



Mòdul 4 - Llenguatge Marques i SGI

UF2 – RA3 ÀMBITS D'APLICACIÓ DE L'XML

Pràctica 9: Expressions Regulars (REGEX)

Silvia Sandoval Maldonado

2023-204

ÍNDICE


1. Exercici 1: Analitza documents XML	3
2. Exercici 2: Analitza documents JSON	5
3. Exercici 3: Troba les paraules	15

Exercici 1: Analitza documents XML

Clona el repositori <https://github.com/pauitic/practica9>

Escriu les expressions regulars que seleccionin els continguts que s'indiquen del fitxer **xml_for_regex.xml**. Per cada exercici, trobaràs una captura de pantalla que especifica la manera que s'ha de fer la captura de caràcters.

1. Selecciona les etiquetes **<price>** i el seu contingut.



```
<price>.*
```

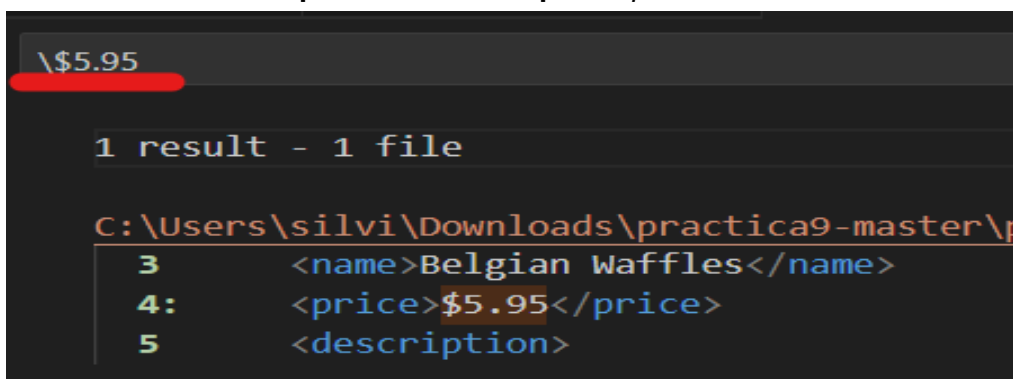
5 results - 1 file

C:\Users\silvi\Downloads\practica9-master\practica9-master\xml_for_regex.xml

```
3      <name>Belgian Waffles</name>
4:     <price>$5.95</price>
5      <description>

11     <name>Strawberry Belgian Waffles</name>
12:    <price>$7.95</price>
13    <description>
```

2. Selecciona els preus sense l'etiqueta **<price>**



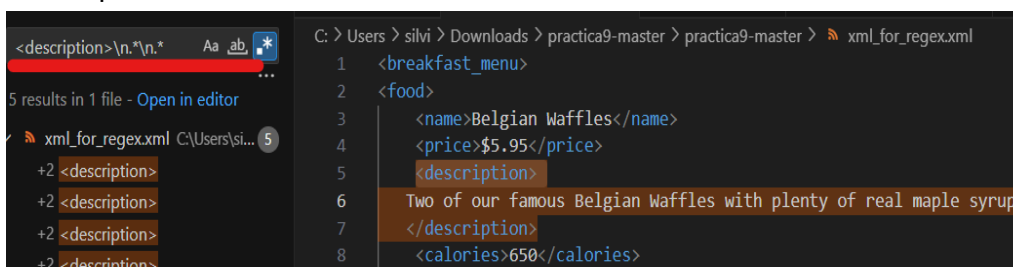
```
\$.95
```

1 result - 1 file

C:\Users\silvi\Downloads\practica9-master\practica9-master\xml_for_regex.xml

```
3      <name>Belgian Waffles</name>
4:     <price>$5.95</price>
5      <description>
```

3. Selecciona les etiquetes **<description>** i el seu contingut. Compte que ara poden haver-hi salts de línia!



```
<description>\n.*\n.*
```

