

Proyecto 1

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)

Autor:	Alex Julian Mantilla Rios
Perteneciente al grupo:	B1A.G6

[Aspectos a mejorar en la guía](#)

[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 guía

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

- Por ahora no 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 suficientes 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 flotante
 - 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 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
<code>git clone URL</code>	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

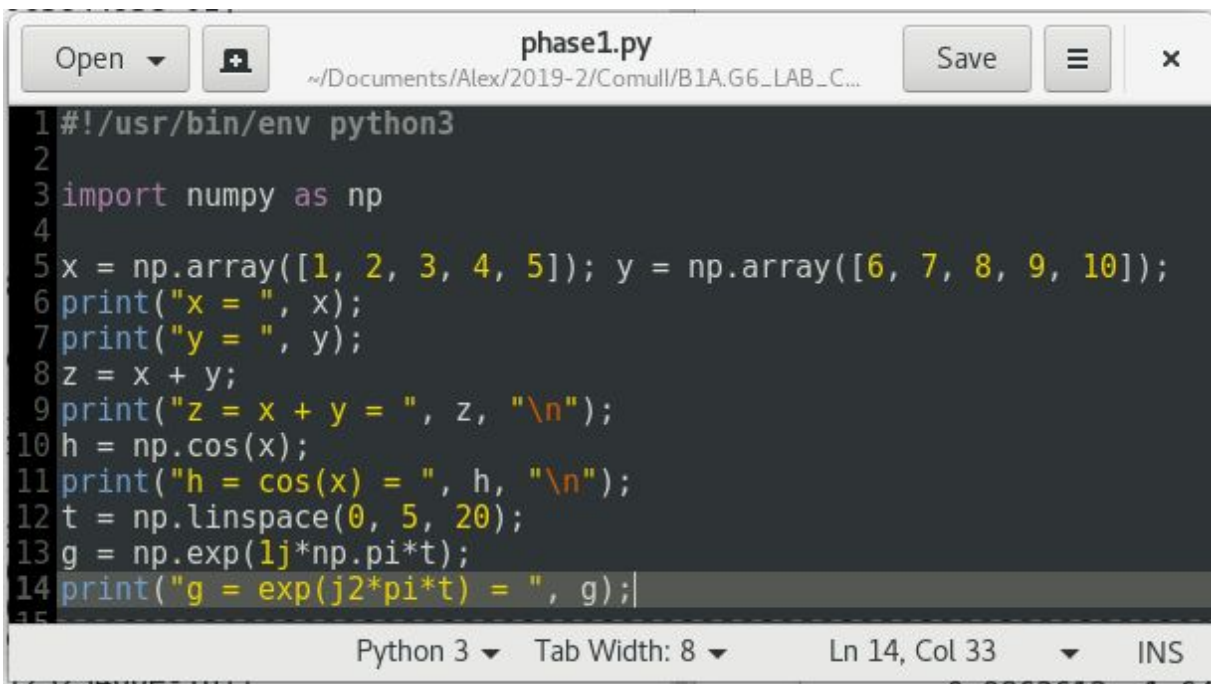
Desarrollo del Objetivo 1. Presente a continuación los resultados del objetivo 1.

```
onchip@onchipd-112550: ~
File Edit View Search Terminal Help
onchip@onchipd-112550:~$ python3
Python 3.5.3 (default, Sep 27 2018, 17:25:39)
[GCC 6.3.0 20170516] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> x = 0.4; y = 1.3;
>>> z = x + y; z
1.7000000000000002
>>> import math
>>> h = math.cos(x); h
0.9210609940028851
>>> import cmath
>>> g = cmath.exp(1j*2*cmath.pi); g
(1-2.4492935982947064e-16j)
>>>
```

```
onchip@onchipd-112550: ~
File Edit View Search Terminal Help
onchip@onchipd-112550:~$ python3
Python 3.5.3 (default, Sep 27 2018, 17:25:39)
[GCC 6.3.0 20170516] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import numpy as np
>>> x = np.array([1, 2, 3, 4, 5]); y = np.array([6, 7, 8, 9, 10]);
>>> z = x + y; z
array([ 7,  9, 11, 13, 15])
>>> h = np.cos(x); h
array([ 0.54030231, -0.41614684, -0.9899925 , -0.65364362,  0.28366219])
>>> t = np.linspace(0, 5, 20);
>>> g = np.exp(1j*np.pi*t); g
array([ 1.          +0.00000000e+00j,  0.67728157+7.35723911e-01j,
        -0.08257935+9.96584493e-01j, -0.78914051+6.14212713e-01j,
        -0.9863613 -1.64594590e-01j, -0.54694816-8.37166478e-01j,
         0.24548549-9.69400266e-01j,  0.87947375-4.75947393e-01j,
         0.94581724+3.24699469e-01j,  0.40169542+9.15773327e-01j,
        -0.40169542+9.15773327e-01j, -0.94581724+3.24699469e-01j,
        -0.87947375-4.75947393e-01j, -0.24548549-9.69400266e-01j,
         0.54694816-8.37166478e-01j,  0.9863613 -1.64594590e-01j,
         0.78914051+6.14212713e-01j,  0.08257935+9.96584493e-01j,
        -0.67728157+7.35723911e-01j, -1.          +6.12323400e-16j])
>>>
```

