



COMUNICACIONES II – 27145

PROGRAMACIÓN EN RADIO DEFINIDA POR SOFTWARE (GNURADIO)

JUSTIFICACIÓN

A partir del desarrollo que se viene dando en las distintas tecnologías, los entornos para programación de dispositivos ya no solo tienen bloques que permiten interactuar y conocer las distintas plataformas, sino que también ofrecen a los usuarios poder ir más allá, y programar sus propios algoritmos, de esta manera, el estudiante aprende no solo a utilizar los bloques sino a programar funcionalidades nuevas para complemento de las actividades que ya puede realizar con GNU Radio.

Objetivo general

Desarrollar competencias que permitan la interacción del estudiante con la programación de funciones en GNU Radio.

Objetivos específicos

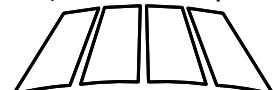
- Identificar los aspectos fundamentales en los sistemas de tiempo real y la radio definida por software.
- Generar funciones a partir de los bloques de implementación de código y evaluar los resultados con otros bloques.
- Utilizar los bloques implementados para producir una aplicación específica para señales reales.

Software a utilizar

- Linux (Ubuntu).
- Terminal de Linux (git).
- GNU Radio.

Procedimiento

1. Cree una rama para el desarrollo de la práctica desde el repositorio de su navegador a partir del siguiente tutorial (nombre de la rama: Practica_I a partir de la rama main o master):
<https://docs.github.com/es/pull-requests/collaborating-with-pull-requests/proposing-changes-to-your-work-with-pull-requests/creating-and-deleting-branches-within-your-repository>
2. El usuario que haya creado el repositorio debe compartir con sus compañeros de grupo para edición de los contenidos.
3. En la rama nueva (Practica_I), cree un directorio llamado de la misma manera y dentro de este, cree dos directorios: GNURadio e Informe, después de este paso, utilice los pasos de la práctica anterior (si no tiene llave SSH o token) y ejecute en la terminal “git pull” para actualizar los cambios de la rama (recuerde que para ejecutar este comando debe estar en la carpeta local donde tiene su repositorio de GitHub). Esta rama debe ser la que observe el profesor al revisar, esta funcionará como una vista de “preproducción” donde cada integrante subirá los cambios.
4. Ingrese a tal rama a partir del comando “git checkout Practica_I”.
5. Si por alguna razón no pudo crear los directorios dentro de GitHub (paso 3), hágalo entonces desde los comandos mkdir (que aprendió en la práctica anterior) en su estación de cómputo local, luego actualice en la nube (git add .., git commit -m “msg”, git push), recuerde que el





usuario que sube esta actualización debe ser el usuario de GitHub y la clave el token que generó en la práctica anterior. Verifique ahora en el navegador que los resultados nuevos se incluyan en la rama “Practica_I” que usted creó en GitHub.

6. Abra la aplicación GNURadio y guarde cambios del archivo creado en el directorio establecido para dicha práctica (numeral 3 o 5 si no pudo en 3).
7. Puede seguir el siguiente tutorial, para conocer un poco el bloque de python: https://wiki.gnuradio.org/index.php?title=Creating_Your_First_Block
8. Teniendo creado al menos el bloque de Python en la rama Practica_I (puede incluirle por ejemplo un bloque vector source y un QT GUI sink), cada uno de los integrantes del grupo debe hacer una rama adicional (a partir de la rama Practica_I) en donde cada uno de los integrantes realice los algoritmos, como no todos pueden trabajar en el mismo PC, se recomienda que, si el grupo tiene 3 personas, entonces nombren las ramas como PI_Nombre1, PI_Nombre2, PI_Nombre3 y cada uno queda con su rama.
9. Utilice el [libro guía](#) en la sección 1.2.0.1 e implemente en el bloque de Python el algoritmo sugerido.
10. Implemente los bloques de acumulador y de diferenciador.
11. Teniendo en cuenta lo aprendido, implemente un bloque para mostrar la estadística vista en clase y sugiera una aplicación.
12. Sugiera una aplicación de los conceptos vistos en clase y recrea una vista de los resultados.
13. Utilice el comando “git status” para verificar que hubo cambios de manera local en su PC.
14. Si en la ventana de comandos se encontraron cambios, se mostrará en color rojo los archivos que cambiaron (se agregaron o quitaron). Si está de acuerdo con esos cambios, utilice el comando “git add .”
15. Si no encuentra errores, ejecute el comando “git commit -m “mensaje” .”. El “mensaje” debe ir entre comillas y generalmente es un comentario que hace referencia a los archivos que subió (o eliminó) o por qué hizo los cambios. Si encuentra errores ejecute el paso 13.
16. Ejecute el comando git push (recuerde que necesita su usuario y la contraseña es la que le dieron del token, si lo creo por token) y verifique que los cambios hayan sido subidos a la nube (plataforma GitHub en el navegador).

Para entregar.

1. En los grupos de trabajo, cada estudiante debe realizar los pasos aquí ejecutados ya sea en clase o en el aula en otro momento que tenga disponible. Al compartir el repositorio es indispensable que se vea el **trabajo de cada uno** a partir de los cambios del **log del GitHub**.
2. Si es posible, utilice los formatos dados para presentación de informes en Comunicaciones I (formato Latex). Es necesario que mantenga la estructura que allí se les dio, pero la debe poderse ingresar primero a la rama Practica_I y luego se debe hacer un “**merge**” a las otras ramas (investigar como se hace el “merge”). Cada integrante debe aportar al desarrollo correcto del informe y debe quedar evidenciado en el repositorio.
3. El punto 12 se debe hacer en clase, pero cada grupo debería incluir algún tipo de ruido en el que utilicen la estadística como solución a tal problema de ruido.

Sugerencia: Los grupos de trabajo deben saber administrar las tareas para que ninguno de los integrantes quede sobrecargado. El link del repositorio se entrega en la plataforma Moodle y en la parte superior del informe.

