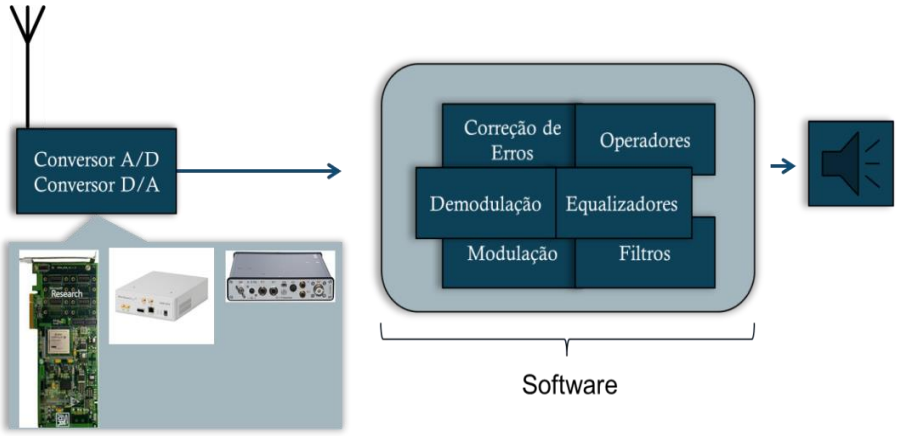

Figura B.2: Arquitetura do SDR.

Embora o SDR possa ter parte de suas funcionalidades implementadas por *hardware* por motivos de desempenho, a grande vantagem desses equipamentos está na possibilidade de integrar e configurar diferentes parâmetros e mecanismos apenas alterando a programação de seu *software*.

O rádio definido por *software* tem uma utilidade significativa para os serviços de telefonia militar e célular, oferecendo uma grande variedade de protocolos de rádio mudando em tempo real.

Um rádio definido por *software* pode ser flexível o suficiente para evitar o congestionamento do espectro através da implementação de tecnologias como:

- *Spread spectrum* e técnicas *ultrawideband*, que permitem a vários transmissores operar em no mesmo local e na mesma frequência, com pouca interferência, tipicamente combinando uma ou mais técnicas de detecção de erros e de correção de erros causados pela interferência;
- Antenas adaptativas, que permitem aos receptores rejeitar a interferência de outras direções e, assim, detectar transmissões mais fracas;
- Técnicas de rádio cognitivo;
- Ajuste dinâmico de potência no transmissor, com base em informações transmitidas dos receptores, diminuindo energia ao mínimo necessário e reduzindo o problema perto-distante e a interferência.

- Rede *mesh* sem fio, onde cada rádio adicionado aumenta a capacidade total e reduz a potência necessária em qualquer nó [52].

- **Vantagens de utilizar SDR.**
  - Flexibilidade.
  - Redução do tempo de desenvolvimento.
  - Baixo custo.
    - Poucas unidades.
  - Inovações:
    - Novas técnicas de modulação.
    - Acesso dinâmico ao espectro.
    - Rádios Cognitivos.
      - ✓ SDR oferecem a flexibilidade necessária para implementar um rádio cognitivo.
      - ✓ SDR é um habilitador para rádio cognitivo.
  - SDR é perfeito para desenvolvimento de novos protocolos de camada física.
  - SDR reduz o tempo de desenvolvimento de produtos.
  - SDR possibilita o desenvolvimento de Rádios Cognitivos.

Existem diversas soluções no mercado, mas uma das mais utilizadas hoje é a solução que engloba o *GNU Radio* e o *Universal Software Radio Peripheral* (USRP). A maior vantagem dessa solução é o fato de ela ser totalmente aberta (*opensource*), e por este motivo foi a solução utilizada nesta dissertação.

B.1
GNU Radio

GNU Radio é um kit de ferramentas de desenvolvimento de software livre que fornece os blocos de processamento de sinal para implementar rádios definidos por *software*, utilizando equipamentos de rádio de baixo custo e computadores pessoais. Combina um grande conjunto de blocos DSP (*Digital Signal Processor*) de construção fácil de integrar e curto tempo de execução.

Pode ser usado com *hardware* de RF externo de baixo custo imediatamente disponíveis para criar rádios definidos por *software* ou mesmo sem *hardware* em um ambiente de simulação. Os aplicativos *GNU Radio* são desenvolvidos principalmente usando a linguagem de programação *Python* para o código de alto nível, utilizado para a criação de fluxos de dados (*FlowGraphs*), enquanto o processamento de sinal é implementado em C + + (mais rápido e eficiente) usando o processador de extensões de ponto flutuante e aumentando o desempenho.

A integração entre as linguagens C++ e *Python* é feita utilizando a ferramenta SWIG, o *Simplified Wrapper and Interface Generator*. Desta forma é possível implementar sistemas de rádio de alta capacidade e desempenho que utilizam a eficiência e a rapidez da linguagem C++ com a simplicidade de um ambiente de desenvolvimento de aplicações com a linguagem *Python*. A Figura 2.11 esquematiza o funcionamento do *GNU Radio*, seus componentes e suas diferentes linguagens, quando utilizado com dispositivos USRP.

Embora a princípio o *GNU Radio* não seja uma ferramenta de simulação, essa ferramenta dá suporte ao desenvolvimento de algoritmos de processamento de sinais utilizando dados previamente armazenados ou gerados por *software* e permitindo seu uso sem o *hardware* de radio.

O *GNU Radio* fornece ao desenvolvedor uma biblioteca de blocos de processamento de sinal, além da estrutura necessária para interconectá-los. Um sistema de SDR é implementado na forma de um grafo em que os vértices representam os blocos de processamento de sinais e as arestas representam o fluxo de dados entre os mesmos [53].

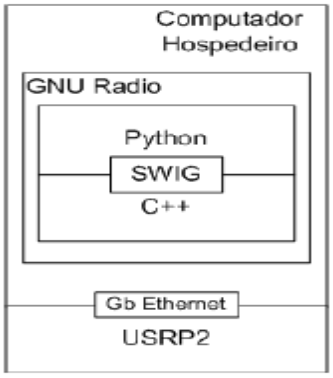

Figura B.3: Esquema do funcionamento da plataforma GNU Radio em conjunto com o Universal Software Radio Peripheral.

Em geral, os blocos de processamento de sinais são implementados na linguagem C++ e os grafos são construídos com a linguagem *Python*, efetivamente interconectando os blocos, mas há esforços recentes entre a equipe de desenvolvimento do *GNU Radio* para permitir que a programação com a ferramenta seja realizada inteiramente em C++, sem impedir que a linguagem *Python* também seja utilizada [54]. Além de realizar a criação e configuração dos grafos é possível utilizar os recursos da linguagem de programação para se adicionar qualquer tipo de funcionalidade aos programas criados, como a execução condicional de diferentes grafos, a leitura periódica de informações para apresentação na tela ou para armazenamento em arquivos, etc.

Conceitualmente, os blocos do *GNU Radio* processam fluxos infinitos de dados que seguem de suas portas de entrada para suas portas de saída. Certos blocos possuem somente portas de entrada ou somente portas de saída. Eles funcionam como fontes (*sources*) ou sumidouros (*sinks*) de dados no grafo. As fontes podem ser blocos que geram um sinal ou um ruído no *GNU Radio*, leem o conteúdo de um arquivo ou recebem uma transmissão de rádio. Já os sumidouros podem escrever em um arquivo, apresentar informações em um mostrador (*display*) gráfico ou realizar uma transmissão de rádio.

Atualmente, o *GNU Radio* conta com mais de 300 blocos de processamento em sua biblioteca. Dentre as funcionalidades oferecidas por esses blocos, estão: filtros, conversão de tipos, modulação/demodulação de sinais, operações matemáticas, equalização, codificação/decodificação de voz e dados, sincronização, transformadas como a transformada de Fourier, interfaces para leitura e escrita de dados em arquivos, interfaces gráficas para análise em tempo real, entre outras. Além disso, há blocos que realizam a interface com dispositivos de *hardware* como placas de áudio USRP e USRP2, tanto para recepção ou captura, quanto para transmissão.

Os dados transmitidos entre os blocos do grafo podem assumir qualquer formato, desde que possuam um tipo de dados que possa ser definidos na linguagem C++. Na prática, os tipos de dados mais comuns são valores complexos ou reais inteiros ou de ponto flutuante, já que na maior parte do tempo os fluxos de dados pelos grafos serão formados por amostras ou bits [54]. Ademais, é possível configurar fluxos de vetores de dados entre os blocos dos grafos, que efetivamente funcionam como múltiplos fluxos distintos que são processados em

paralelo pelos blocos do GNU Radio. A Figura B.4 mostra os blocos de processamento do GNU Radio.

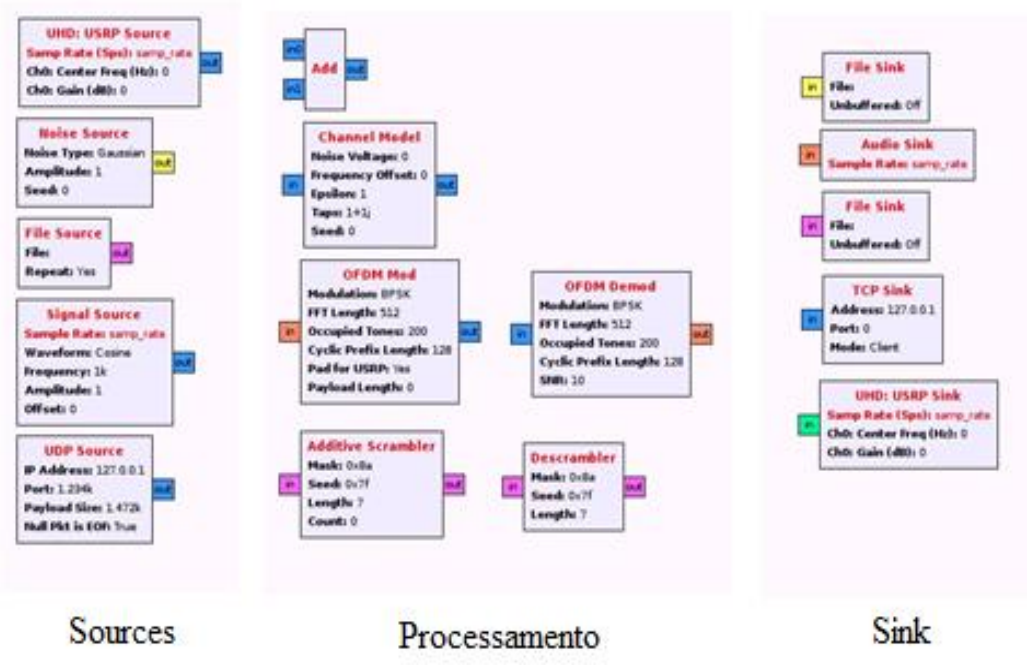

Figura B.4: Blocos de processamento do GNU Radio.