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

gedit:

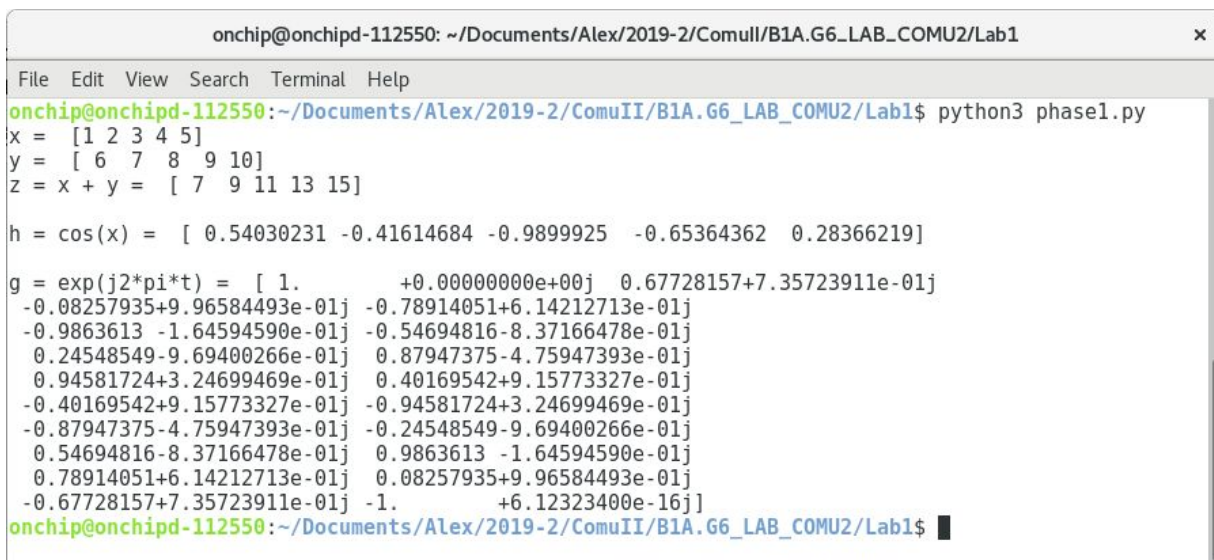


```

1#!/usr/bin/env python3
2
3import numpy as np
4
5x = np.array([1, 2, 3, 4, 5]); y = np.array([6, 7, 8, 9, 10]);
6print("x = ", x);
7print("y = ", y);
8z = x + y;
9print("z = x + y = ", z, "\n");
10h = np.cos(x);
11print("h = cos(x) = ", h, "\n");
12t = np.linspace(0, 5, 20);
13g = np.exp(1j*np.pi*t);
14print("g = exp(j2*pi*t) = ", g);
15