5 results in 1 file - Open in editor

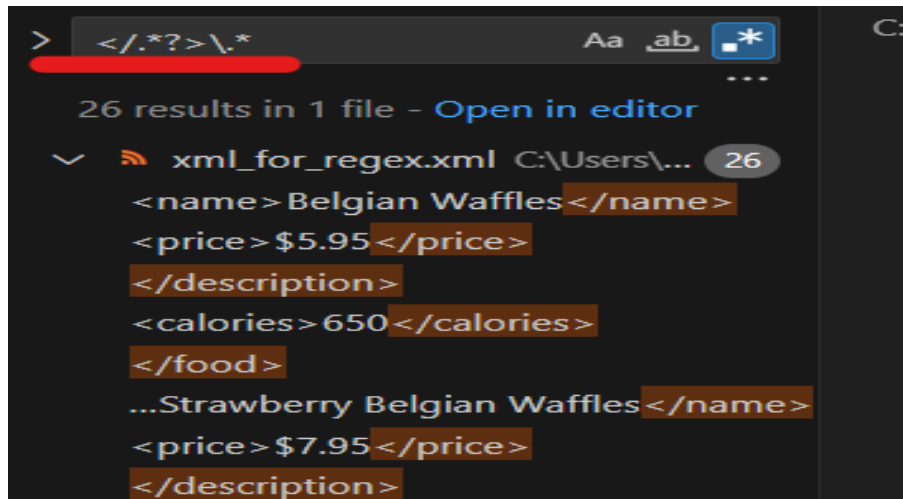
xml_for_regex.xml C:\Users\silvi\Downloads\practica9-master\practica9-master\xml_for_regex.xml

```
+2 <description>
+2 <description>
+2 <description>
+2 <description>
```

C:\Users\silvi\Downloads\practica9-master\practica9-master\xml_for_regex.xml

```
1  <breakfast_menu>
2  <food>
3      <name>Belgian Waffles</name>
4      <price>$5.95</price>
5      <description>
6          Two of our famous Belgian Waffles with plenty of real maple syrup
7      </description>
8      <calories>650</calories>
```

4. Selecciona totes (i només) les etiquetes de tancament.



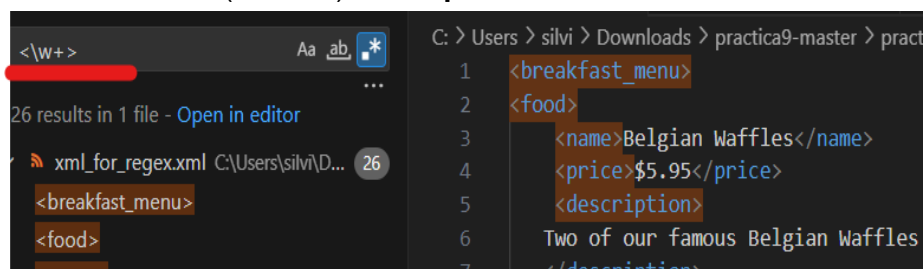
```
> </.*?>\\.*
```

26 results in 1 file - Open in editor

xml_for_regex.xml C:\Users\... 26

```
<name>Belgian Waffles</name>
<price>$5.95</price>
</description>
<calories>650</calories>
</food>
...Strawberry Belgian Waffles</name>
<price>$7.95</price>
</description>
```

5. Selecciona totes (i només) les **etiquetes d'obertura**.



```
<w+>
```

26 results in 1 file - Open in editor

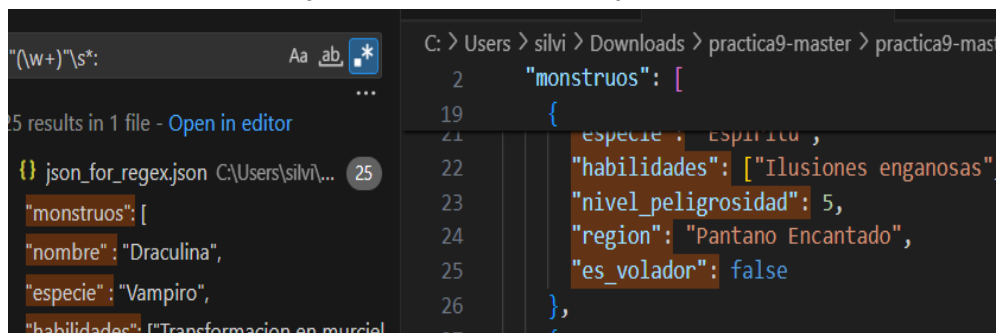
xml_for_regex.xml C:\Users\silvi\D... 26

```
<breakfast_menu>
<food>
  <name>Belgian Waffles</name>
  <price>$5.95</price>
  <description>
    Two of our famous Belgian Waffles
  </description>
```

Exercici 2: Analitza documents JSON

Desenvolupa una expressió regular específica per capturar les cadenes de caràcters indicades en el fitxer **json_for_regex.json**. L'expressió regular que utilitzis ha de servir per capturar els *strings* d'aquest document, i no ha de ser genèrica en cap cas.

