



GUÍA PRÁCTICA

Preparación de los equipos de cómputo para GNU Radio.

Escrito por: Homero Ortega Boada.

Integrantes:

Lugo Vásquez Vanny Sofia- 2154709

Angarita Pertuz Jose David- 2142992

Cancino Rey William Andrés - 2151534

Grupo B1A.g2

El problema

Al momento de escribir esta guía estamos afrontando una situación especial: la necesidad de realizar prácticas por telepresencia. Teniendo en cuenta que en un laboratorio de comunicaciones se maneja tanto hardware como software muy avanzado y sobre Linux, el reto puede ser mayor que en otras asignaturas que afrontan una situación similar. En esta oportunidad deseamos comprobar si es posible aprovechar el cómputo en la Nube, usando Google Cloud, como apoyo para las prácticas de la asignatura.

Objetivo General

Responder a diversos interrogantes que surgen sobre la posibilidad de usar Google Cloud con máquinas virtuales con GNU Radio en prácticas de comunicaciones II. Entre esos interrogantes tenemos:

- En las máquinas que tenemos en Google Cloud tenemos instalada la nueva versión de GNU Radio, algo que nunca hemos explorado y que se basa en Python 3 a diferencia de las versiones que se tenían al día de hoy que se basaban en Python 2. Al mismo tiempo en la UIS tenemos librería que hemos ido desarrollando, la pregunta es ¿se instalarán fácilmente? ¿se requiere algo más? ¿funcionarán una vez instaladas?
- La nueva versión usa unas líneas curvas para unir unos bloques con otros. Eso puede ir en contra del método de diseño de sistemas de comunicaciones por capas que hemos impulsado desde la UIS. ¿qué se puede hacer al respecto? ¿se tiene la opción de volver a las líneas rectas?
- Para el trabajo en equipo usualmente 3 estudiantes trabajan sobre una misma máquina. Al menos así lo hacíamos en computadores físicos del Laboratorio de comunicaciones. Debemos comprobar qué tan cómodo resulta continuar igual pero por videoconferencia.



Pero también es importante analizar si el hecho de tener los computadores en la Nube dan pie a usar alguna estrategia más avanzada.

- ¿Es factible que cada estudiante cree su propia máquina virtual en Google Cloud para no depender de las máquinas que el laboratorio ofrece?

Aspectos teóricos

Investiga y discute con profesor y compañeros:

- Qué es el cómputo en la Nube
- Qué es Google Cloud

Trabajo previo. Preparativos y material de soporte

- [El Manual del laboratorista](#) para máquina virtual en Google Cloud
- [El Manual del Estudiante para Máquina Virtual en Google Cloud](#)
- [Manual para crear una máquina propia en Google Cloud con GNU radio](#)
- [El Manual de todos los Manuales](#)

Objetivos específicos

1. conocer el laboratorio de comunicaciones:
<https://sites.google.com/e3t.uis.edu.co/e3tsalas/laboratorios-e3t-todos>
2. entrar a la máquina en google cloud y hacer una prueba básica en GNU Radio
3. Para la nueva versión de GNU RAdio Investigar lo que implica:
 - a. instalar un OOT
 - b. Desarrollar un bloque para un OOT

Informe de resultados

Desarrollo del Objetivo 1. Presente a continuación los resultados del objetivo 1.



Figura 1. Sala digital del laboratorio de comunicaciones.

Se ingresa al laboratorio virtual de comunicaciones II, donde se observa que por medio de un ID se logra acceder de manera remota a un equipo que cuenta con gnruradio para realizar laboratorios que así lo requieran. También se visualiza que cada subgrupo de trabajo tiene una sala virtual en la cual puede compartir con los compañeros de laboratorio y el profesor.

Desarrollo del Objetivo 2. Presente a continuación los resultados del objetivo 2.

