



PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE  
ESCUELA DE INGENIERÍA  
DEPARTAMENTO DE CIENCIA DE LA COMPUTACIÓN

IIC2233 - Programación Avanzada  
1º semestre 2015

# Actividad 16

## Testing de Sistema en pytest

### Descripción

El SAIDING últimamente ha sufrido ataques a sus servidores, por lo que los encargados decidieron aumentar la seguridad agregando encriptación a la información.

Usted deberá ser el encargado de testear el encriptador, no es necesario saber como funciona este, solo debe encargarse de testear algunas funcionalidades.

Lo único que sabe del sistema es lo siguiente:

- El encriptador es simétrico, es decir, para encriptar y desencriptar se usa la misma función. Este está descrito en la clase `Encoder`. El método para encriptar se llama `encrypt`.
- El encriptador se basa en 2 clases llamadas `Rotor` y `Reflector`. Para iniciar las clases se leen multiples archivos, una lista de rotores y otro archivo llamado reflector.
- Antes de la encriptación se debe entregar un alfabeto, este dice las palabras que son aceptadas o no. Para el testeo y archivos del programa puede asumir que el alfabeto es el abecedario inglés en minúscula, por lo que para testear debe hacer un setup haciendo un llamado a `create_alphabet` como se muestra en el `main`.
- Si se ingresa una palabra a encriptar con caracteres que no están en el alfabeto, el programa levantará una excepción `ValueError`

### Rotor

Toma un archivo de texto y crea una función a partir de ella. No es necesario que sepa como funciona, solo tiene que comprobar propiedades de la función.

La aplicación de esta función se hace a través del método `get`. Si el valor no esta en el dominio retorna `None`. Para que funcione correctamente es necesario que esta función sea biyectiva.

## Reflector

Toma un archivo de texto. El reflector es igual al rotor, solo que tiene la función tiene simetría. Si  $f(x) = y$  entonces  $f(y) = x$ .

La función en la clase se accede a través del método `get`. Si el valor no está en el dominio retorna `None`. Debe mostrar que cumple la biyección y simetría. Para testear simetría basta ver que si `get(x)=y` con  $y \neq \text{None}$  entonces se debe cumplir que `get(y)=x`.

## Encriptador

Tiene como parámetros de inicialización una lista con las direcciones de los rotores y la dirección del reflector.

Para testear que la encriptación es correcta debe probar para la siguiente lista de strings que llamando a `encrypt` la palabra queda distinta y llamándolo nuevamente la palabra queda descriptada.

```
lista= ['thequickbrownfoxjumpsoverthelazydog', 'python', 'bang', 'libelula', 'csharp']
```

Para testear de la excepción basta probar con una palabra cualquiera que tenga caracteres que no están al alfabeto. Puede probar por ejemplo con `'ñandu'`.

## Notas

- Puede basarse en el programa `encoder_test.py` para hacer su solución.
- Revise el `main` por si tiene dudas de como usar el encriptador.
- Para testear la biyección de las funciones puedes llamar a la función `check_bijection` del código base que recibe a un rotor o a un reflector y comprueba que cumpla la biyección.

## To-DO

- (1.00 pts) Inicialice el alfabeto como el abecedario inglés en minúsculas con un setup de módulo.
- (1.50 pts) Compruebe que la función del rotor es biyectiva. Debe probarlo para los 3 archivos adjuntos (`rotor1.txt`, `rotor2.txt`, `rotor3.txt`)
- (1.50 pts) Compruebe que la función del reflector es biyectiva y simétrica. Debe probarlo para el archivo adjunto (`reflector.txt`)
- (1.50 pts) Compruebe que la encriptación y decriptación sea correcta, esto incluye testear que levante la excepción cuando es necesario.

**Todo lo anterior debe ser desarrollado en `pytest`.**