6. Selecciona totes les **keys** del document JSON juntament amb els dos punts.



```
"(w+)\"\\s*:
```

25 results in 1 file - Open in editor

json_for_regex.json C:\Users\silvi\... 25

```
{
  "monstruos": [
    {
      "nombre": "Draculina",
      "especie": "Vampiro",
      "habilidades": ["Transformacion en murcielago"]
    }
  ]
}
```

Entre la comillas abrimos los paréntesis y dentro aplicamos el **carácter alfanumérico** `w` cerramos paréntesis con el **carácter invisible** `\s` seleccionamos todos los keys sin olvidar los **dos puntos**:

7. Selecciona tots¹ els **valors** (*values*) JSON. Pots utilitzar com a referència els dos punts anteriors i la coma, com es mostra a la imatge.

¹ Excepte el valor de la clau "monstruos"

```
C:\Users\silvi\Downloads\practica9-master\practica9-master\json_for_regex.json:
6:      "habilidades": ["Transformacion en murcielago", "Control mental"],
7:      "nivel_peligrosidad": 8,
8:      "region": "Transilvania",
9:      "es_volador": true
```

8. Selecciona les **llistes** de *strings* del document.

3 results - 1 file

```
C:\Users\silvi\Downloads\practica9-master\practica9-master\json_for_regex.json:
```

```
5     "especie" : "Vampiro",
6:     "habilidades": ["Transformacion en murcielago", "Control mental"],
7     "nivel_peligrosidad": 8,

13    "especie": "Cefalopodo Gigante",
14:    "habilidades" : ["Control del mar", "Tentaculos gigantes"],
15    "nivel_peligrosidad": 10,

29    "especie": "Criatura del Hielo",
30:    "habilidades": ["Camuflaje en la nieve", "Fuerza sobrehumana"],
31    "nivel_peligrosidad": 10,
```

9. Selecciona els **booleans**. Compte no seleccionar els strings “true” i “false” dins de *strings*.

```
(?<!" )\b(true|false)\b(?!" )
```

4 results - 1 file

```
C:\Users\silvi\Downloads\practica9-master\practica9-master
```

```

8      "region": "Transilvania",
9      "es_volador": true
10   },

16      "region" : "Oceano Pacifico",
17      "es_volador": false
18   },

24      "region": "Pantano Encantado",
25      "es_volador": false
26   },

32      "region": "Montanas Heladas",
33      "es_volador": false

```

10. Selecciona els **strings**, però no les *keys* (si t'ajuda, pots seleccionar les comes i els `]` tal com es mostra a la imatge)

```
(?<=:s\\D(".*?"|\\.[^?\\\\](?=,|s\\N\\N))
```



```
C:\\Users\\silvi\\Downloads\\practica9-master\\practica9-master\\json_for_regex.json:
```

```
4:     "nombre": "Draculina",
5:     "especie": "Vampiro",
6:     "habilidades": ["Transformacion en murcielago", "Control mental"],
7:     "nivel_peligrosidad": 8,
8:     "region": "Transilvania",
9:     "es_volador": true

11  {
12:     "nombre": "Kraken",
13:     "especie": "Cefalopodo Gigante",
14:     "habilidades": ["Control del mar", "Tentaculos gigantes"],
15:     "nivel_peligrosidad": 10,
16:     "region": "Oceano Pacifico",
17:     "es_volador": false
```

- \d\d: Esto significa que debe haber dos dígitos consecutivos.

- [-/]: Representa un guió o una barra diagonal.
- ([012]\d): Este grupo se refiere a un mes. Puede ser 01, 02 o 12.
- [-/]: Nuevamente, esto representa un guió o una barra diagonal.
- \d\d\d\d: Esto significa que debe haber cuatro dígitos consecutivos.

d.