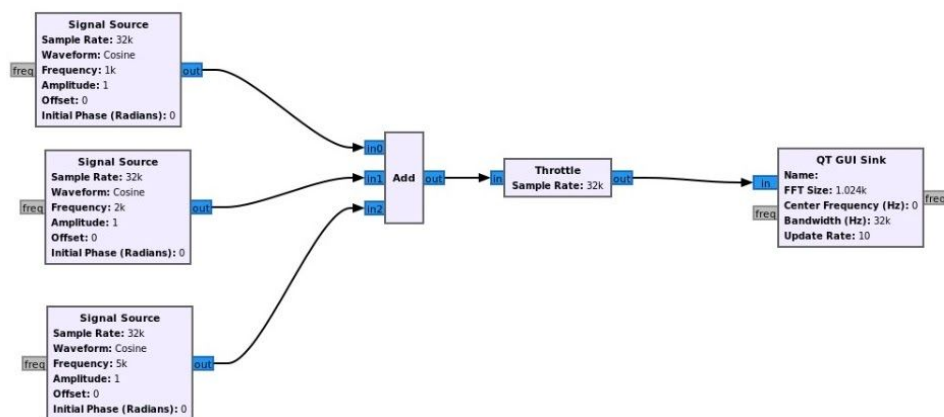


Figura 2. Flujograma para generar la señal.

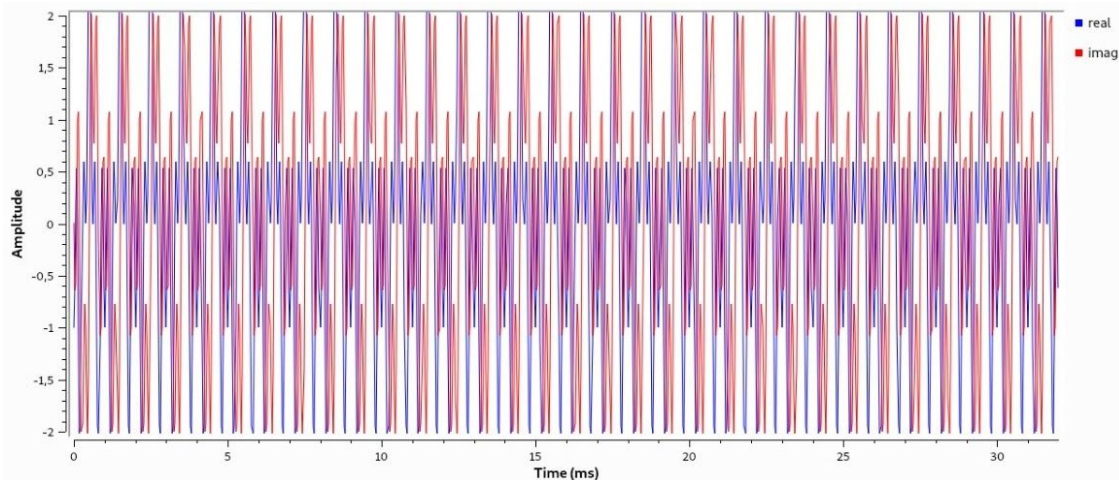


Figura 3. Señal en el dominio del tiempo.

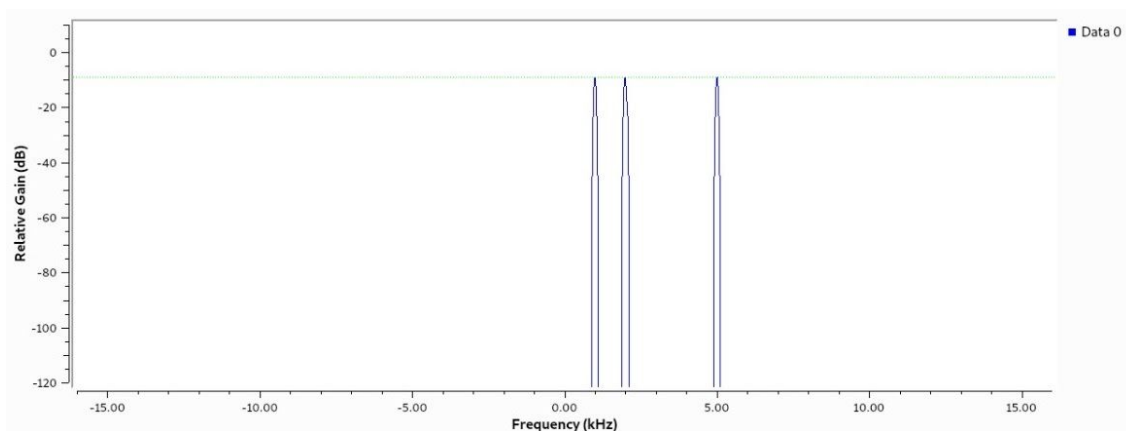


Figura 4. Señal en el dominio de la frecuencia.