O *GNU Radio* é compatível com diversos tipos de *hardware* para sua operação. Virtualmente, qualquer equipamento que realize conversão analógica-digital ou digital analógica, pode ser usado para desenvolver um SDR. Há compatibilidade com placas de áudio comuns de computadores pessoais, tanto para produzir quanto para analisar amostras de som e a possibilidade de se usar placas conversoras analógico-digitais que utilizam a interface PCI (*Peripheral Component Interconnect*) para realizar a ponte entre o módulo de radiofrequência (que de fato sintoniza na frequência a ser utilizada) e computadores pessoais [53]. A família de dispositivos USRP foi desenvolvida pela *Ettus Research LLC* especificamente para operar com o *software GNU Radio*, embora existam adaptações para outras plataformas como *LabVIEW* e *Simulink* [55].

O *GNU Radio* é distribuído com uma ferramenta de interface gráfica para a criação de seus grafos, o *GNU Radio Companion* ou GRC. O GRC apresenta uma interface simples em que os grafos são criados ligando-se graficamente os blocos do *GNU Radio* para os quais há suporte. Uma vez criado e configurado o grafo de execução, há recursos para se gerar o código - fonte na linguagem *Python* e

executá-lo através do próprio GRC, contudo, há algumas limitações nessa ferramenta: nem todos os blocos de processamento de sinais do *GNU Radio* estão disponíveis para utilização no GRC e não há como adicionar funcionalidades programadas em *Python* diretamente pela ferramenta. A utilidade da ferramenta, entretanto, ainda pode ser muito bem aproveitada, uma vez que essas limitações podem ser superadas facilmente editando diretamente o código *Python* gerado, após criar uma configuração inicial do grafo no GRC.

B.2
Universal Software Radio Peripheral (USRP)

O *Universal Software Radio Peripheral*, ou USRP, é um dispositivo desenvolvido pelo *Ettus Research LLC* [56] que transforma computadores comuns em plataformas flexíveis de SDR. Suas principais vantagens são a alta flexibilidade e um baixo custo. O USRP é composto por duas partes: a placa-mãe (*motherboard*), responsável pelas funções mais complexas, e as placas-filhas (*daughterboards*), que contêm o módulo RF.

O coração do USRP é sua placa-mãe, com quatro ADC/DCA de alta velocidade e um *Field Programáble Gate Array* (FPGA) Altera EP1C12 *Cyclone*. Os ADC/DAC são conectados às *daughterboards* (muitas vezes referidas como *Front Ends*), enquanto o FPGA é conectado a um *chip* de interface UCB2 para a conexão com um computador comum, conforme mostrado na Figura 2.12.

O USRP possui várias funções, tais como digitalização do sinal de entrada, sintonização dentro da banda IF e redução da taxa de amostragem antes que os dados (na banda base) sejam mandados para o computador via interface USB. Funções inversas às citadas são fornecidas para a transmissão. A Figura B.5 mostra o diagrama de blocos da USRRP.

O princípio fundamental do USRP são tarefas divididas entre FPGA e o computador hospedeiro. O computador é responsável pelo processamento mais específico como, por exemplo a modulação e a demodulação de sinais, a codificação e decodificação, etc. Já o processamento de alta velocidade, como decimacão e interpolação, ocorre no FPGA.

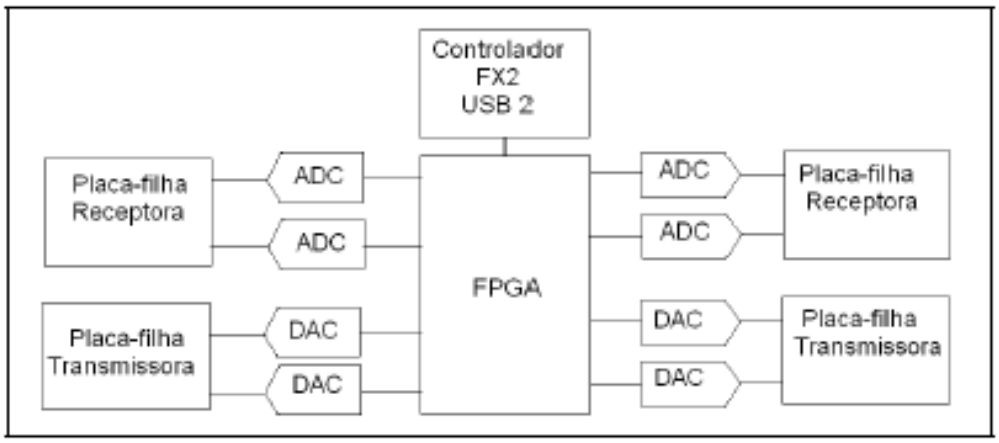

Figura B.5: Diagrama simplificado de blocos do USRP.

O USRP também oferece suporte ao recurso *multiple-input and multiple-output* (MIMO). Assim ele pode utilizar múltiplas antenas, simultaneamente, para a recepção e a transmissão de sinais.

As *daughterboards* são responsáveis por sintonizar a faixa do espectro que interessa ao projetista e transladá-la para uma banda intermediária ou banda base, para minimizar o efeito do *aliasing*. Com sua utilização é possível cobrir diferentes faixas do espectro, sendo necessária apenas a troca de uma *daughterboard* por outra.

B.3
Resultados obtidos na simulação usando a USRP.

Os resultados mostrados a seguir foram obtidos no curso: ¨Treinamento em *Software GNU Radio* e *Hardware USRP*¨ do dia 02 a 06/09/2013, que foi realizado em INMETRO com o projeto PLATCOG – Plataforma teórica e experimental de Sistemas Rádio Cognitivo para Subsidiar aspectos normativos e regulatórios.

O projeto PLATCOG em curso no INMETRO tem como objetivo básico o desenvolvimento de uma *testbed* para investigação das redes oportunistas, que permitam o compartilhamento de banda de frequência entre diversos usuários, envolvendo emprego de novos esquemas de acesso rádio, baseados em técnicas cognitivas ou de percepção do ambiente radioelétrico.

Redes de rádio cognitivos (*cognitive radio networks* - CRNs) se referem a redes de rádios adaptativas e auto organizáveis, as quais são capazes de responder a alterações ambientais tais como interferências, densidade de aparelhos e requisitos das aplicações dos usuários. CRNs com capacidade de acesso dinâmico ao espectro têm o potencial de melhorar de forma dramática a eficiência espectral, com o decorrente aumento de capacidade para novas redes e aplicações.

CRNs são sistemas extremamente complexos e embora o desenvolvimento de cada um de seus componentes seja importante, um grande desafio é conceber como se comportam e trabalham juntos sobre condições reais. Para responder a este desafio, é fundamental o desenvolvimento de plataformas de teste (*testbeds*) que possam ser empregadas para avaliar redes cognitivas em vários estágios de seu desenvolvimento.

O treinamento foi ministrado por J.M. Corgan de Corgan Labs, o *Corgan Labs* (anteriormente nomeado *Corgan Enterprises*) foi fundado em 2006 por J.M. Corgan, com o objetivo específico de atender ao mercado mundial na área de rádio definido por *software*.

Apresentamos, nos itens seguintes os trabalhos desenvolvidos e resultados obtidos no transcurso da semana do curso:

B.3.1
O tom de discagem (*Dial Tone*)

O *dial tone* foi composto pela inserção de dois *Signal Sources* com uma taxa de amostragem de 32 Kbps (dois tons de interferência 350 Hz e 440 Hz), na primeira Fonte do Sinal (*Signal Source*) a frequência é 350 Hz e na segunda fonte do sinal a frequência é 440 Hz. Incluímos também, ruído (*Noise Source*), usamos um somador para adicionar os três sinais (Add) para depois terminar no *áudio sink* como é mostrado na Figura B.6.

Adicionalmente este exemplo é considerado dos blocos WX GUI Slider que permitem controlar a frequência e a amplitude do ruído.

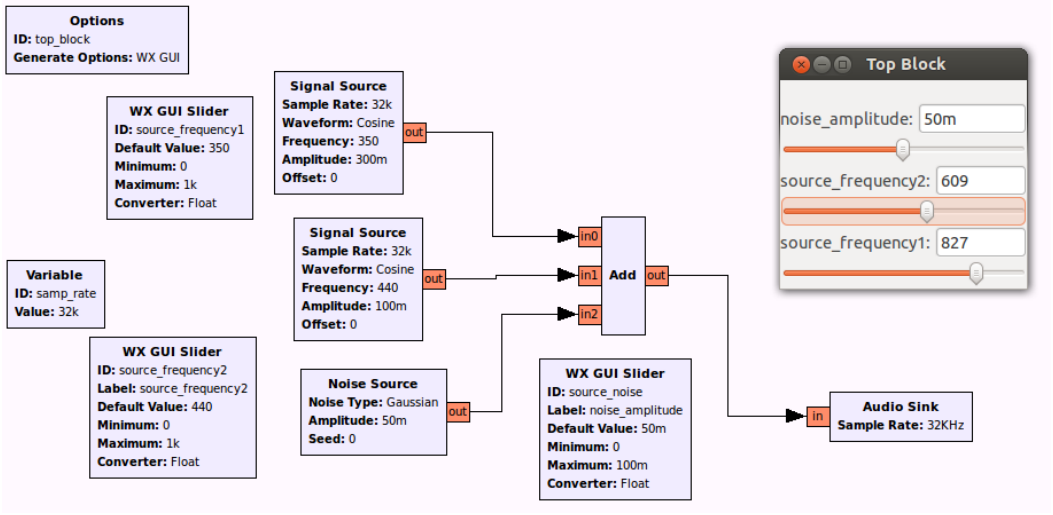

Figura B.6: Dial Ton (GNU-Radio).

B.3.2
Geração do Sinal

Neste trabalho, foram desenvolvidos dois grupos de blocos, um para a transmissão e o outro para a recepção.

Para a transmissão.

O objetivo é criar um transmissor com a ferramenta *GNU-Radio*, como no diagrama de blocos mostrado na Figura B.7.

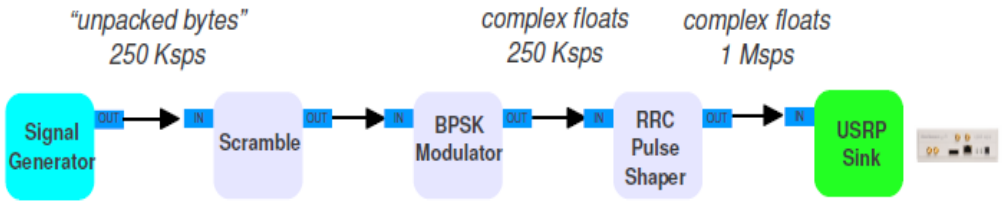

Figura B.7: Diagrama para a transmissão.

A princípio 250 [Ksps] é a taxa de amostragem e, depois do formador de pulso (pulse shaper), é incrementada para 1 [Msps].

Na prática, os blocos desejados são introduzidos no *GNU-Radio Companion*. Geramos um sinal com frequência central de 917 MHz, com faixa de 80 kHz, como é mostrado na Figura B.13.

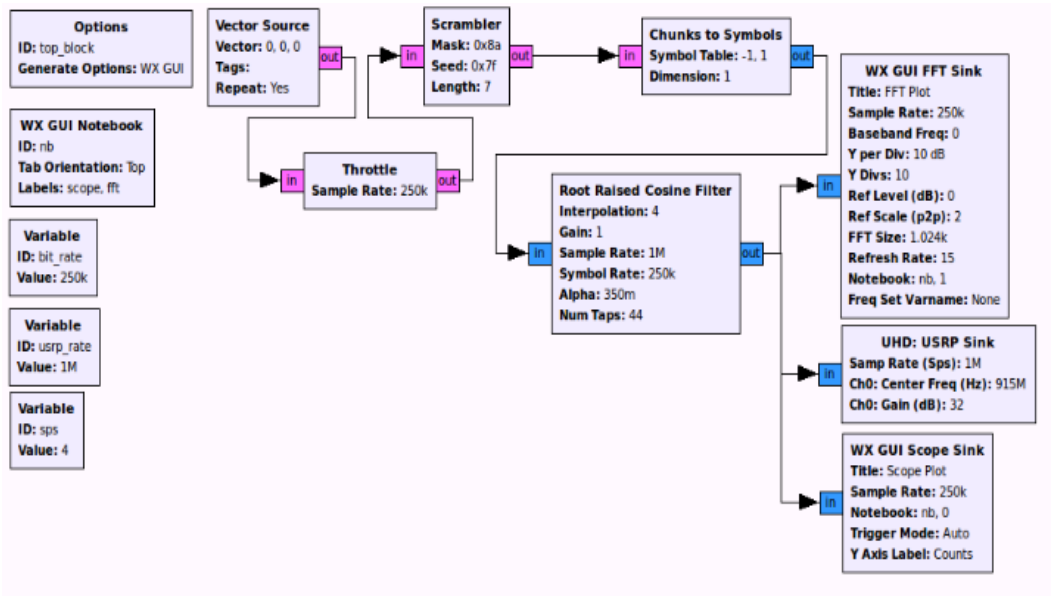

Figura B.8: Diagrama de Blocos (GNU-Radio) para a transmissão.

Para a recepção.

Nosso diagrama da recepção é mostrado na Figura B.9.

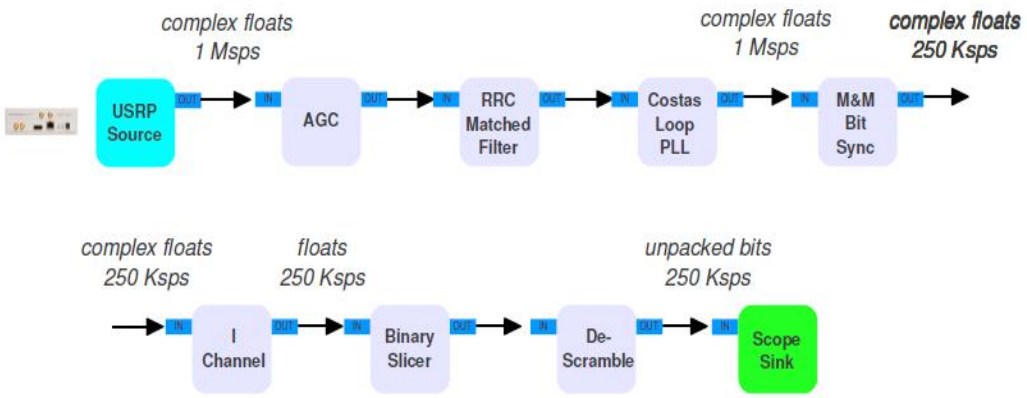

Figura B.9: Diagrama de Blocos na Recepção.

Na prática, geramos um conjunto de blocos que permite mostrar o sinal recebido. Uma das coisas que mais se destaca é a diminuição da taxa de amostragem chegada á USRP como 1 [Msps] e depois da recuperação relógio digital (*digital clock recovery*) tem-se 250 [Ksps].

O diagrama de blocos experimental (no *GNU-Radio Companion*) é apresentado na Figura B.11.

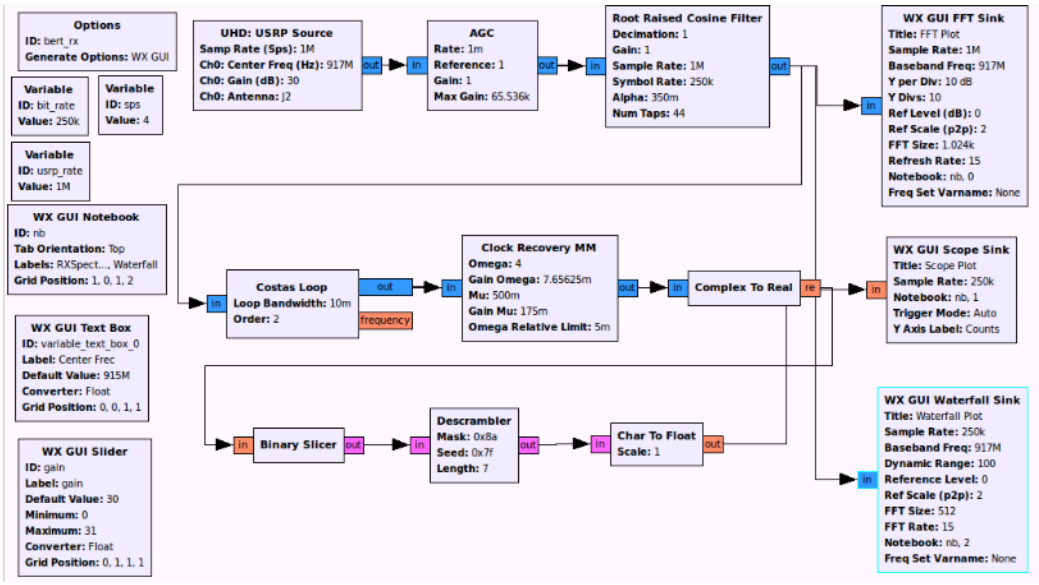

Figura B.10: Diagrama de Blocos (GNU-Radio) para a recepção.

Obtivemos três resultados do diagrama de recepção. Um deles é a Figura B.11 obtida pelo bloco WX GUI FFT *Sink*, o qual mostra o sinal recebido no domínio da frequência, com as configurações realizadas previamente.

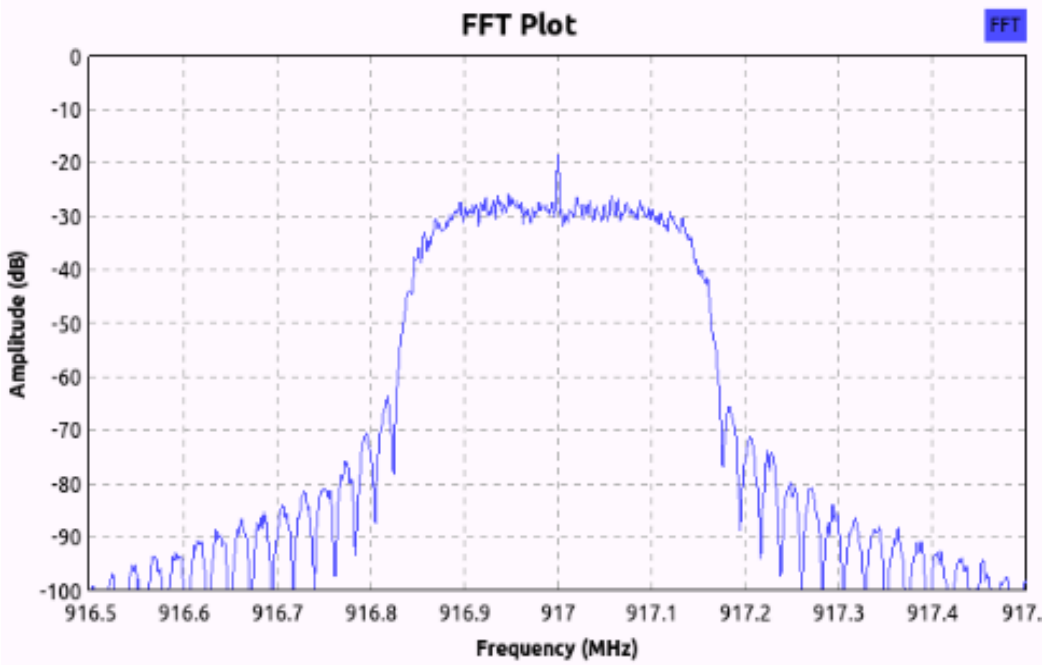

Figura B.11: Sinal no domínio da Frequência (GNU-Radio).

A Figura B.12 mostra quando o canal está sendo usado pelo primeiro usuário (quando os valores se mostram em 1) e quando este deixa de transmitir

(quando os valores se mostram em 0). O osciloscópio é uma das funções que tem o *GNU-Radio*, e mostra o sinal no domínio do tempo.

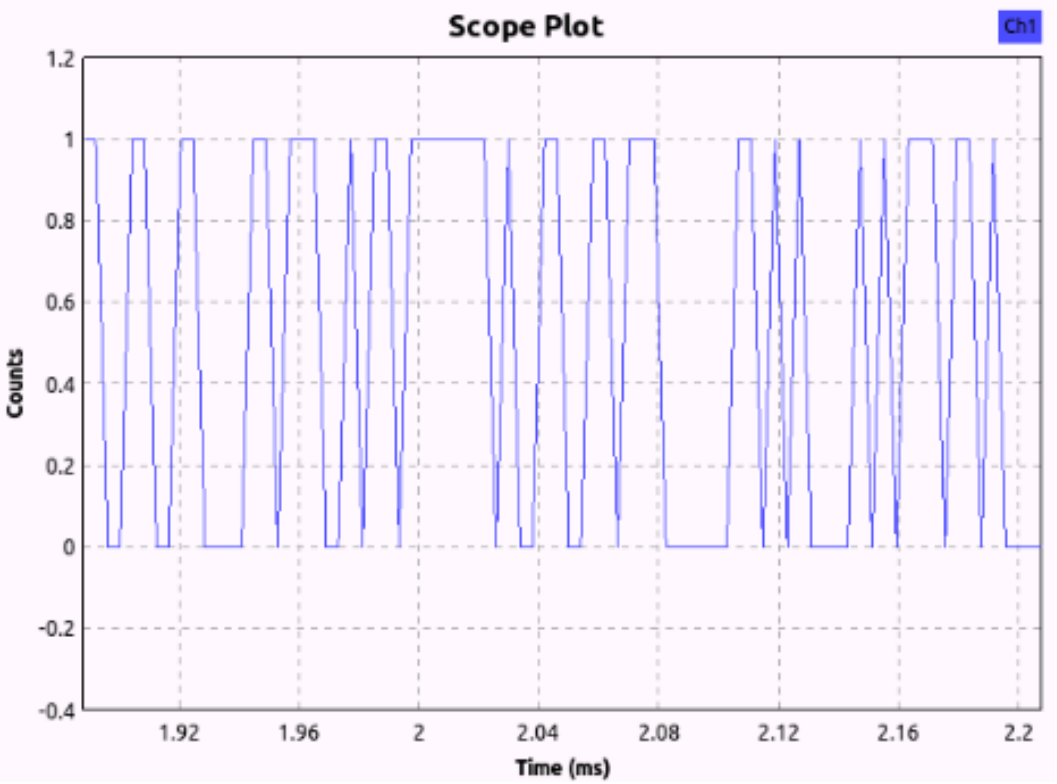

Figura B.12: Sinal no domínio do Tempo (GNU-Radio).

O gráfico que apresentamos a Figura B.13 mostra a ocupação espectral. Esta função no *GNU-Radio* tem o nome de *Waterfall Plot* e permite apreciar melhor o uso do espectro.

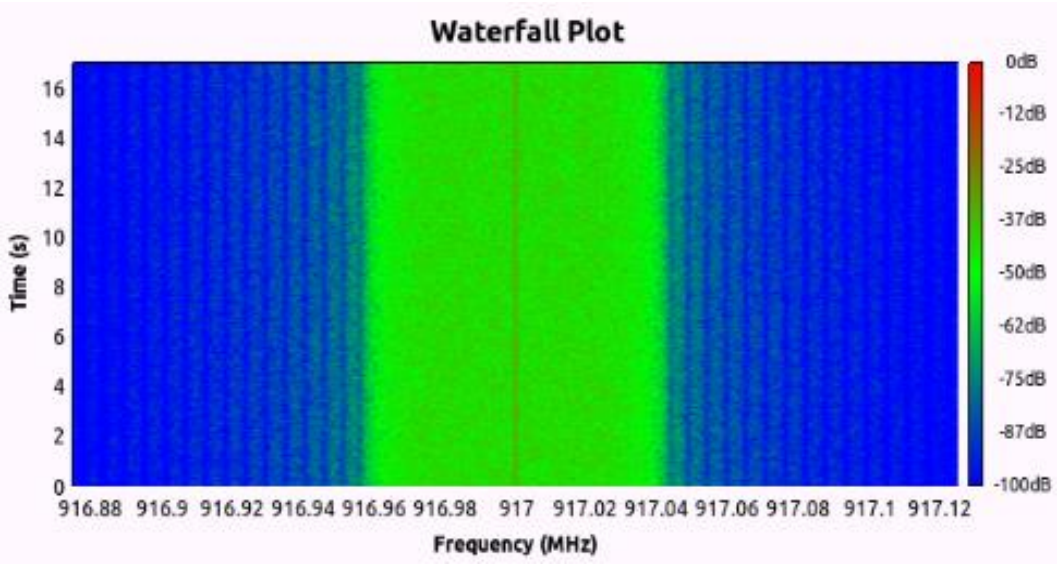

Figura B.13: Ocupação da largura de banda (GNU-Radio).

B.4.
Criação de um novo bloco tipo sync.

Esta parte do trabalho é muito importante porque dá origem a próximos desenvolvimentos em rádio cognitivo, fazendo uso do *GNU-Radio* com a criação de novos blocos que permitam o desenvolvimento do projeto. Partindo deste ponto, o *GNU-Radio* oferece a possibilidade de criar novos blocos, os quais tenham suas próprias características. O principal objetivo é cumprir os requisitos que o projeto assim o requeira.

Apresentamos os seguintes passos a seguir para a criação de blocos:

No terminal, fazemos a petição para criar um novo bloco dentro do file *gr-inmetro.* Com o comando *gr_modtool add,* declaramos o bloco que queremos criar, entre eles temos: *sink, source, sync, decimator, interpolator, general, tagged_stream, hier, noblock*. Neste caso de estudo, mostraremos a criação do bloco *sync.* É preciso ingressar um nome que em nosso caso é *sync_prov.* A linguagem a ser utilizado é *Python*. A tela do terminal é mostrada na Figura B.14.

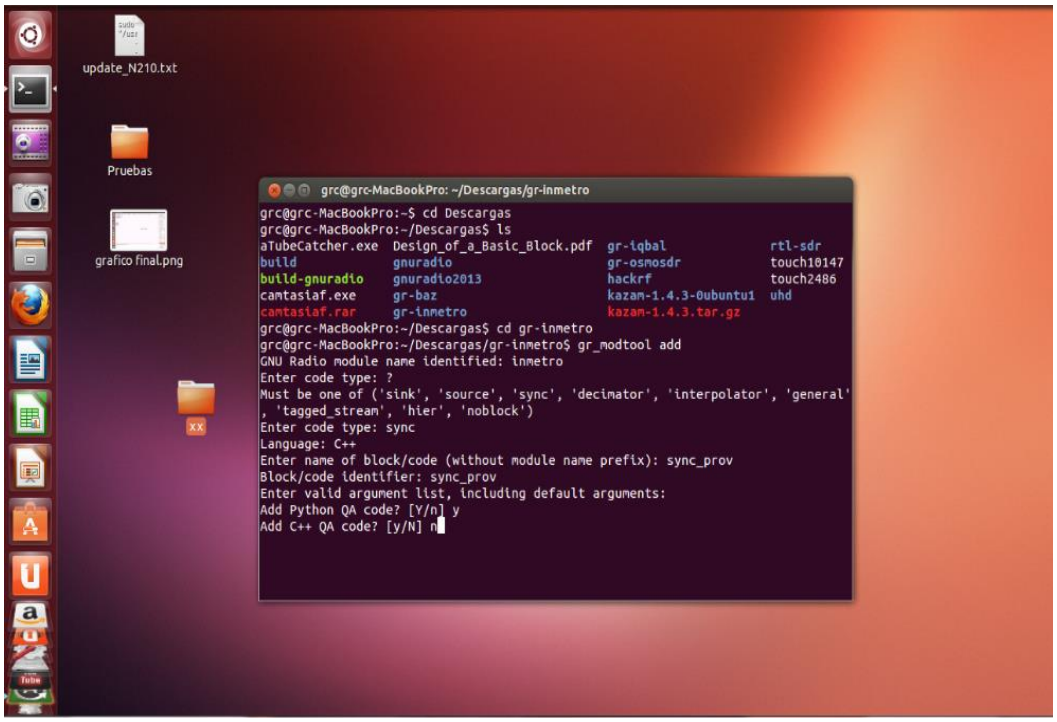

Figura B.14: Tela do terminal.

Uma vez ingressados os dados acima mencionados vemos uma resposta no terminal que nos permite continuar, como mostra a Figura B.15.

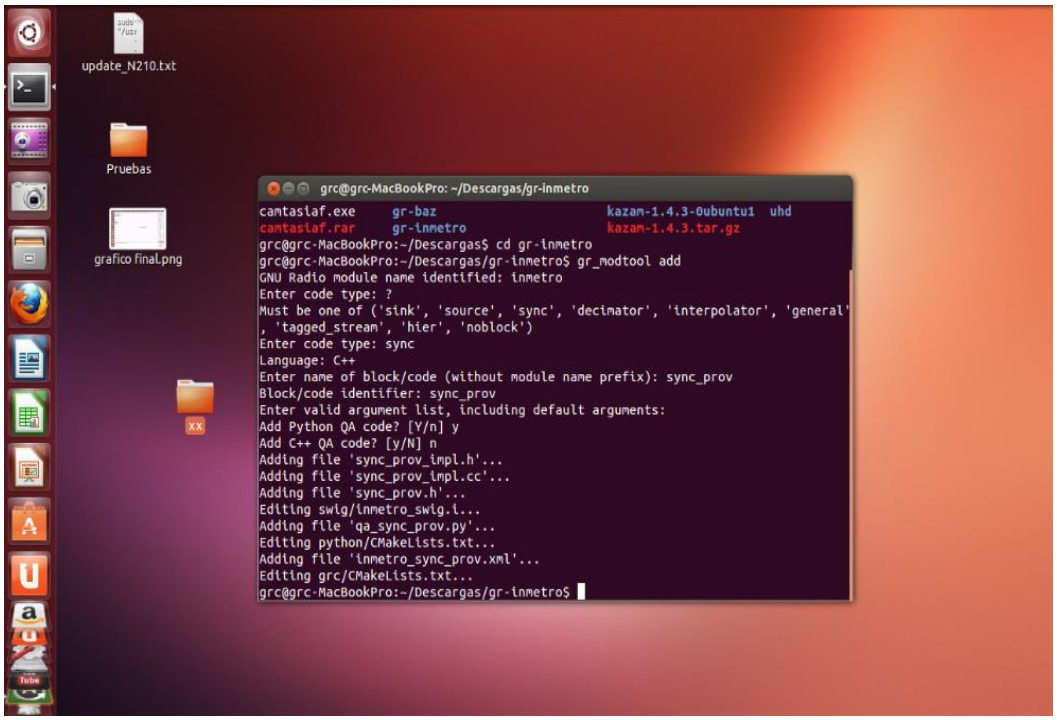

Figura B.15: Resposta do Terminal.

Uma vez que fazemos a petição, procuramos quatro arquivos que foram criados. Estes têm uma semelhança com o nome que declaramos na criação (*sync_prov*), sendo eles:

- sync_prov.h
- sync_prov _impl.h
- sync_prov 1_impl.cc
- inmetro_ sync_prov.xml

O primeiro destes *sync_prov.h* se encontra na Direção: *Descargas\gr-inmetro\include\inmetro,* conforme á tela mostrada na Figura B.16.

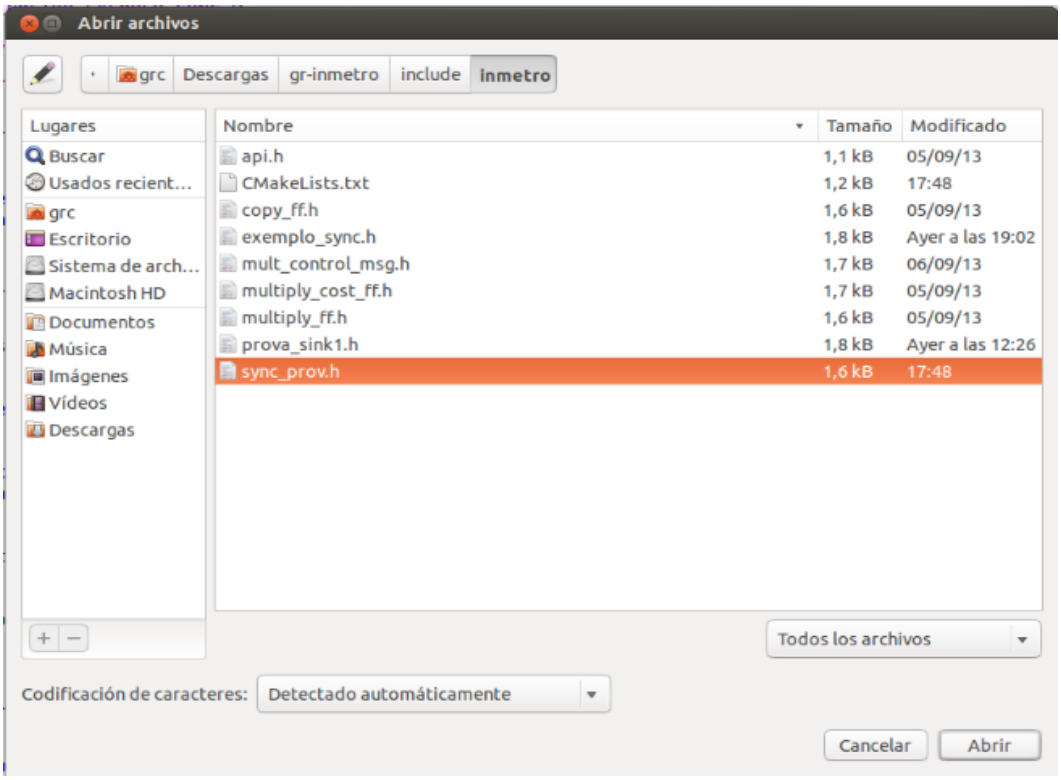

Figura B.16: Lugar no diretório onde se encontra o file sync_prov.h (Descargas\gr-inmetro\include\inmetro).

Dentro do arquivo, temos que aumentar os comandos mostrados na Figura B.17

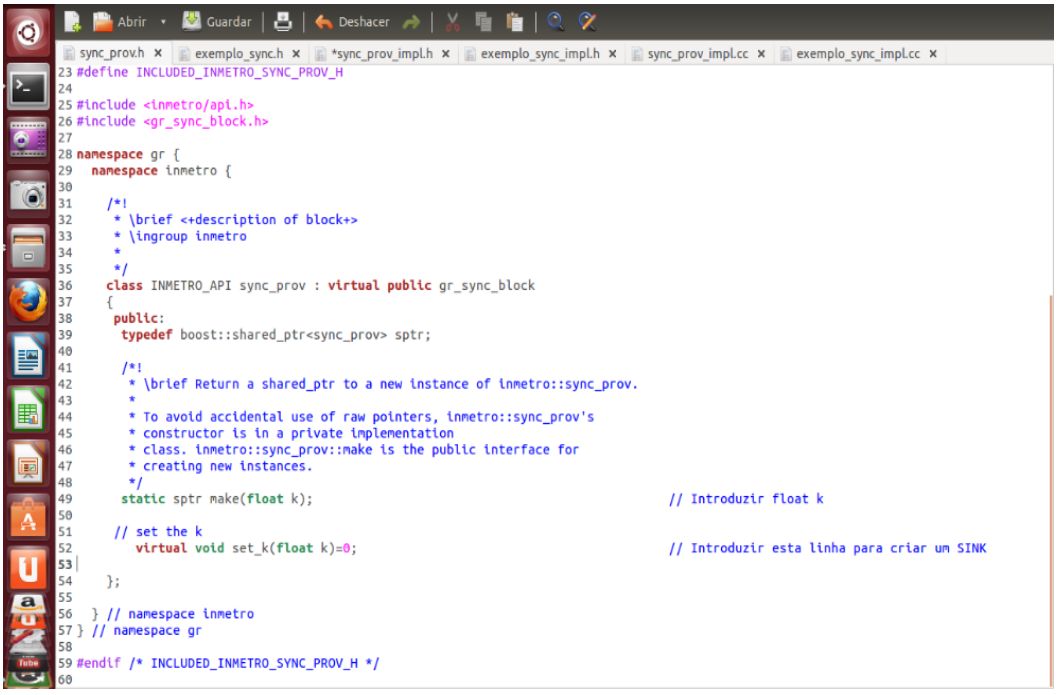

Figura B.17: Dentro do file sync_prov.h.

O segundo arquivo a ser procurado é *sync_prov _impl.h* este se encontra na direção: *Descargas\gr-inmetro\lib*, conforme mostra a Figura B.18.

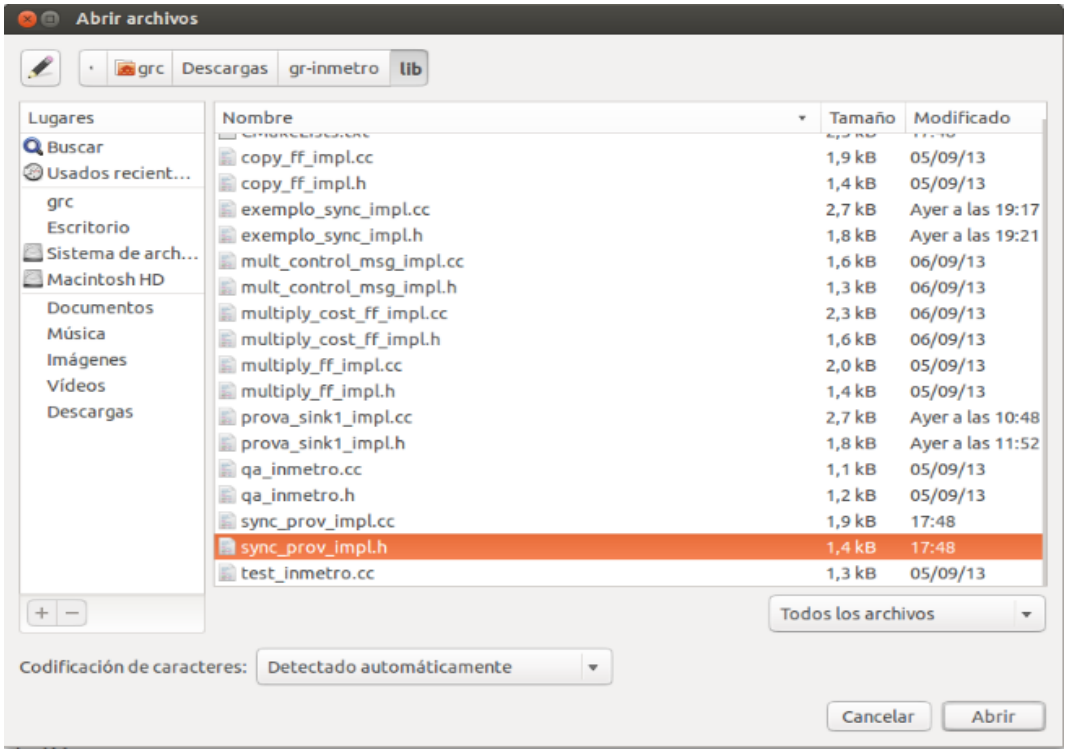

Figura B.18: Lugar no diretório onde se encontra o file sync_prov_impl.h (Descargas\gr-inmetro\lib).