`[0123]\d[-/](([012]\d) | [a-z]{3}) [-/]\d\d\d\d`

01/nov/2024

- `[0123]\d`: Esto significa que debe haber un dígito que sea 0, 1, 2 o 3, seguido de otro dígito numérico.
- `[-/]`: Un guió o una barra diagonal. Puede coincidir con cualquiera de estos caracteres. podría ser “-” o “/”.
- `(([012]\d)[a-z]{3})`: Contiene dos opciones:
 - `[012]\d`: Esto significa que debe haber un dígito que sea 0, 1 o 2, seguido de otro dígito numérico.
 - `[a-z]{3}`: Esto significa que debe haber tres letras minúsculas consecutivas. podría coincidir con “abc” o “xyz”.
- `[-/]`: Nuevamente, esto representa un guió o una barra diagonal.
- `\d\d\d\d`: Esto significa que debe haber cuatro dígitos consecutivos.

e.

`\w*\.(jpg|png|pdf)`

foto.jpg

imagen.png

documento.pdf

- `\w*`: coincide con cualquier carácter alfanumérico, incluyendo el guió bajo (_). El asterisco * significa que puede haber cero o más de estos caracteres.
- `\.`: El punto normalmente coincide con cualquier carácter, pero cuando se escapa con una barra invertida \, se toma literalmente como un punto en el texto.
- `(jpg|png|pdf)`: Los paréntesis crean un grupo de captura, y el símbolo | funciona como un operador OR. La expresión coincide con jpg, png, o pdf.

Telèfons

Escriu una expressió regex que validi els telèfons espanyols. Tingues en compte que:

- Pot o no començar amb +34
- El número està format per 9 dígitos
- El número comença per 6 o 7 si és mòbil i 8 o 9 si és fix
- Els dígitos poden estar seguits o separats per un guionet o espai

Casos vàlids	Casos invàlids
645540844 64 554 08 44 74-554-08-44 +34 645540844 +34945540844	+34445540844 64554084 +346+45540844 +34-6455--40844 +34 6455 40844

`^(?:\+34\s)?(6|7|8|9)\d{2}(\s|-)?\d{2}(\s|-)?\d{2}(\s|-)?\d{2}$`

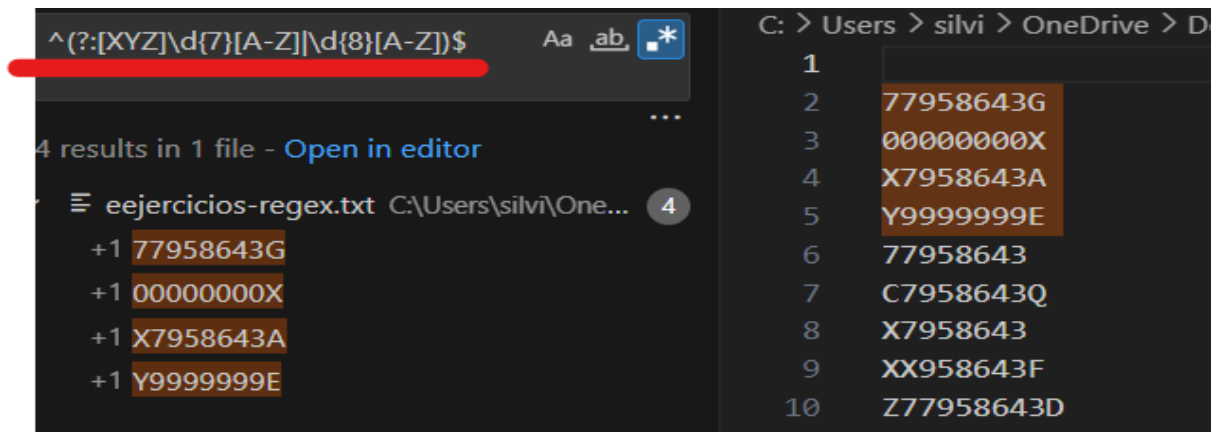
- `^`: Indica el inicio de la línea.
- `(?:\+34)?`: Indica que el +34 es opcional.
- `\s?`: Indica que puede haber un espacio opcional después de +34.
- `(6|7|8|9)`: El número debe comenzar por 6, 7, 8 o 9.
- `\d{2}`: Seguido de dos dígitos.
- `(\s?\d{2}){3}`: Un espacio opcional seguido de dos dígitos, y este patrón se repite tres veces para completar los 9 dígitos necesarios.
- `$`: Indica el final de la línea.

