

Reto 1

Reto de Programación: Constructor de Cifradores César

Objetivo:

En este reto, crearás un constructor de cifradores César.

El cifrado César es una técnica de criptografía muy sencilla que consiste en desplazar cada letra de un mensaje un número fijo de posiciones en el alfabeto. Por ejemplo, con una distancia de desplazamiento de 3, 'A' se convierte en 'D', 'B' en 'E', y así sucesivamente. El descifrado se realiza desplazando en la dirección opuesta (cambiando el signo de la distancia).

Parte 1: `creaCifrador(d: int) -> Callable`

En esta primera parte, deberás crear una función llamada `creaCifrador(d: int)`. Esta función debe devolver otra función que tome una cadena y aplique el cifrado César con una distancia `d`.

Instrucciones:

1. Define la función `creaCifrador(d: int)`.
2. Dentro de `creaCifrador(d: int)`, define una función interna que tome una cadena de texto y devuelva la cadena cifrada.
3. La función interna debe desplazarse cada letra del alfabeto `d` posiciones.
4. Para simplificar y dado que estamos haciendo una maquina latina, solo trabajaremos con mayúsculas. Y dado que el espacio da pistas sobre la longitud de las palabras elegiremos como alfabeto:

```
1 alfabeto = "ABCDEFGH IJKLMNÑOPQRSTUVWXYZ " # Fíjate que incluye el espacio en blanco
```

5. Recuerda tratar los casos especiales tales como desplazar 2 posiciones en el caso de la letra Z, debe ir a B

Tests Parte 1

1. Probar la función `cifrar(mensaje, d)`

- `cifrar("ZigZag", 1) == "AJHABH"`
- `cifrar("ajhabh", -1) == "ZIGZAG"`

Nota como he elegido palabras que estan en el limite de cifrado para que se vea que el cifrado es circular. Si la letra a cifrar es Z y la distancia es 1, debe devolver A

2. Probar el creador de cifradores, es decir, crear el cifrador `cesar3` y luego probar que funciona

```
1 cesar3 = creaCifrador(3)
2 assert cesar3("ZigZag") == "CLJCDJ"
```

Parte 2 (Opcional): `creaParCesar(d: int) -> tuple(Callable, Callable)`

En esta segunda parte, crearás una función llamada `creaParCesar(d: int)`. Esta función debe devolver dos funciones: una para cifrar y otra para descifrar.

Instrucciones:

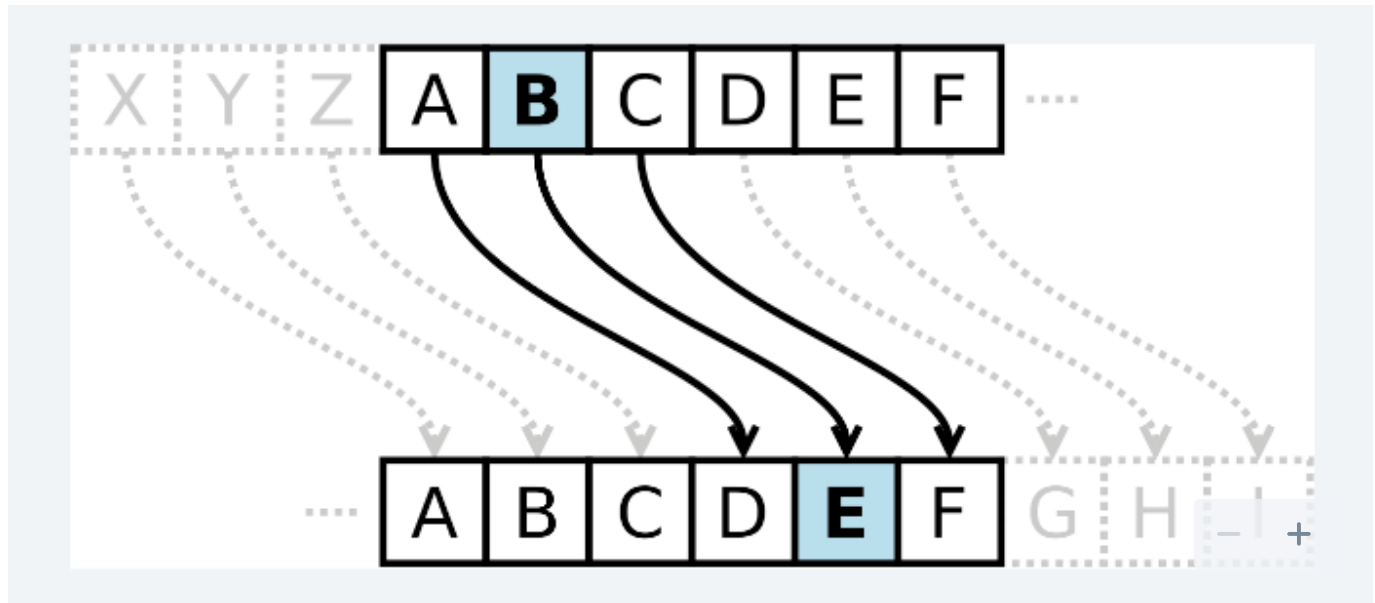
1. Define la función `creaParCesar(d: int)`.
2. Dentro de `creaParCesar(d: int)`, define una función de cifrado que funcione igual que en la Parte 1.
3. Define también una función de descifrado que aplique el cifrado César con una distancia `-d`.
4. Devuelve ambas funciones.

Tests Parte 2

1. Necesitaras un test similar al dos de la parte 1. Crear un par `cifrador`, `descifrador` con la nueva funcion `creaParCesar` y luego hacer un test para cada uno de los elementos obtenidos. Uno para `cifrador` y otro para `descifrador`

Explicación del Cifrado César:

- El cifrado César desplaza cada letra del mensaje un número fijo de posiciones en el alfabeto. Por ejemplo, con un desplazamiento de 3, 'A' se convierte en 'D', 'B' en 'E', etc.



distancia 3

- Para descifrar el mensaje, simplemente se desplaza en la dirección opuesta, es decir, utilizando la distancia `-d`.

Misma imagen que arriba pero con la flecha cambiada de sentido, es decir, desde D a A, de E a B,...

Tu Tarea:

Implementa las funciones `creaCifrador(d: int)` y, si te animas, también `creaParCesar(d: int)` siguiendo las instrucciones proporcionadas.