

# Proyecto1

fase i

Conquista de gnuradio a nivel de programación

Prácticas de programación en Python

“[darle un nombre aqui al sistema, por ejemplo: el Colibri2]”  
para 2 horas (1 semanas)

Autores: \_\_\_\_\_

Perteneciente al grupo: \_\_\_\_\_



**ESCUELA DE INGENIERIAS ELECTRICA ELECTRONICA Y TELECOMUNICACIONES- E3T**  
**UNIVERSIDAD INDUSTRIAL DE SANTANDER – UIS**

ORTEGA /Comunicaciones Digitales basadas en SRD / 2

[Aspectos a mejorar en la guia](#)

[Enlace a materiales de apoyo](#)

[El Problema:](#)

[El objetivo general es:](#)

[Preparativos](#)

[Apuntes de interés](#)

[Objetivos específicos](#)

[Informe de resultados](#)

[Desarrollo del Objetivo 1. Presente a continuación los resultados del objetivo 1.](#)

[Desarrollo del Objetivo 2. Presente a continuación los resultados del objetivo 2.](#)

[Desarrollo del Objetivo 3. Presente a continuación los resultados del objetivo 3.](#)

## **Aspectos a mejorar en la guia**

Los siguientes son apuntes del profesor para introducir mejoras a futuras prácticas:

- Por ahor ano hay apuntes

## **Enlace a materiales de apoyo**

- [Manual de manuales](#)
- [El libro de la asignatura](#)
- [Página del libro](#)

## **El Problema:**

Por ahora el problema a resolver consiste en que el estudiante no tiene las sufientes bases de programación por objetos en Python para pasar a realizar desarrollos usando GNU Radio y herramientas profesionales.

## **El objetivo general es:**

Retar al estudiante a construir su propio conocimiento para programar por objetos usando python y herramientas profesionales como Github y Visual Studio Code.

## Preparativos

- Baje una version actualizada del libro para concentrarse en el capítulo 5.2.12 donde está la teoría necesaria. Tenga en cuenta que: en [el libro de la asignatura](#). Observe que en los capítulos del libro ofrecen enlaces a código de software, a flujogramas y otros recursos que son parte del libro. Por ejemplo, observa que debajo de cada gráfica con flujogramas hay una nota que dice: “Flujograma usado”. Esos recursos usados en el libro están en la página del libro: <https://sites.google.com/saber.uis.edu.co/comdig/sw>

## Apuntes de interés

- Python es un lenguaje interpretado
- Python es un lenguaje indentado
- En Python los tipos de las variables se deduce de manera automática, por ejemplo:
  - Si se escribe `x=0.4` el lenguaje decide que `x` es una variable de tipo flotante
  - Si se escribe `x=4` el lenguaje decide que `x` es una variable de tipo entero
  - Si se escribe `x=[1,2,d,2,4,fuerte,9]` el lenguaje decide que `x` es una variable de tipo lista
  - para trabajar con vectores se debe usar una librería llamada numpy:
    - `import numpy as np` # para importar la libreria
    - `x=np.array([1,2,3,4,5])` # crea un vector
    - `x=np.linspace(2.0, 3.0, num=5)` # es otro ejemplo para crear un vector
    - ver más ejemplos de trabajo con vectores en: [Manual de manuales](#), sección “Python para desmemoriados”

## Objetivos específicos

1. Programación por Terminal de Ubuntu. Descubra la utilidad de programar por terminal de Ubuntu para realizar cálculos rápidos o para comprobar que algunas sentencias o métodos funcionan correctamente antes pasarlos a un programa más complejo. Para ello, abra un terminal de Ubuntu (consulte el atajo para abrir una terminal de Ubuntu). Compruebe que puede realizar los siguientes cálculos:
  - a. Programación tradicional: `x`, `y` son números escalares, por ejemplo `x=0.4`, `y=1.3`
    - i. `z=x+y`
    - ii. `h=cos(x)`
    - iii.  $g = e^{j2\pi t}$
  - b. Programación vectorial: `x`, `y` son números vectores
    - i. `z=x+y`

ii.  $h = \cos(x)$

iii.  $g = e^{j2\pi t}$

2. Programación sobre archivos. Con los resultados del punto 1, cree un archivo con extensión .py y compruebe su funcionamiento:
  - a. use “gedit” como editor
  - b. usar “Visual Studio Code” como editor
3. Uso de Github. Suscribase en [www.github.com](https://www.github.com) siguiendo las siguientes pautas:
  - a. la inscripción es personal, es decir, cada persona debe crear su cuenta
  - b. crear un repositorio por grupo. Es decir, solo una de las personas del grupo crea un repositorio y lo comparte con los demás. Nota: cuando se le pregunte, dele la opción de incluir README.
  - c. El nombre del repositorio coincide con el nombre del grupo
  - d. Desde terminal de Ubuntu use los comandos (Nota: Encuentre más detalles en [Manual de manuales](#), sección “Manual de Github” ):

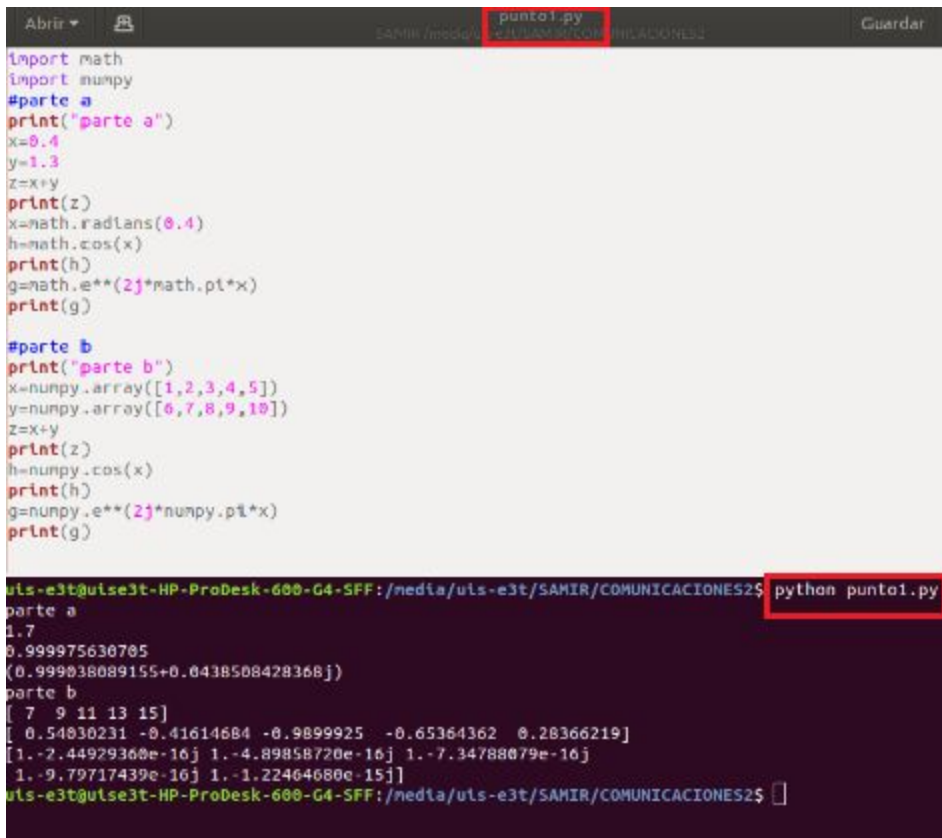
comando	explicación
git clone URL	para clonar su repositorio. Desde su cuenta copia la URL del repositorio; en terminal Ubuntu envía el comando; en tu computador aparecerá la carpeta del repositorio
git add .	para agregar nuevo contenido a una lista
git commit -m “comentario”	para pasar el nuevo contenido a base local
git push	para que el contenido suba al repositorio en la nube


