

Python y sistemas eléctricos de potencia

Francisco Escobar Prado

`francisco.escobarprado@ucr.ac.cr`

28 de abril de 2021

Contenidos

1. Motivación

¿Por qué escribir nuestros propios programas?

¿Por qué utilizar Python?

2. Introducción a Python

3. Aplicaciones a sistemas de potencia

Demanda residencial

Calidad de la energía

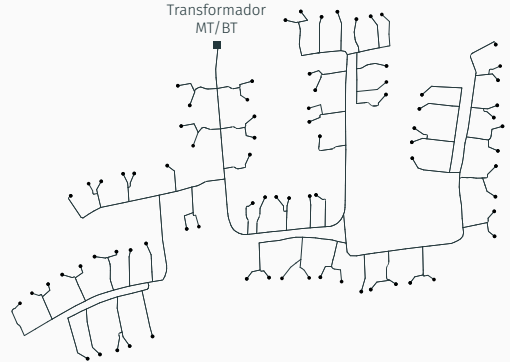
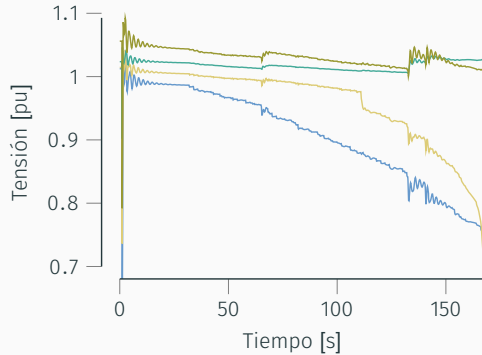
Flujo de potencia

4. Conclusión

Motivación

¿Por qué escribir nuestros propios programas?

¿Por qué escribir nuestros propios programas?



¿Por qué utilizar Python?



Python, en cambio, ...

- es *fácil* de aprender,
- favorece la *comunicación*,
- reduce los *tiempos* de desarrollo,
- es *universal* y
- ofrece numerosas *bibliotecas* competitivas.



¿Por qué utilizar Python?



Python, en cambio, ...

- es *fácil* de aprender,
- favorece la *comunicación*,
- reduce los *tiempos* de desarrollo,
- es *universal* y
- ofrece numerosas *bibliotecas* competitivas.



Ejemplo de código

Sucesión de Fibonacci: $f_0 = 0, f_1 = 1$ y $f_n = f_{n-1} + f_{n-2}$
0, 1, 1, 2, 3, 5, 8, ...

```
def fib(n):  
    return n if n < 2 else fib(n - 1) + fib(n - 2)
```

```
def print_fib(n):  
    a, b = 0, 1  
    while a < n:  
        print(a, end=' ')  
        a, b = b, a+b  
    print()
```


The joy of coding Python should be in seeing short, concise, readable classes that express a lot of action in a small amount of clear code, not in reams of trivial code that bores the reader to death.

— Guido van Rossum



Introducción a Python

Linealidad ↓

```
webinar.py: # Recordatorio
            for i in range(100):
                print('¡Hoy es el webinar de Python!')

            # Itinerario
            tareas = {6: 'Desayunar',
                      7: 'Prepararme',
                      8: 'Asistir al webinar'}

            with open('tareas.txt', 'w') as f:
                for hora in tareas.keys():
                    linea = 'A las ' + str(hora) \
                           + ' debo ' + tareas[hora] + '\n'
                    f.write(linea)
```

Modo interactivo

Aplicaciones a sistemas de potencia

Demanda residencial

Calidad de la energía

Flujo de potencia

Conclusión



PowerFactory



y más...

- La programación alivia los estudios de sistemas de potencia.
- Python es un lenguaje versátil.
- Las implementaciones en Python son más sencillas que en otros lenguajes.
- Se dispone de numerosos módulos competitivos.

Python y sistemas eléctricos de potencia

Francisco Escobar Prado

`francisco.escobarprado@ucr.ac.cr`

28 de abril de 2021