Se genera una señal aleatoria en gnuradio, y se observa el comportamiento tanto en el dominio temporal como en el dominio frecuencial, con el fin de comprobar el funcionamiento.

Desarrollo del Objetivo 3. Presente a continuación los resultados del objetivo 3.

En este paso se procede a realizar la instalación de las librerías necesarias para el curso de comunicaciones II.

Como un prerrequisito de una de estas, se requiere la instalación de la librería “Eyediagram”. Para ello es necesario instalar algunas dependencias como “cython”, “matplotlib” y “scipy”, además de las bibliotecas tradicionales de python.

Una vez se cumple con estas dependencias, se instala la librería. Con el fin de comprobar su funcionamiento se ejecuta uno de los ejemplos que en ella se presenta, según se evidencia



en Figura 5.

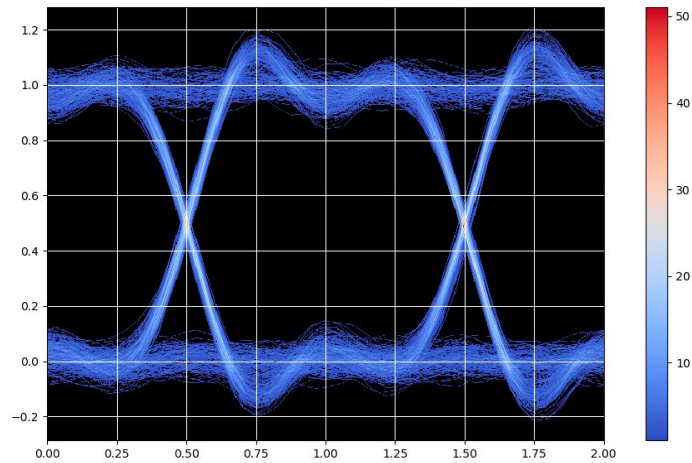


Figura 5. Ejemplo de diagrama de ojo.

Continuando con el proceso, como el módulo OOT depende de la anterior librería, finalmente se puede instalar. Una vez se instala, en el panel ubicado en la izquierda de la ventana de gnuradio aparece una categoría que se llama “E3TRadio”, la cual contiene los bloques respectivos de la librería, como se muestra en Figura 6. De igual forma, algunos de estos se presentan en la zona de trabajo, según se evidencia en dicha figura.



Figura 6. Algunos bloques del módulo OOT.

Finalmente, se instala la librería correspondiente a los bloques jerárquicos. Para esto se debe instalar bloque por bloque. Cuando se realiza esto, nuevamente en el panel izquierdo aparece dentro de la categoría “(no module specified)”, la subcategoría “comdiguis” que contiene estos bloques. En Figura 7., se presenta el ejemplo “fsk_demod” de esta librería. De igual forma en Figura 8., se muestra la simulación del ejemplo que se menciona anteriormente.



UNIVERSIDAD INDUSTRIAL DE SANTANDER
ESCUELA DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA Y DE TELECOMUNICACIONES