Dentro do arquivo, incrementamos os seguintes comandos:

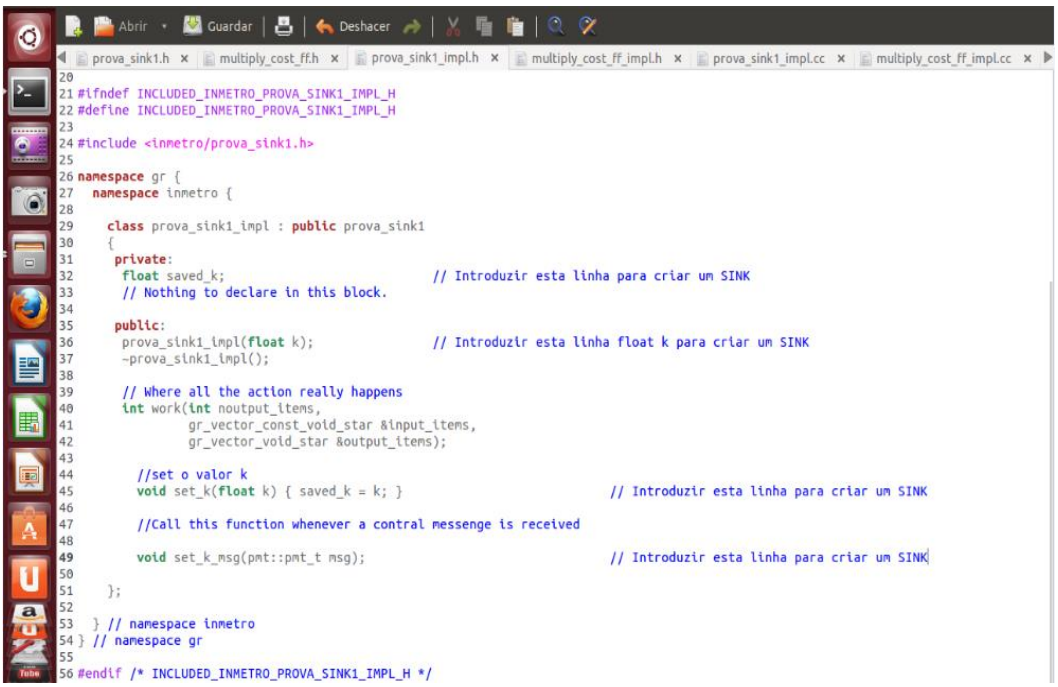

Figura B.19: Dentro do file sync_prov _impl.h.

O terceiro arquivo a ser procurado é *sync_prov 1_impl.cc* e este se encontra na direção: *Descargas\gr-inmetro\lib*. Como mostra a Figura B.20.

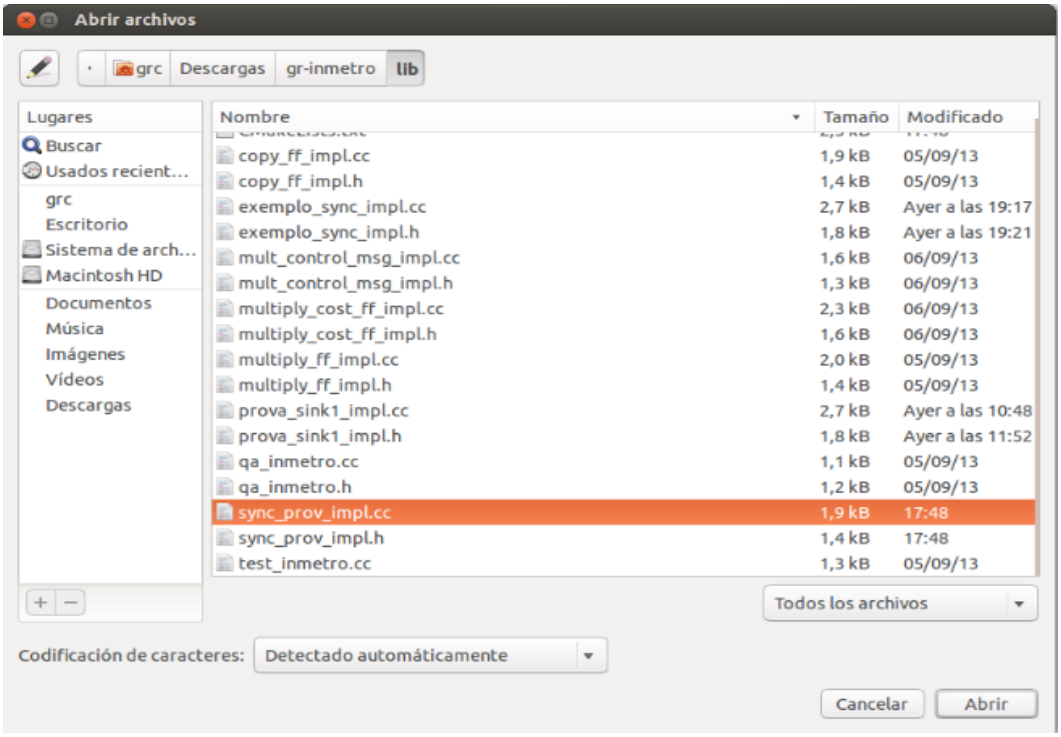

Figura B.20: Lugar no diretório onde se encontra o file sync_prov 1_impl.cc (Descargas\gr-inmetro\lib).

Dentro do arquivo temos que incrementar os comandos mostrados na Figura B.21 e na Figura B.22.

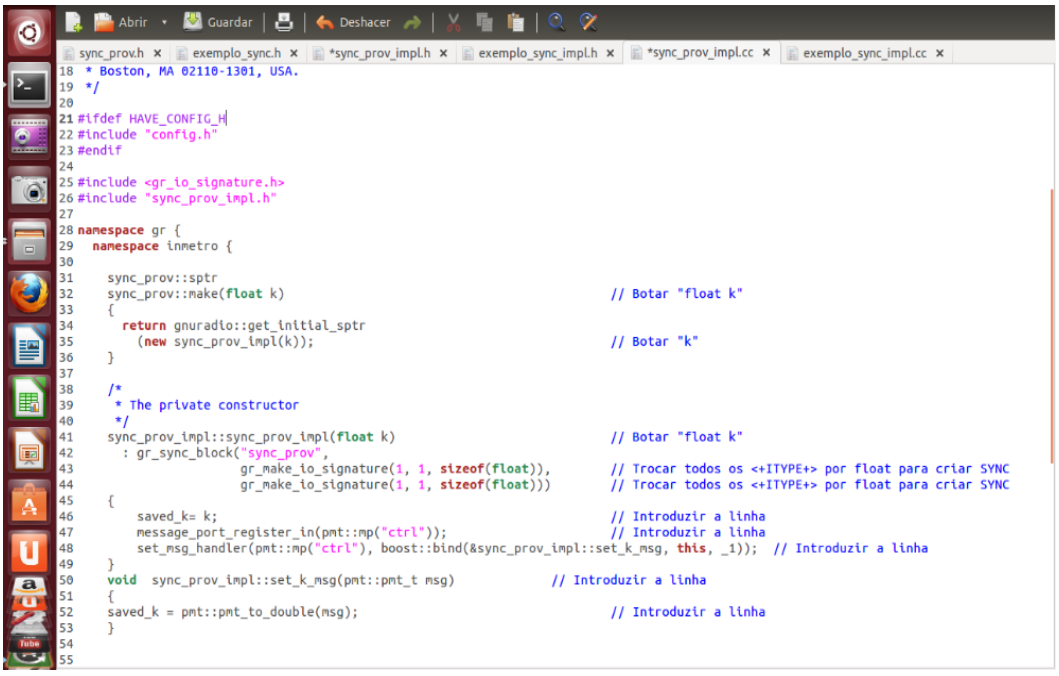

Figura B.21: Dentro do file sync_prov 1_impl.cc.

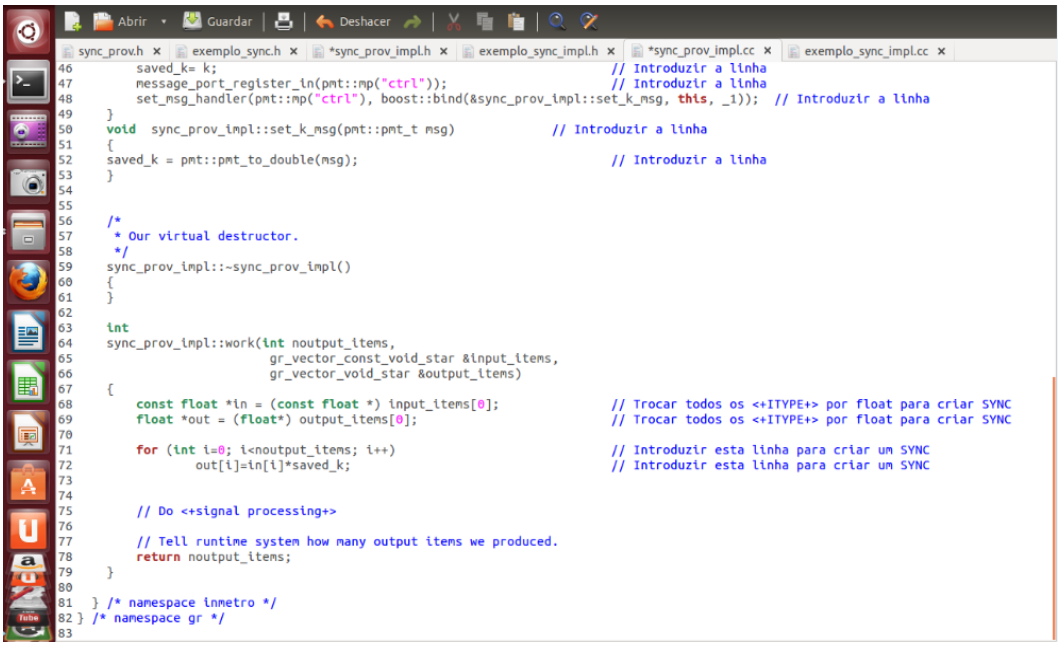

Figura B.22: Dentro do file sync_prov 1_impl.cc.

O quarto arquivo a ser procurado e modificado é *inmetro_ sync_prov.xml* este se encontra na direção: *Descargas\gr-inmetro\grc.*Como se ve na tela da Figura B.23.

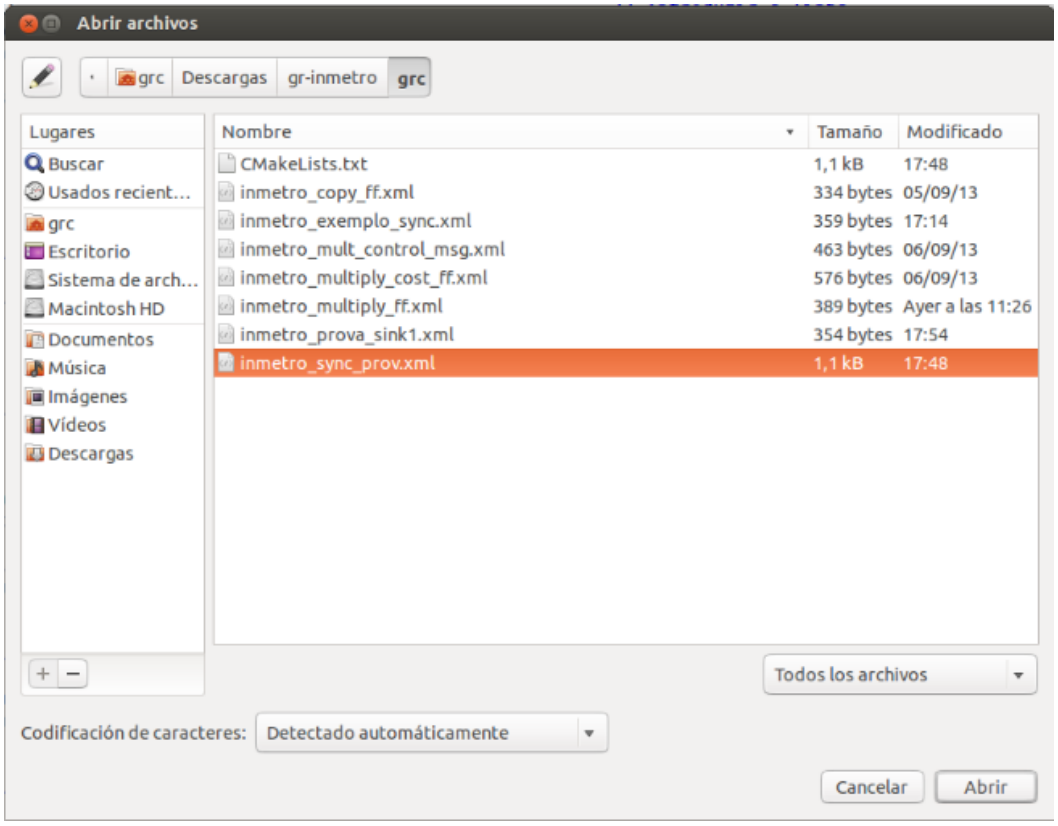

Figura B.23: Lugar no diretório onde se encontra o file inmetro_ sync_prov.xml (Descargas\gr-inmetro\grc).

Dentro do arquivo temos que incrementar e apagar alguns comandos, conforme mostrado na Figura B.24.

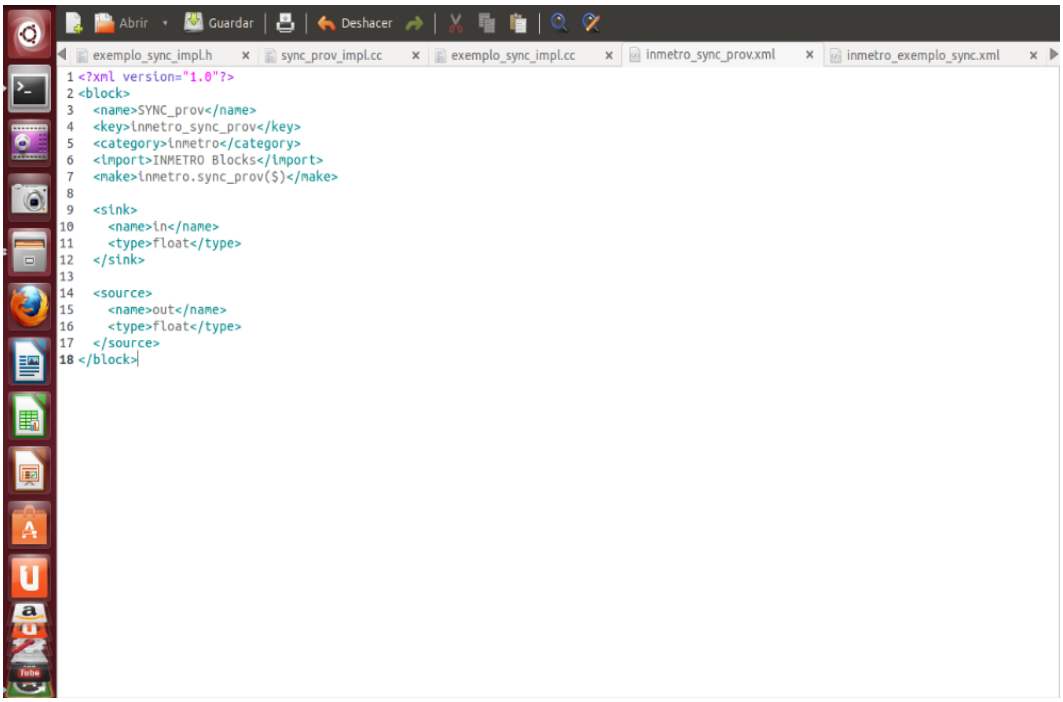

Figura B.24: Dentro do file inmetro_ sync_prov.xml.

Finalmente, depois de fazer todas estas mudanças, guardamos os quatro arquivo e rodamos o programa no terminal no arquivo *build* com a seguinte sequencia de comandos:

- grc@grc-MacBookPro:~/Descargas/gr-inmetro/build$ make clean
- grc@grc-MacBookPro:~/Descargas/gr-inmetro/build$ cmake ..
- grc@grc-MacBookPro:~/Descargas/gr-inmetro/build$ make

Conferimos que não tenha erro com o comando *make*, conforme se vê na tela da Figura B.25.

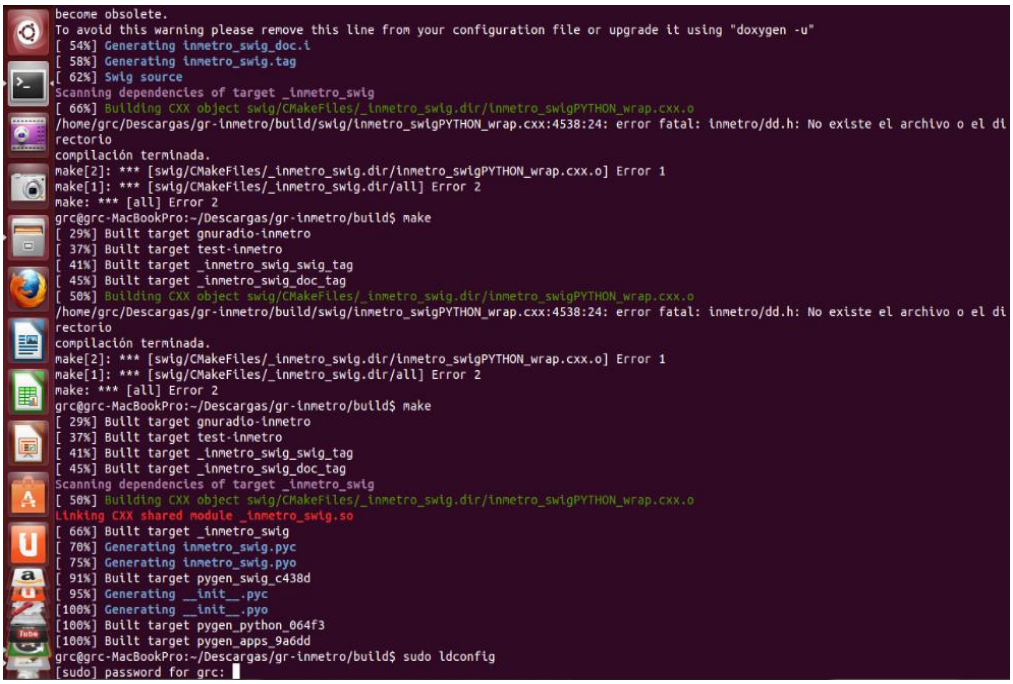

Figura B.25: Terminal, comando make.

Finalmente para guardar os dados obtidos e obter o bloco no *GNU-Radio Companion* usamos.

- grc@grc-MacBookPro:~/Descargas/gr-inmetro/build$ sudo ldconfig
- grc@grc-MacBookPro:~/Descargas/gr-inmetro/build$ sudo make install

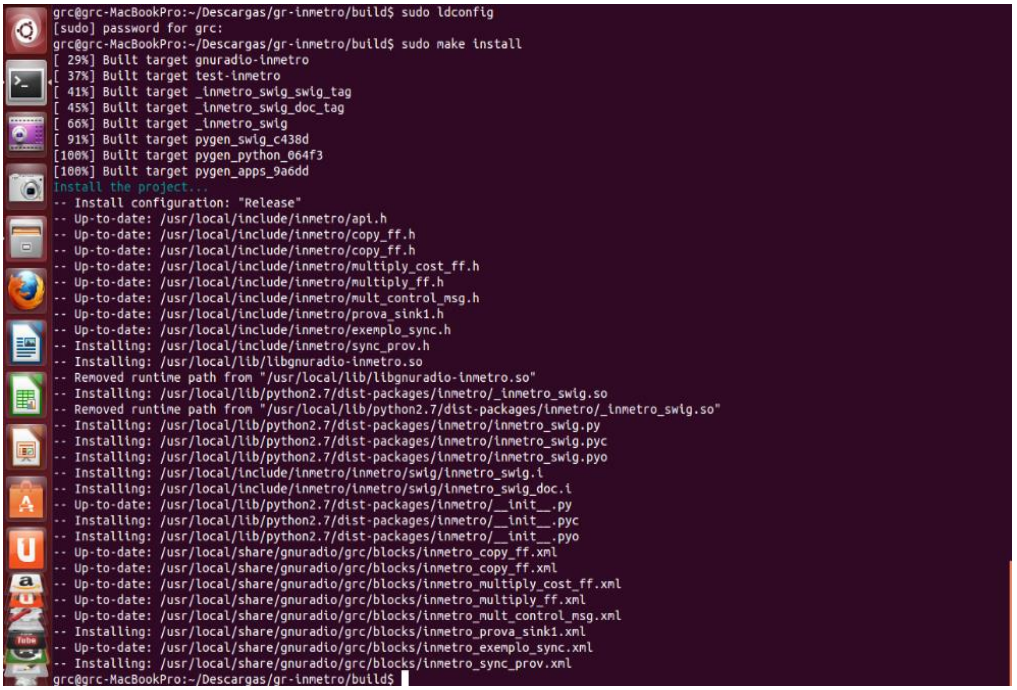

Figura B.26: Terminal comando.

