

Universidad Del Norte

Compiladores NRC 7410

Laboratorio 1 - Expresiones Regulares

Rubens Andre Apresa Echeverría Felipe José Benítez Avilez Laura Sofía Gómez Rosales Luna Julio Martínez

Prof. José Márquez



Índice

1.	Ejercicio 12	2
	1.1. Expresión Regular	4
	1.2. Código de validación	
	1.3. Pruebas	•
2.	Ejercicio 15	4
	2.1. Expresión Regular	4
	2.2. Código de validación	
	2.3. Pruebas	
3.	Cuaderno de Colab	7



1. Ejercicio 12

Validar una matrícula de un vehículo con formato XXXNNN, donde X es una letra y N es un dígito. En total hay seis caracteres. Puede aceptar mayúsculas y minúsculas.

Salida esperada:

- Si la placa es válida (tres letras seguidas de tres dígitos), muestra: Placa válida
- Si la placa no es válida, muestra: Placa inválida

1.1. Expresión Regular

```
[a-z,A-Z]{3}[0-9]{3}
```

- [a-z, A-Z] {3}: Captura tres letras, donde cada letra puede ser mayúscula o minúscula (Grupo 1).
- [0-9] {3}: Captura tres dígitos (Grupo 2).

1.2. Código de validación

```
def validar_placa(placa):
    veri = re.fullmatch('[a-z,A-Z]{3}[0-9]{3}', placa)
    if veri:
        return print('Placa valida')
    else:
        return print('Placa invalida')
```

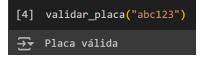


1.3. Pruebas

■ Entrada: ABC123



■ Entrada: abc123



■ Entrada: AB1234



■ Entrada: A1B2C3





2. Ejercicio 15

Desglosar una dirección de la ciudad de Barranquilla, de la cual se pueda obtener si vive en Calle (Cl), Carrera (Cra), Diagonal (Diag) o Transversal (Tr); luego el número (seguido de una letra, que es opcional); seguido de la nomenclatura del predio la cual contiene un primer número seguido de una letra (puede o no ser precedido de # o No), seguido o no de un guion medio (-) y luego otro número. Ejemplo: Cra. 47B No. 74C-45 Salida:

■ Vía: Cra

■ Número Vía: 47B

Número Vía origen dirección: 74C

■ Número predio: 45

2.1. Expresión Regular

```
\s*
(Cra|C1|Tr|Diag)\.?
\s*([0-9]+[a-z,A-Z]?)
\s*(No\.?\s*|#)?
([0-9]+[a-z,A-Z])
\s*-?\s*([0-9]+)
\s*
```

- \s*: Captura espacios opcionales a lo largo de la dirección.
- (Cra|Cl|Tr|Diag)\.?: Captura la abreviatura de la vía (Grupo 1), como Cra, Cl, Tr o Diag, seguido opcionalmente por un punto.
- \s*([0-9]+[a-z,A-Z]?): Captura el número de la vía (Grupo 2), que puede incluir uno o más dígitos y opcionalmente una letra.
- \s*(No\.?\s*|#)?: Captura opcionalmente 'No.' o '#' seguido de espacios.
- ([0-9]+[a-z, A-Z]): Captura el número de la vía origen dirección (Grupo 4), con uno o más dígitos seguido de una letra.
- \s*-?\s*([0-9]+): Captura un guion opcional seguido de espacios opcionales y luego el número del predio (Grupo 5), que puede tener uno o más dígitos.



2.2. Código de validación

```
def validar_direccion(dir):
    ver = re.fullmatch(r"\s*(Cra|Cl|Tr|Diag)\.?\s*([0-9]+[a-z,A-Z]?)\s*(No
    \.?\s*|#)?([0-9]+[a-z,A-Z])\s*-?\s*([0-9]+)\s*", dir)

if ver:
    print("Direccion:", ver.group(0))
    print("Via:", ver.group(1))
    print("Numero de Via:", ver.group(2))
    print("Numero de Via Origen Direccion:", ver.group(4))
    print("Numero Predio:", ver.group(5))

else:
    print("Direccion invalida")
```

2.3. Pruebas

■ Entrada: Cra. 47B No. 74C-45

```
[10] validar_direccion("Cra. 47B No. 74C-45")

Dirección: Cra. 47B No. 74C-45

Vía: Cra

Número de Vía: 47B

Número de Vía Origen Dirección: 74C

Número Predio: 45
```

■ Entrada: Cl 52 No. 21-15

```
[15] validar_direccion("Cl 52 No. 21-15")

The process of the proc
```

■ Entrada: Tr 10A No. 30B50

```
validar_direccion("Tr 10A No. 30B50")

Dirección: Tr 10A No. 30B50

Vía: Tr

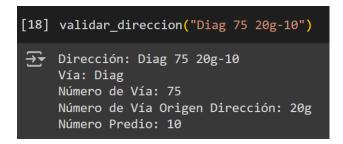
Número de Vía: 10A

Número de Vía Origen Dirección: 30B

Número Predio: 50
```



■ Entrada: Diag 75 20g-10



■ Entrada: Cra. 15 30A

[20] validar_direccion("Cra. 15 30A")

Dirección inválida



3. Cuaderno de Colab

Expresiones Regulares 12 y 15