



Features

Business

Explore

Marketplace

Pricing

This repository Search

Sign in or Sign up

jbenavidesv87 / FlujoRedes

Watch

1

Star

0

Fork

0

Code

Issues 0

Pull requests 0

Projects 0

Insights

Branch: master FlujoRedes / ejemplos / 01GrafoSimple /

Create new file

Find file

History



jbenavidesv87 La función de Color ahora acepta valores RGB y no sólo RGBA

Latest commit 1eaf6af Mar 1, 2018

..

arco.gnu	La función de Color ahora acepta valores RGB y no sólo RGBA	Mar 1, 2018
arco.png	Ejemplo 1 completo:	Feb 28, 2018
grafo.gnu	Ejemplo 1 completo:	Feb 28, 2018
grafo.png	Ejemplo 1 completo:	Feb 28, 2018
main.py	Ejemplo 1 completo:	Feb 28, 2018
readme.md	-Controlada la eventualidad de que un usuario nombrara a un grafo con...	Mar 1, 2018

readme.md

Ejemplo 1. Grafo simple

Para crear un grafo G basta con escribir el siguiente código:

```
from Grafo import Grafo
from Nodo import Nodo

G = Grafo()
```

Por defecto, un grafo inicializado de esta manera tiene las siguientes propiedades:

- Nombre: "grafo" (cadena de texto)
- Dirigido: No

Para agregar un nodo n_1 , hay que declararlo primero y luego añadirlo a un grafo previamente existente:

```
[...]
n1 = Nodo()
G.AgregarNodo(n1)
```

De esta manera, el grafo G contendrá al nodo n_1 que, a su vez, tendrá las siguientes propiedades iniciales:

- Identificador: "1" (cadena de texto)
- Radio: 0.1
- Color: Gris (80808000_{hex})

- Posición: (0.5, 0.5)

Generar una imagen PNG con nombre de archivo `grafo.png` y título **Ejemplo 1.1 Un nodo**, requiere una línea de código más:

```
[...]
G.DibujarGrafo("Ejemplo 1.1 Un nodo")
```

Hasta aquí, el código de este primer ejemplo queda como sigue:

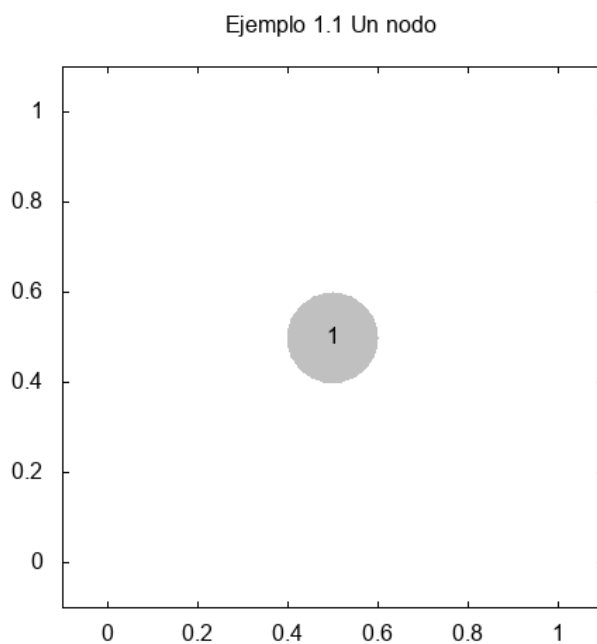
```
from Grafo import Grafo
from Nodo import Nodo

G = Grafo()
n1 = Nodo()
G.AgregarNodo(n1)

G.DibujarGrafo("Ejemplo 1.1 Un nodo")
```

Cabe señalar que antes de obtener la imagen PNG, se crea un archivo GNU nombrado `grafo.gnu` (por el nombre del grafo) que contiene las instrucciones para que `gnuplot` procese la información del grafo extraída de python.

La imagen del grafo generado hasta este punto muestra un eje de coordenadas con título **Ejemplo 1.1 Un nodo** donde aparece plasmado un nodo en color gris, con centro en (0.5, 0.5), radio 0.1 e identificador "1":



De momento [falta agregar conexiones de un mismo nodo consigo mismo], para trazar arcos en un grafo hacen falta al menos dos nodos vecinos que ocupen una posición distinta, por lo que primero se tendría que agregar otro nodo, n_2 , al grafo. En esta ocasión, tras crear el nodo, se modificará su identificador y posición, para finalmente agregarlo al grafo G antes definido, todo lo cual se hará mediante las instrucciones:

```
[...]
n2 = Nodo()
n2.id = "2" # Identificador modificado
n2.posicion = (1,1) # Centro en (1, 1)
G.AgregarNodo(n2)
```

Enseguida se conectan ambos nodos para establecer su vecindad mediante la instrucción:

```
[...]
G.ConectarNodos(n1, n2)
```

Un arco creado de este modo, tiene de manera predefinida un peso de 1 y una conexión bidireccional entre ambos nodos, ya

que al no ser un grafo dirigido, se considera que ambos nodos están conectados entre sí.

Por último, se cambia el nombre del grafo `aarco` y se dibuja bajo el título **Ejemplo 1.2 Un arco**. Esto generará los archivos `arco.gnu` y `arco.png` correspondientes al nuevo nombre del grafo. El código para realizar esto es:

```
[...]
G.nombre = "arco"
G.DibujarGrafo("Ejemplo 1.2 Un arco")
```

La nueva imagen muestra a n_1 acompañado de n_2 con sus propiedades predeterminadas pero con centro en (1, 1) e identificador "2"; ambos nodos unidos mediante un arco que simboliza su vecindad, como se aprecia en la imagen:

