

```
Archivo  Editar  Selección  Ver  Ir  Ejecutar  ...  <  ->  Buscar

leslie.py x
C:\Users\leslie\Desktop> .\leslie.py e
3 def extract_emails(text):
12     return emails
13
14 def extract_comments(code):
15     """
16     Extrae todos los comentarios de un código fuente dado.
17     Soporta comentarios de una línea y multilinea.
18
19     :param code: Cadena de texto que representa el código fuente.
20     :return: Lista de comentarios encontrados.
21     """
22     # Comentarios de una línea
23     single_line_comment_pattern = r'//.*'
24     # Comentarios multilinea
25     multi_line_comment_pattern = r'/*.*(?:\n|\\n)**/'
26
27     # Buscar comentarios de una línea y multilinea
28     single_line_comments = re.findall(single_line_comment_pattern, code)
29     multi_line_comments = re.findall(multi_line_comment_pattern, code)
30
31     return single_line_comments + multi_line_comments
32
33 def validate_date(date):
34     """
35     Valida si una fecha dada está en el formato mm/dd/yyyy y es una fecha válida.
36
37     :param date: Fecha en formato de cadena 'mm/dd/yyyy'.
38     :return: True si la fecha es válida, False en caso contrario.
39     """
40     date_pattern = r'^(0[1-9]|1[0-2])/([0-9]|1[2])/([0-9]{3}|[01])\d{4}$'
41     match = re.match(date_pattern, date)
42
```

```
Archivo  Editar  Selección  Ver  Ir  Ejecutar  ...  <  ->  Buscar

leslie.py x
C:\Users\leslie\Desktop> .\leslie.py e
3 def extract_emails(text):
12     return emails
13
14 def extract_comments(code):
15     """
16     Extrae todos los comentarios de un código fuente dado.
17
18     :param code: Cadena de texto que representa el código fuente.
19     :return: Lista de comentarios encontrados.
20     """
21     # Comentarios de una línea
22     single_line_comment_pattern = r'//.*'
23     # Comentarios multilinea
24     multi_line_comment_pattern = r'/*.*(?:\n|\\n)**/'
25
26     # Buscar comentarios de una línea y multilinea
27     single_line_comments = re.findall(single_line_comment_pattern, code)
28     multi_line_comments = re.findall(multi_line_comment_pattern, code)
29
30     return single_line_comments + multi_line_comments
31
32 def validate_date(date):
33     """
34     Valida si una fecha dada está en el formato mm/dd/yyyy y es una fecha válida.
35
36     :param date: Fecha en formato de cadena 'mm/dd/yyyy'.
37     :return: True si la fecha es válida, False en caso contrario.
38     """
39     date_pattern = r'^(0[1-9]|1[0-2])/([0-9]|1[2])/([0-9]{3}|[01])\d{4}$'
40     match = re.match(date_pattern, date)
41
42     return match is not None
43
```

PROBLEMAS TERMINAL PUERTOS SEARCH ERROR

```
PS C:\Users\leslie> & C:/Users/leslie/AppData/Local/Programs/Python/Python310/python.exe c:/Users/leslie/Desktop/leslie.py
Correos electrónicos encontrados: ['example@gmail.com', 'contacto@empresa.com', 'otro-correo@dominio.org']
Comentarios encontrados: ['// Este es un comentario de una línea', '/* Este es un comentario\n    multilinea */']
¿Fecha válida?: True
PS C:\Users\leslie>
```