```

output:



```

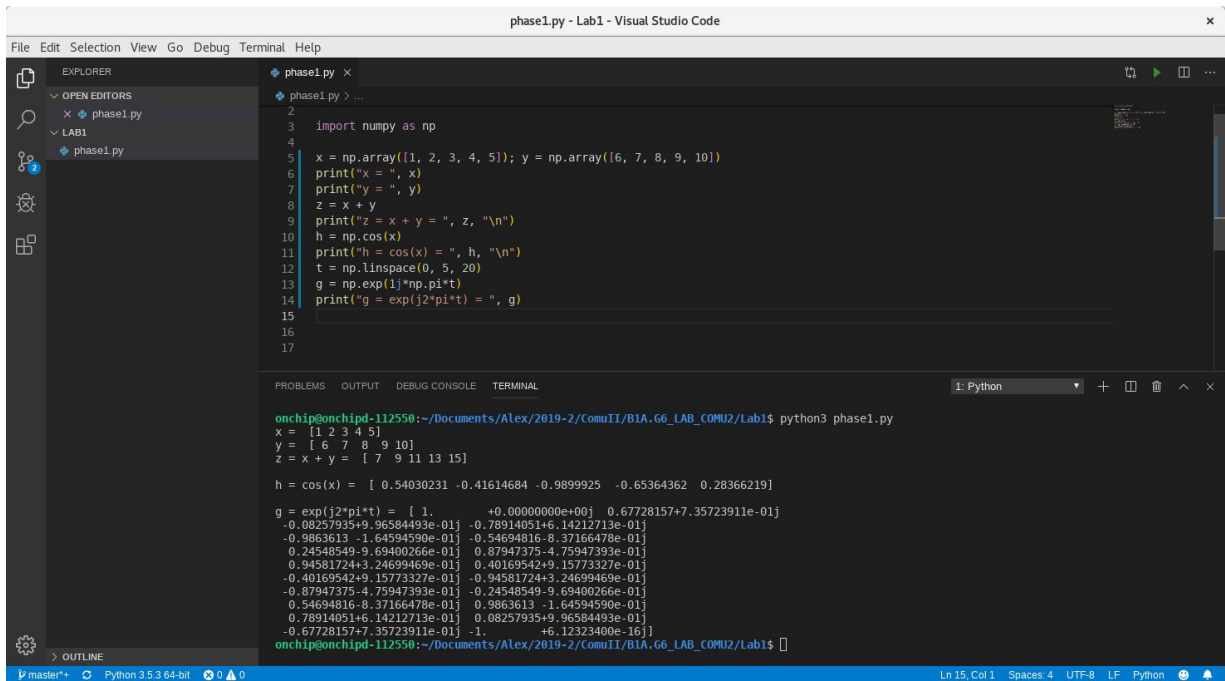
onchip@onchipd-112550: ~/Documents/Alex/2019-2/ComuII/B1A.G6_LAB_COMU2/Lab1
File Edit View Search Terminal Help
onchip@onchipd-112550:~/Documents/Alex/2019-2/ComuII/B1A.G6_LAB_COMU2/Lab1$ python3 phase1.py
x = [1 2 3 4 5]
y = [ 6  7  8  9 10]
z = x + y = [ 7  9 11 13 15]

h = cos(x) = [ 0.54030231 -0.41614684 -0.9899925  -0.65364362  0.28366219]

g = exp(j2*pi*t) = [ 1.          +0.00000000e+00j  0.67728157+7.35723911e-01j
-0.08257935+9.96584493e-01j -0.78914051+6.14212713e-01j
-0.9863613  -1.64594590e-01j -0.54694816-8.37166478e-01j
 0.24548549-9.69400266e-01j  0.87947375-4.75947393e-01j
 0.94581724+3.24699469e-01j  0.40169542+9.15773327e-01j
-0.40169542+9.15773327e-01j -0.94581724+3.24699469e-01j
-0.87947375-4.75947393e-01j -0.24548549-9.69400266e-01j
 0.54694816-8.37166478e-01j  0.9863613  -1.64594590e-01j
 0.78914051+6.14212713e-01j  0.08257935+9.96584493e-01j
-0.67728157+7.35723911e-01j -1.          +6.12323400e-16j]
onchip@onchipd-112550:~/Documents/Alex/2019-2/ComuII/B1A.G6_LAB_COMU2/Lab1$

```

Visual Studio Code:



```

phase1.py - Lab1 - Visual Studio Code
File Edit Selection View Go Debug Terminal Help

EXPLORER
  OPEN EDITORS
    X phase1.py
  LAB1
    phase1.py

phase1.py
2
3 import numpy as np
4
5 x = np.array([1, 2, 3, 4, 5]); y = np.array([6, 7, 8, 9, 10])
6 print("x = ", x)
7 print("y = ", y)
8 z = x + y
9 print("z = x + y = ", z, "\n")
10 h = np.cos(x)
11 print("h = cos(x) = ", h, "\n")
12 t = np.linspace(0, 5, 20)
13 g = np.exp(1j*np.pi*t)
14 print("g = exp(j2*pi*t) = ", g)
15
16
17

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Python

onchip@onchipd-112550:~/Documents/Alex/2019-2/ComuII/B1A.G6_LAB_COMU2/Lab1$ python3 phase1.py
x = [1 2 3 4 5]
y = [6 7 8 9 10]
z = x + y = [ 7  9 11 13 15]

h = cos(x) = [ 0.54030231 -0.41614684 -0.9899925  -0.65364362  0.28366219]

g = exp(j2*pi*t) = [ 1.          +0.00000000e+00j  0.67728157+7.35723911e-01j
-0.08257935+9.96584493e-01j -0.78914051+6.14212713e-01j
-0.9863613 -1.64594590e-01j -0.54694816-8.37166478e-01j
0.24548549-9.69400266e-01j  0.87947375-4.75947393e-01j
0.94581724+3.24699469e-01j  0.40169542+9.15773327e-01j
-0.40169542+9.15773327e-01j -0.94581724+3.24699469e-01j
-0.87947375-4.75947393e-01j -0.24548549-9.69400266e-01j
0.54694816-8.37166478e-01j  0.9863613 -1.64594590e-01j
0.78914051+6.14212713e-01j  0.08257935+9.96584493e-01j
-0.67728157+7.35723911e-01j -1.          +6.12323400e-16j]
onchip@onchipd-112550:~/Documents/Alex/2019-2/ComuII/B1A.G6_LAB_COMU2/Lab1$
  