Uma vez que tudo foi instalado corretamente ingressamos no *GNU-Radio Companion* para conferir a criação do bloco, como se vê na tela da Figura B.27.

- grc@grc-MacBookPro:~/Descargas/gr-inmetro/build$gnuradio-companion

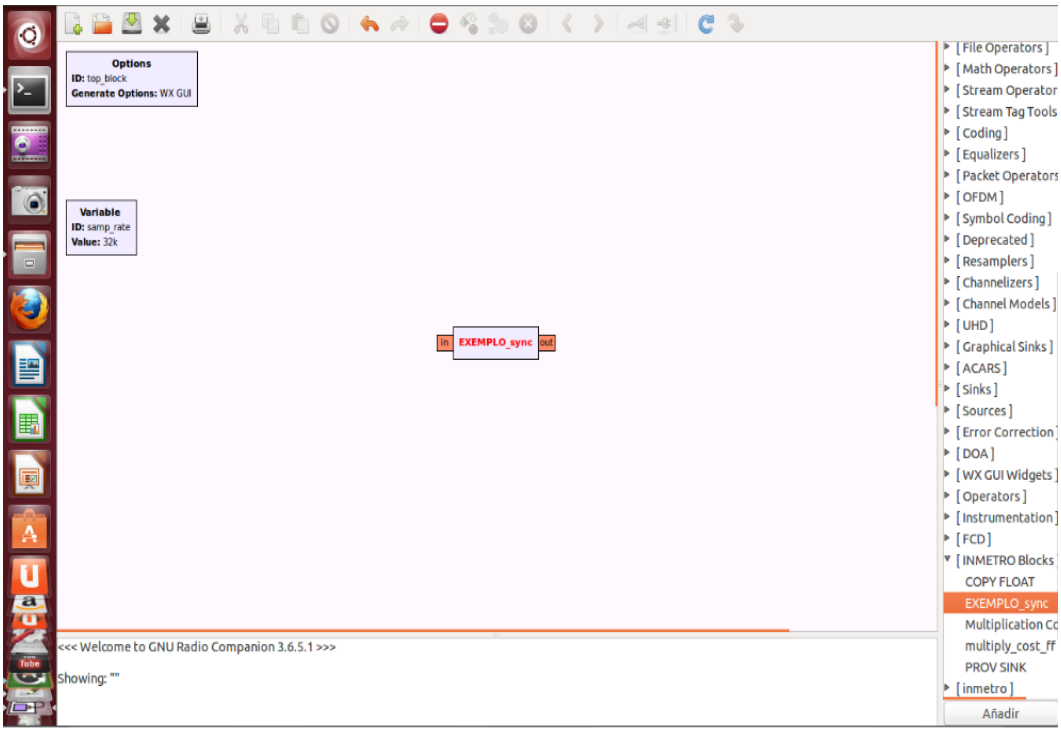

Figura B.27: GNU-Radio-Companion.

O bloco criado se encontra dentro do grupo INMETRO *Blocks* com o nome EXEMPLO_sync, assim como o definimos na Figura B.24.

C
Tempo de execução dos algoritmos.

Para avaliar os algoritmos pelo tempo de rodagem do programa, tem-se que realizar uma programação adequada à exigência do problema. Isso significa que é possível realizar e obter os mesmos resultados com um nível de programação maior, que possibilite uma resposta mais rápida do algoritmo. Por essa razão este resultado do trabalho se encontra no apêndice. A tabela C.1 mostra tempos de duração do processamento de cada técnica de sensoriamento de espectro.

		Minutos
1,75 [MHz]	**DE**	14 Minutos:10 segundos
	CAV	24 Minutos:37 segundos
	SCS	25 Minutos:52 segundos
3,5 [MHz]	**DE**	10 Minutos:22 segundos
	CAV	20 Minutos:32 segundos
	SCS	21 Minutos:52 segundos
7 [MHz]	**DE**	15 Minutos:33 segundos
	CAV	23 Minutos:12 segundos
	SCS	26 Minutos:09 segundos

Tabela C.1: Tempo de Rodagem dos programas ED, CAV, SCS.

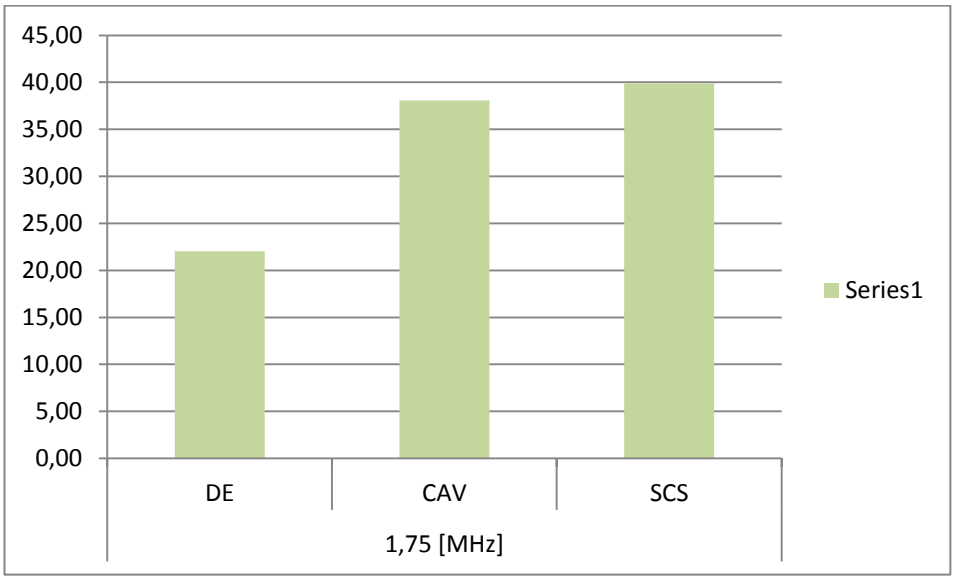

Figura C.1: Porcentagem dos tempos de processamento para a banda de 1.75 [MHz].

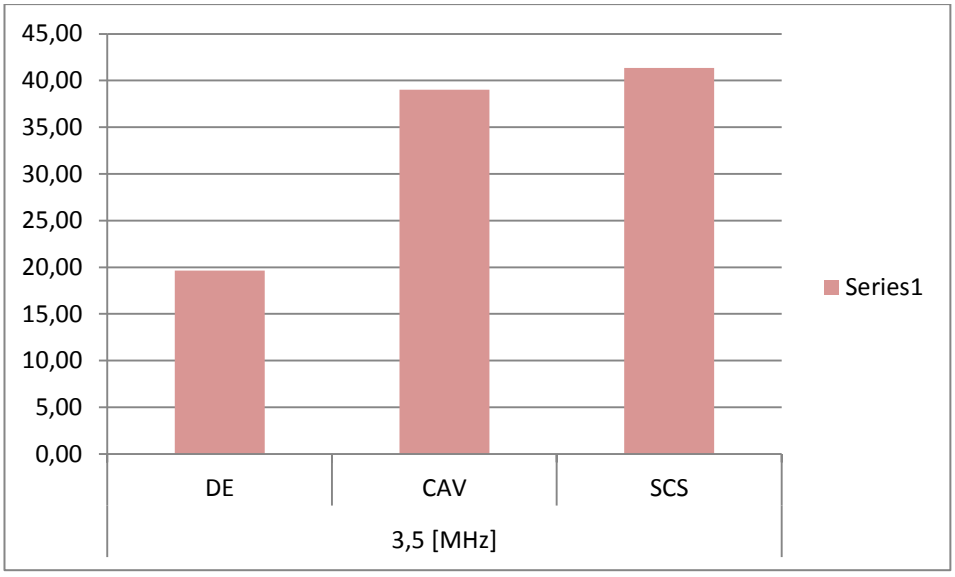

Figura C.2: Porcentagem dos tempos de processamento para a banda de 3.5 [MHz].

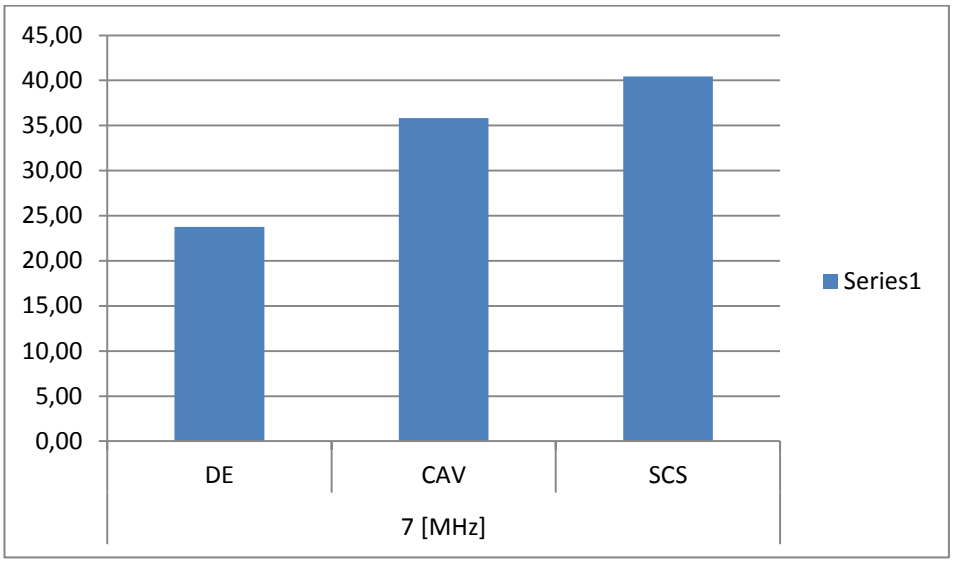

Figura C.3: Porcentagem dos tempos de processamento para a banda de 7 [MHz].

As figuras apresentadas mostram que o algoritmo mais rápido no processamento do programa é o algoritmo de detecção de energia (DE). Sua rapidez tem efeito graças a sua baixa complexidade computacional. Os algoritmos de covariância respondem quase do mesmo jeito, mas o algoritmo de sensoriamento de covariância espectral (SCS) tem um leve atraso em relação ao algoritmo baseado no valor absoluto de covariância (CAV).