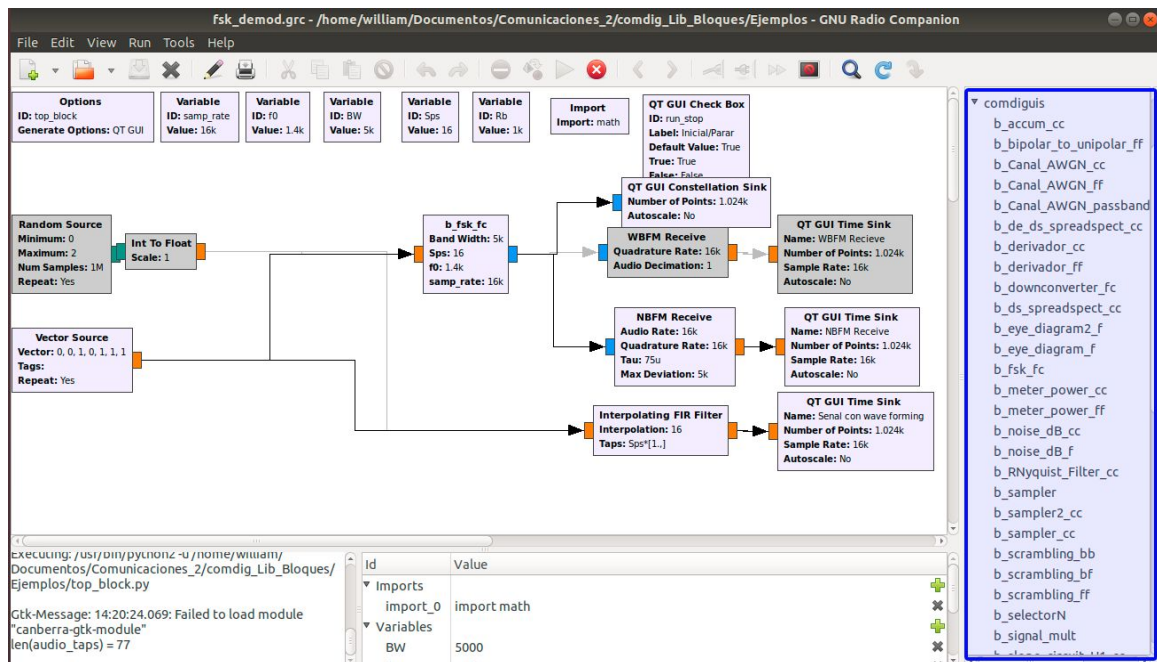


Figura 7. Algunos bloques jerárquicos y un ejemplo de esta librería.

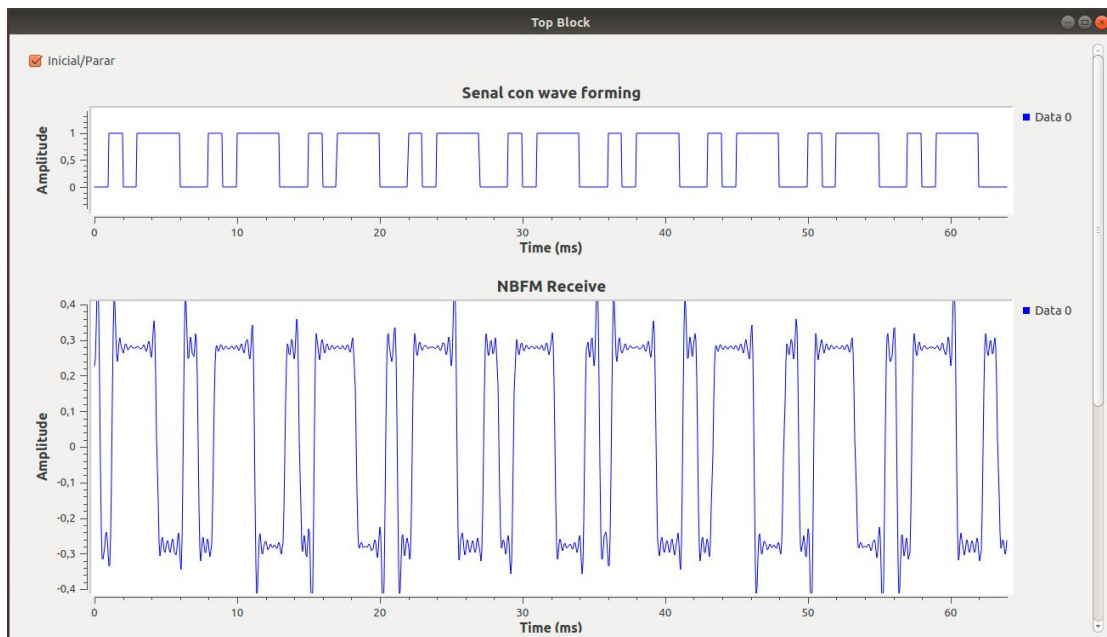


Figura 7. Simulación de ejemplo.

Un módulo OOT es un componente de gnuradio que no se encuentra dentro del árbol fuente de este. Es útil para extender el uso de gnuradio a funciones y bloques personalizados de un usuario.

Inicialmente, para crear un módulo se realiza mediante el script “gr_modtool”, el cual facilita este tipo de tareas. Una vez se crea este bajo el nombre “newoot”, se observa que hay una carpeta que se llama “gr-newoot” y está compuesta por un conjunto de subdirectorios. Los



pasos que se mencionan y la estructura de la carpeta del nuevo módulo se muestra en Figura 8.

```
william@william:~$ gr_modtool newmod newoot
Creating out-of-tree module in ./gr-newoot... Done.
Use 'gr_modtool add' to add a new block to this currently empty module.
william@william:~$ cd gr-newoot/
william@william:~/gr-newoot$ ls
apps  CMakeLists.txt  examples  include  MANIFEST.md  swig
cmake  docs            grc       lib       python
```

Figura 8. Creación del módulo “newoot” y estructura de su carpeta.

Seguidamente, se procede a crear un nuevo bloque mediante el comando que se muestra en Figura 9., esto genera varios archivos vacíos.

```
william@william:~/gr-newoot$ gr_modtool add -t sync -l cpp square2_ff
GNU Radio module name identified: newoot
Block/code identifier: square2_ff
Language: C++
Please specify the copyright holder: William
Enter valid argument list, including default arguments:
Add Python QA code? [Y/n] n
Add C++ QA code? [Y/n] n
Adding file 'lib/square2_ff_impl.h'...
Adding file 'lib/square2_ff_impl.cc'...
Adding file 'include/newoot/square2_ff.h'...
Editing swig/newoot_swig.i...
Adding file 'grc/newoot_square2_ff.xml'...
Editing grc/CMakeLists.txt...
```

Figura 9. Creación del bloque.

El objetivo del bloque es calcular el cuadrado de una entrada flotante individual. El bloque se escribe en C++. Parte del código necesario para dicho cálculo se evidencia en Figura 10. Sin embargo, no basta con este paso, ya que con los pasos que se realizan el bloque no está disponible en GRC, por lo tanto, es necesario crear y editar un archivo XML.

```
54  int
55  square2_ff_impl::work(int noutput_items,
56      gr_vector_const_void_star &input_items,
57      gr_vector_void_star &output_items)
58  {
59      const float *in = (const float *) input_items[0];
60      float *out = (float *) output_items[0];
61
62      for(int i = 0; i < noutput_items; i++) {
63          out[i] = in[i] * in[i];
64      }
65
66      // Do <+signal processing+>
67
68      // Tell runtime system how many output items we produced.
69      return noutput_items;
70  }
71
72  } /* namespace newoot */
73 } /* namespace gr */
74
```

Figura 10. Código del bloque para el cálculo del cuadrado de una entrada.

Para la instalación del módulo basta con ubicarse en la carpeta principal (“newoot”), y seguir los siguiente comandos.

- mkdir build
- cd build/



UNIVERSIDAD INDUSTRIAL DE SANTANDER
ESCUELA DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA Y DE TELECOMUNICACIONES

- cmake ../
- sudo make install
- sudo ldconfig

Con estos pasos se puede visualizar en el panel izquierdo el módulo “newoot”, y el respectivo bloque “Square2_ff” para calcular el cuadrado de una entrada. Lo que se menciona anteriormente, al igual que el flujograma para verificar el funcionamiento del bloque se muestra en Figura 11. De igual forma, la respectiva simulación se muestra en la Figura 12.