DNI / NIE

Escriu una expressió *regex* pels DNIs i NIE

- Els DNI tenen **8 números** i un **dígit de control** alfabètic
- Els NIE comencen per **X, Y o Z**, tenen **7 nombres** i un dígit de **control** alfabètic

Casos vàlids	Casos invàlids
77958643G 00000000X X7958643A Y9999999E	77958643 C7958643Q X7958643 XX958643F Z77958643D



`^(?:[XYZ]\d{7}[A-Z]|\d{8}[A-Z])$`

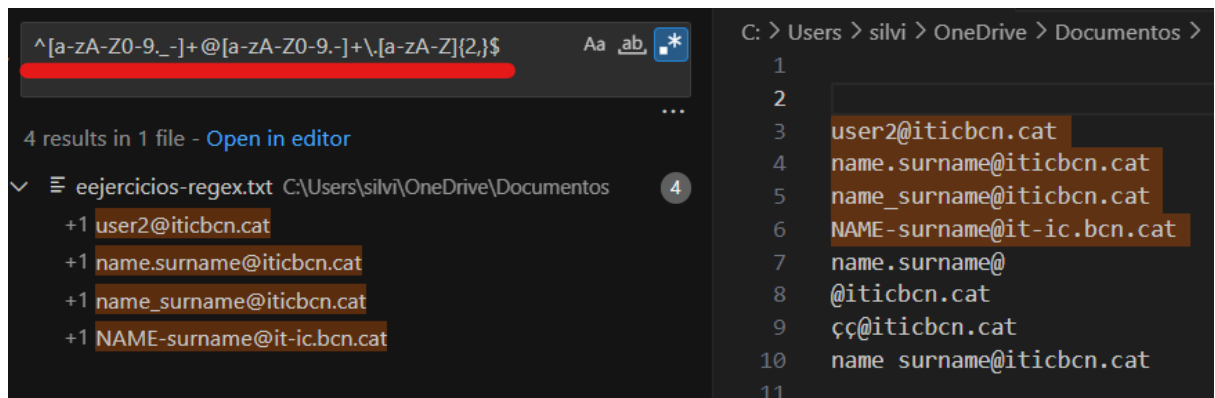
- `^`: Inicio de la cadena.
- `(?:`: Comienza un grupo no captador, lo que permite agrupar varias opciones sin capturarlas para referencias posteriores.
- `[XYZ]`: Coincide con cualquiera de los caracteres 'X', 'Y' o 'Z'.
- `\d{7}`: Coincide con exactamente 7 dígitos.
- `[A-Z]`: Coincide con una única letra mayúscula.
- `|`: Operador OR, significa "o" en regex.
- `\d{8}`: Coincide con exactamente 8 dígitos (para los DNI).
- `[A-Z]`: Coincide nuevamente con una única letra mayúscula (para la letra de control de los DNI).
- `)`: Cierra el grupo no captador.
- `$`: Ancla el final de la cadena.

Correus electrònics

Escriu una expressió regex que validi els emails seguint les següents condicions:

- La paraula que precedeix l'arrova "@" pot tenir lletres no accentuades, números, guions, punts i barra baixes
- El domini de la direcció pot tenir lletres, punts i guions

Casos vàlids	Casos invàlids
user2@iticbcn.cat name.surname@iticbcn.cat name_surname@iticbcn.cat NAME-surname@it-ic.bcn.cat	name.surname@ @iticbcn.cat çç@iticbcn.cat name surname@iticbcn.cat



`^[a-zA-Z0-9._-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$`

- `^`: Ancla el inicio de la cadena.
- `[a-zA-Z0-9._-]+`: Uno o más caracteres que pueden ser letras mayúsculas o minúsculas, dígitos, puntos, guiones bajos o guiones.
- `@`: El carácter que separa la parte local del dominio en una dirección de correo electrónico.
- `[a-zA-Z0-9.-]+`: Uno o más caracteres que pueden ser letras mayúsculas o minúsculas, dígitos, puntos o guiones para la parte del dominio.
- `\.`: Un punto literal que precede al TLD (dominio de nivel superior).
- `[a-zA-Z]{2,}`: Dos o más letras que conforman el TLD.
- `$`: Ancla el final de la cadena

Dominiis d'URLs

Escriu una expressió regex que validi els dominis dels URL tenint en compte les següents condicions

- L'URL comença per "http://" o "https://"
- El domini pot tenir lletres, guiones, punts
- Pot acabar amb barra

Casos vàlids	Casos invàlids
https://www.educaciodigital.cat/ https://educacio-digital.fr https://www.educacio	educacio-digital.es http://educacio-digital.cat/hoola/404 http://educacio.digital.cat/nomesDomini

```
^(https?:\V)[w-]+(\.[w-]+)+V?$
```

```
C:\Users> Users > silvi > OneDrive > Documentos > eejercicios-regex.txt
22 #Dominios de URLs
23 https://www.educaciodigital.cat/
24 https://educacio-digital.fr
25 https://www.educacio
26 educacio-digital.es
27 http://educacio-digital.cat/hoola/404
28 http://educacio.digital.cat/nomesDomini
29
```

`^(https?:\V)[w-]+(\.[w-]+)+V?$`

- `^`: Indica el inicio de la cadena.
- `(https?:\V)`: Un grupo de captura para el inicio de la URL que debe ser **http://** o **https://**. El signo "?" hace que la "s" sea opcional.
- `[w-]+`: Coincide con uno o más caracteres de palabra (que incluyen letras y dígitos) o guiones. Esto asume que el dominio no comienza con un punto o guión.
- `(\.[w-]+)+`: Un grupo de captura para los segmentos del dominio que comienzan con un punto y están seguidos por uno o más caracteres de palabra o guiones. El signo "+" asegura que haya al menos un segmento de dominio y permite subdominios.
- `V?`: Un carácter opcional de barra al final de la URL.
- `$`: Indica el final de la cadena.

URLs completes

Escriu una expressió regex que validi els URL tenint en compte les següents condicions

- L'URL comença per "http://" o "https://"
- El domini pot tenir lletres, guions
 - El domini no pot tenir subdomini
 - El domini ha de pertànyer a .es, .cat, .org o .edu
- La ruta pot tenir lletres i números, guions, punts i barra baixes
 - A més, es poden incloure paràmetres, i per això s'han de permetre els símbols ? % & i =
- Pot acabar amb barra

Casos vàlids
https://educaciodigital.cat/ http://educacio-digital.cat/apt1/apt3 http://educacio-digital.cat/sim.bo-l_s/me?s?param=1&param2=2
Casos invàlids
http://educacio-digital.cat//DOBLE http://educacio.digital.cat/te_subdomini http://educacio_digital.cat/te_barrabaixa_al_domini https://educacio-digital.fr/fr_no_permes educacio-digital.es https://www.educacio

```

eejercicios-regex.txt • ejercicio.py
C: > Users > silvi > OneDrive > Documentos > eejercicios-regex.txt
22 https://educaciodigital.cat/
23 http://educacio-digital.cat/apt1/apt3
24 http://educacio-digital.cat/sim.bo-l_s/me?s?param=1&param2=2
25 Casos invàlids
26 http://educacio-digital.cat//DOBLE
27 http://educacio.digital.cat/te_subdomini
28 http://educacio_digital.cat/te_barrabaixa_al_domini
29 https://educacio-digital.fr/fr_no_permes
30 educacio-digital.es
31 https://www.educacio
32

```

`^(https?:\W)(?!.*\W)(?!.*te_)(?!.*fr/)([w.-]+\cat)(V[wV.-]*) (\?[\^s]*)? $`

- `^`: Ancla el comienzo de la línea.
- `(https?:\W)`: Captura el inicio de la URL que debe ser `http://` o `https://`.
- `(?!.*\W)`: Un lookahead negativo para asegurarse de que no haya doble barra `//` en ninguna parte de la URL después del protocolo.
- `(?!.*te_)`: Un lookahead negativo para asegurarse de que la secuencia `te_` no esté en ninguna parte de la URL.
- `(?!.*fr/)`: Un lookahead negativo para asegurarse de que la secuencia `fr/` no esté en ninguna parte de la URL.
- `([w.-]+\cat)`: Captura el nombre de dominio que debe terminar con `.cat`.
- `(V[wV.-]*)`: Captura opcionalmente el resto de la URL que puede contener letras, números, guiones, puntos o barras.

- `(\?[^\s]*)?`: Captura opcionalment la cadena de consulta que comença amb un signe d'interrogació seguit de qualsevol cosa que no sigui un espai.

Adreces

Escriu una expressió regex que validi les adreces que segueixin les següents condicions.

- **Comença** per: C/ Av. Pg. Rb
- Segueix del **nom del carrer** que pot ser una o diverses paraules amb lletres majúscules i minúscules accentuades
- Continua amb el **número de porta** que pot tenir diversos dígits
- Pot tenir **número de pis** i **número de porta**
- Continua amb el **nom de la ciutat**, que pot estar formada per diferents paraules
- Acaba amb la **província** entre parèntesis. Només pot ser Barcelona, Girona, Tarragona o Lleida.

Casos vàlids
C/ Diputació 31 1 2 Badalona (Barcelona) Av. Girona 42 1 2 Badalona (Barcelona) Av. Rossello 35 Arbucies (Girona) Rb. Les Rambles 4432 Lleida (Lleida) Av. Gran via de les corts catalanes 32 Santa Coloma de Gramanet (Barcelona) Av. Rosselló 32 1 2 Reus (Tarragona)
Casos invàlids
Av. Gran via de les corts catalanes 32 (Barcelona) Gran via de les corts catalanes 32 Badalona (Barcelona) C/ 32 1 2 Badalona (Barcelona) Av. Rosselló 32 1 2 4 Salt (Girona)

```
> Users > silvi > OneDrive > Documentos > ejercicios-regex.txt
26
27
28 C/ Diputació 31 1 2 Badalona (Barcelona)
29 Av. Girona 42 1 2 Badalona (Barcelona)
30 Av. Rossello 35 Arbucies (Girona)
31 Rb. Les Rambles 4432 Lleida (Lleida)
32 Av. Gran via de les corts catalanes 32 Santa Coloma de Gramanet (Barcelona)
33 Av. Rosselló 32 1 2 Reus (Tarragona)
34 # Casos invàlids
35 Av. Gran via de les corts catalanes 32 (Barcelona)
36 Gran via de les corts catalanes 32 Badalona (Barcelona)
37 C/ 32 1 2 Badalona (Barcelona)
```

`^(CV|Av\.|Rb\.)\s+[\p{L}\s]+\s+\d.\s+\d*\s*[\p{L}\s]?(\s*(Barcelo
na|Girona|Tarragona|Lleida)\s*\s*))?$`

- `^`: Inicio de la cadena.
- `(CV|Av\.|Rb\.)`: Coincide con las abreviaturas de tipos de vías permitidas, seguidas de un punto o una barra, según corresponda.
- `\s+`: Uno o más espacios en blanco.
- `[\p{L}\s]+`: Uno o más caracteres de letras (incluyendo caracteres Unicode para letras acentuadas) o espacios.
- `\s+`: Uno o más espacios en blanco.
- `\d+`: Uno o más dígitos.
- `\s+`: Uno o más espacios en blanco.
- `\d*`: Cero o más dígitos (para el número de piso/puerta que es opcional).
- `\s*`: Cero o más espacios en blanco.
- `[\p{L}\s]*`: Cero o más letras o espacios (para el resto del nombre de la calle o ciudad que es opcional).

Contrasenyes fortes

Dissenya una expressió regex que validi les contrasenyes fortes.

- Com a mínim ha de tenir una lletra **majúscula** i una **minúscula**
- Com a mínim ha de tenir **dos dígits**
- Com a mínim ha d'incloure un dels següents **símbols**: `. _ ? \ [] ()`
- La contrasenya ha de tenir entre **8 i 30 caràcters**

Casos vàlids	Casos invàlids
12345678aA._? aA._?12345678 aA\[]()12345678	123456789 aA77._ 77fghgfAAAAA

```
C: > Users > silvi > OneDrive > Documentos > ≡ eejercici
37
38 12345678aA._?
39 aA._?12345678
40 aA\[ ]()12345678
41 123456789
42 aA77._
43 77fghgfAAAAA
44
45
```

`^(?=.*[A-Z])(?=.*[a-z])(?=.*\d.*\d)(?=.*[-_?()\[\]]).{8,30}$`

- **^**: Comienzo de la cadena.
- **(?=.*[A-Z])**: Afirmación anticipada (lookahead) que garantiza la presencia de al menos una letra mayúscula.
- **(?=.*[a-z])**: Afirmación anticipada que garantiza la presencia de al menos una letra minúscula.
- **(?=.*\d.*\d)**: Afirmación anticipada que garantiza la presencia de al menos dos dígitos.
- **(?=.*[-_?(){}])**: Afirmación anticipada que garantiza la presencia de al menos uno de los símbolos especificados.
- **{8,30}**: Cualquier carácter (excepto salto de línea), con una longitud de entre 8 y 30 caracteres.
- **\$**: Ancla al final de la cadena.