## Informe de resultados

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

```
uis-e3t@uise3t-HP-ProDesk-600-G4-SFF:~$ python
Python 2.7.15+ (default, Jul  9 2019, 16:51:35)
[GCC 7.4.0] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> x=0.4
>>> y=1.3
>>> z=x+y
>>> print(z)
1.7
>>> import math
>>> x=math.radians(0.4)
>>> h=math.cos(x)
>>> print(h)
0.999975630705
>>> g=math.e**(2j*math.pi*x)
>>> print(g)
(0.999038089155+0.0438508428368j)
uis-e3t@uise3t-HP-ProDesk-600-G4-SFF:~$ python
Python 2.7.15+ (default, Jul  9 2019, 16:51:35)
[GCC 7.4.0] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import numpy
>>> import math
>>> x=numpy.array([1,2,3,4,5])
>>> y=numpy.array([6,7,8,9,10])
>>> z=x+y
>>> print(z)
[ 7  9 11 13 15]
>>> h=numpy.cos(x)
>>> print(h)
[ 0.54030231 -0.41614684 -0.9899925  -0.65364362  0.28366219]
>>> g=numpy.e**(2j*numpy.pi*x)
>>> g
array([1.-2.44929360e-16j, 1.-4.89858720e-16j, 1.-7.34788079e-16j,
       1.-9.79717439e-16j, 1.-1.22464680e-15j])
```

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



```
Abrir +  punto1.py Guardar
SAHIR /media/uis-e3t/SAHIR/COMUNICACIONES2

import math
import numpy
#parte a
print("parte a")
x=0.4
y=1.3
z=x+y
print(z)
x=math.radians(0.4)
h=math.cos(x)
print(h)
g=math.e**(2j*math.pi*x)
print(g)

#parte b
print("parte b")
x=numpy.array([1,2,3,4,5])
y=numpy.array([6,7,8,9,10])
z=x+y
print(z)
h=numpy.cos(x)
print(h)
g=numpy.e**(2j*numpy.pi*x)
print(g)

uls-e3t@ulse3t-HP-ProDesk-600-G4-SFF:/media/uis-e3t/SAHIR/COMUNICACIONES2$ python punto1.py
parte a
1.7
0.999975630705
(0.999938089155+0.0438508428368j)
parte b
[ 7  9 11 13 15]
[ 0.54030231 -0.41614684 -0.9899925  -0.65364362  0.28366219]
[ 1.-2.44929360e-16j  1.-4.89858720e-16j  1.-7.34788079e-16j
  1.-9.79717439e-16j  1.-1.22464680e-15j]
uls-e3t@ulse3t-HP-ProDesk-600-G4-SFF:/media/uis-e3t/SAHIR/COMUNICACIONES2$
```

```

Welcome
puntovisual.py x
home > vis-e3l > Documentos > puntovisual.py > ...
1 import math
2 import numpy
3 #parte a
4 print("parte a")
5 x=0.4
6 y=1.3
7 z=x+y
8 print(z)
9 x=math.radians(0.4)
10 h=math.cos(x)
11 print(h)
12 g=math.e**(2j*math.pi*x)
13 print(g)
14
15 #parte b
16 print("parte b")
17 x=numpy.array([1,2,3,4,5])
18 y=numpy.array([0,7,8,9,10])
19 z=x+y
20 print(z)
21 h=numpy.cos(x)
22 print(h)
23 g=numpy.e**(2j*numpy.pi*x)
24 print(g)

Documentos                               'In'$\303\241'$genes'
Escritorio                               Instaladores
vis-e3t@vis-e3t-HP-ProDesk-600-G4-SFF:~$ cd Documentos
vis-e3t@vis-e3t-HP-ProDesk-600-G4-SFF:~/Documentos$ ls
BAL.g5 puntovisual.py
vis-e3t@vis-e3t-HP-ProDesk-600-G4-SFF:~/Documentos$ python puntovisual.py
parte a
1.7
0.999975630705
(0.999030009155+0.0438508420368j)
parte b
[ 7.  9. 11. 13. 15.]
[ 0.54030231 -0.41614684 -0.98999925 -0.65364362  0.28366219]
[1.-7.44929360e-16j 1.-4.89858720e-16j 1.-7.34788079e-16j
 1.-9.79717439e-16j 1.-1.22464680e-15j]
vis-e3t@vis-e3t-HP-ProDesk-600-G4-SFF:~/Documentos$

```

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