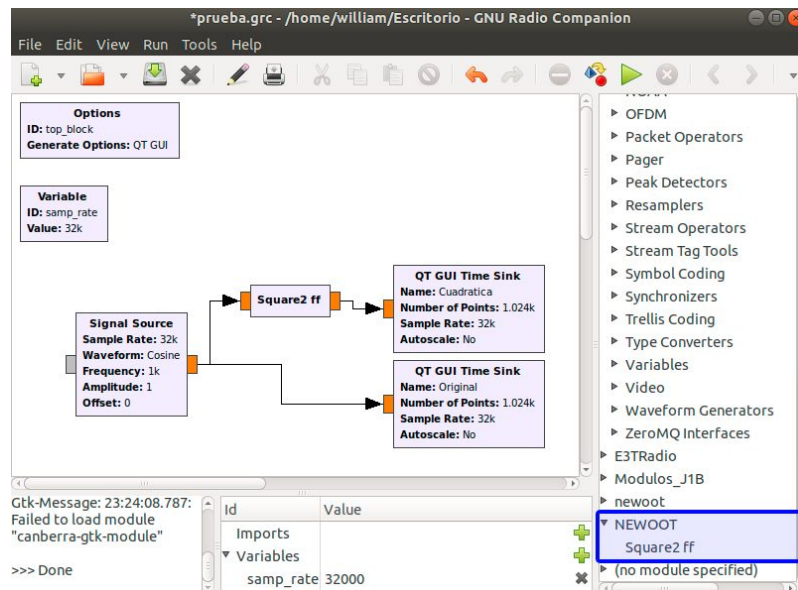


Figura 11. Flujograma y ubicación del bloque en el panel de módulos.

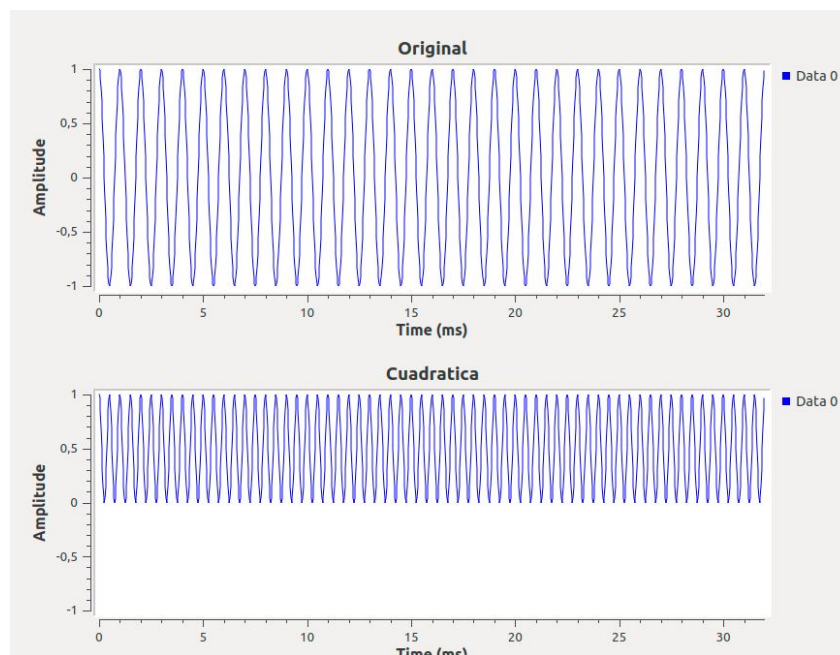


Figura 12. Simulación para comprobar el funcionamiento del bloque.



UNIVERSIDAD INDUSTRIAL DE SANTANDER
ESCUELA DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA Y DE TELECOMUNICACIONES

El proceso completo y los detalles para la creación e instalación de módulos OOT se encuentra en [3].

REFERENCIAS

- [1] Librerías OOT : https://github.com/hortegab/comdig_Lib_OOT_E3TRadio
- [2] Librerías bloques jerárquicos: https://github.com/hortegab/comdig_Lib_Bloques
- [3] OutOfTreeModules: <https://wiki.gnuradio.org/index.php/OutOfTreeModules>