```

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

Registro:

Alex Mantilla
AlexMantilla1

Set status

Joined 22 days ago

Pull requests Issues Marketplace Explore

Overview Repositories 3 Projects 0 Packages 0 Stars 1 Followers 0 Following 0

Popular repositories

Bunnie_Commu

In this repository is saved the Bunnie's communication work.

C++ ★ 1

Hello-World

Description

BIA.G6_LAB_COMU2

This is an academic repository for Digital Communications class at UIS, Bucaramanga, Colombia.

10 contributions in the last year

Learn how we count contributions.

Contribution settings ▾

Contribution activity

September 2019

- Created 6 commits in 3 repositories
- AlexMantilla1/Bunnie_Commu 4 commits
- AlexMantilla1/BIA.G6_LAB_COMU2 1 commit
- AlexMantilla1/Hello-World 1 commit

Repositorio recién creado:

AlexMantilla1 / B1A.G6_LAB_COMU2

Unwatch

1

Star

0

Fork

0

<> Code

Issues 0

Pull requests 0

Projects 0

Wiki

Security

Insights

Settings

This is an academic repository for Digital Communications class at UIS, Bucaramanga, Colombia.

Edit

Manage topics

1 commit

1 branch

0 releases

1 contributor

Branch: master

New pull request

Create new file

Upload files

Find File

Clone or download

AlexMantilla1 Initial commit

Latest commit 2da5e65 1 hour ago

README.md

Initial commit

1 hour ago

README.md

B1A.G6_LAB_COMU2

This is an academic repository for Digital Communications class at UIS, Bucaramanga, Colombia.

First push:

```
onchip@onchipd-112550: ~/Documents/Alex/2019-2/ComuII/B1A.G6_LAB_COMU2
File Edit View Search Terminal Help
onchip@onchipd-112550:~/Documents/Alex/2019-2/ComuII/B1A.G6_LAB_COMU2$ git status
On branch master
Your branch is up-to-date with 'origin/master'.
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    new file:   Lab1/phase1.py

Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

    deleted:    Lab1/phase1.py

Untracked files:
  (use "git add <file>..." to include in what will be committed)

    Lab1/Alex/

onchip@onchipd-112550:~/Documents/Alex/2019-2/ComuII/B1A.G6_LAB_COMU2$ ls
Lab1  README.md
onchip@onchipd-112550:~/Documents/Alex/2019-2/ComuII/B1A.G6_LAB_COMU2$ git add Lab1/
onchip@onchipd-112550:~/Documents/Alex/2019-2/ComuII/B1A.G6_LAB_COMU2$ git commit -m "First Push: Alex's Lab1"
[master 59902f2] First Push: Alex's Lab1
 1 file changed, 16 insertions(+)
 create mode 100644 Lab1/Alex/phase1.py
onchip@onchipd-112550:~/Documents/Alex/2019-2/ComuII/B1A.G6_LAB_COMU2$ git push origin
HEAD        master      origin/HEAD
onchip@onchipd-112550:~/Documents/Alex/2019-2/ComuII/B1A.G6_LAB_COMU2$ git push origin master
Username for 'https://github.com': AlexMantilla1
Password for 'https://AlexMantilla1@github.com':
Counting objects: 5, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (5/5), 566 bytes | 0 bytes/s, done.
Total 5 (delta 0), reused 0 (delta 0)
To https://github.com/AlexMantilla1/B1A.G6_LAB_COMU2.git
   2da5e65..59902f2 master -> master
onchip@onchipd-112550:~/Documents/Alex/2019-2/ComuII/B1A.G6_LAB_COMU2$
```