

# Package ‘datamedios’

June 23, 2025

**Type** Package

**Title** Scraping Chilean Media

**Version** 1.2.2

**Maintainer** Exequiel Trujillo <exequiel.trujillo@ug.uchile.cl>

**Description** A system for extracting news from Chilean media, specifically through Web Scapping from Chilean media. The package allows for news searches using search phrases and date filters, and returns the results in a structured format, ready for analysis. Additionally, it includes functions to clean the extracted data, visualize it, and store it in databases. All of this can be done automatically, facilitating the collection and analysis of relevant information from Chilean media.

**License** MIT + file LICENSE

**Encoding** UTF-8

**Language** es-ES

**Depends** R (>= 4.1)

**Suggests** rcmdcheck

**Imports** dplyr, httr, magrittr, jsonlite, utils, rlang, wordcloud2, tidytext, lubridate, rvest, stringr, xml2, purrr, DT, ggplot2, plotly

**RoxygenNote** 7.3.2

**NeedsCompilation** no

**Author** Exequiel Trujillo [aut, cre, cph, fnd],  
Ismael Aguayo [aut, fnd],  
Klaus Lehmann [aut, fnd]

**Repository** CRAN

**Date/Publication** 2025-06-23 03:30:02 UTC

## Contents

agregar_datos_unicos . . . . .	2
--------------------------------	---

extraccion_parrafos . . . . .	3
extraer_noticias_fecha . . . . .	3
extraer_noticias_fecha_bbcl . . . . .	4
extraer_noticias_fecha_emol . . . . .	5
extraer_noticias_max_res . . . . .	6
extraer_noticias_max_res_bbcl . . . . .	7
extraer_noticias_max_res_emol . . . . .	7
grafico_comparacion_medios . . . . .	8
grafico_notas_fecha . . . . .	9
grafico_notas_por_mes . . . . .	10
init_req_bbcl . . . . .	11
init_req_emol . . . . .	12
iteracion_emol . . . . .	12
limpieza_notas . . . . .	13
parserFuentes . . . . .	14
tabla_frecuencia_palabras . . . . .	14
word_cloud . . . . .	15

<b>Index</b>	<b>17</b>
--------------	-----------

---

agregar\_datos\_unicos *Agregar datos unicos a una tabla MySQL*

---

## Description

Esta funcion agrega datos a una tabla MySQL utilizando endpoints que esperan datos en formato JSON.

## Usage

```
agregar_datos_unicos(data)
```

## Arguments

data                    Un data frame con los datos a insertar.

## Value

No retorna ningun valor.

## Examples

```
## Not run:
# Agregar datos unicos
noticias <- extraer_noticias_max_res("tesla", max_results=10, fuentes="bbcl", subir_a_bd = FALSE)
agregar_datos_unicos(noticias)

## End(Not run)
```

---

extraccion\_parrafos *Extraer parrafos de una columna de texto*

---

**Description**

Esta funcion procesa una columna de texto en un dataframe y extrae los parrafos que coinciden con los sinonimos proporcionados.

**Usage**

```
extraccion_parrafos(datos, sinonimos = c())
```

**Arguments**

`datos` Data frame que contiene los datos de entrada con la columna "contenido".  
`sinonimos` Vector de sinonimos que se incluiran en la busqueda.

**Value**

Data frame con una columna adicional 'parrafos\_filtrados' que contiene los parrafos extraidos como listas.

**Examples**

```
datos <- extraer_noticias_max_res("inteligencia artificial", max_results = 140, subir_a_bd = FALSE)
datos <- extraccion_parrafos(datos, sinonimos = c("IA", "AI"))
```

---

extraer\_noticias\_fecha

*Extraccion de noticias de medios chilenos por rango de fechas*

---

**Description**

Esta funcion permite realizar una extraccion automatizada de noticias de BioBio o Los medios de Emol, utilizando un rango de fechas.

**Usage**

```
extraer_noticias_fecha(  
  search_query,  
  fecha_inicio,  
  fecha_fin,  
  subir_a_bd = TRUE,  
  fuentes = "todas"  
)
```

**Arguments**

search_query	Una frase de búsqueda (obligatoria).
fecha_inicio	Fecha de inicio del rango de búsqueda en formato "YYYY-MM-DD" (obligatoria).
fecha_fin	Fecha de fin del rango de búsqueda en formato "YYYY-MM-DD" (obligatoria).
subir_a_bd	por defecto TRUE, FALSE para test y cosas por el estilo (opcional).
fuentes	es un string con las fuentes a extraer. Puede ser bbcl o las de emol.

**Value**

Un dataframe con las noticias extraídas.

**Examples**

```
## Not run:
noticias <- extraer_noticias_fecha("delincuencia", "2025-04-25",
"2025-04-28", subir_a_bd = FALSE, fuentes="bbcl")

## End(Not run)
```

---

```
extraer_noticias_fecha_bbcl
```

*Extraccion de noticias de BioBio.cl por rango de fechas*

---

**Description**

Esta funcion permite realizar una extraccion automatizada de noticias de BioBio.cl utilizando un rango de fechas.

**Usage**

```
extraer_noticias_fecha_bbcl(search_query, fecha_inicio, fecha_fin)
```

**Arguments**

search_query	Una frase de búsqueda (obligatoria).
fecha_inicio	Fecha de inicio del rango de búsqueda en formato "YYYY-MM-DD" (obligatoria).
fecha_fin	Fecha de fin del rango de búsqueda en formato "YYYY-MM-DD" (obligatoria).

**Value**

Un dataframe con las noticias extraídas.

## Examples

```
## Not run:
noticias <- extraer_noticias_fecha_bbcl("inteligencia artificial", "2025-01-01",
"2025-02-24")

## End(Not run)
```

---

extraer\_noticias\_fecha\_emol

*Extraccion de noticias de emol.com por rango de fechas*

---

## Description

Esta funcion permite realizar una extraccion automatizada de noticias de emol.com utilizando un rango de fechas.

## Usage

```
extraer_noticias_fecha_emol(search_query, fecha_inicio, fecha_fin, fuente)
```

## Arguments

search_query	Una frase de busqueda (obligatoria).
fecha_inicio	Fecha de inicio del rango de busqueda en formato "YYYY-MM-DD" (obligatoria).
fecha_fin	Fecha de fin del rango de busqueda en formato "YYYY-MM-DD" (obligatoria).
fuelle	Fuente de emol para iterar (obligatoria).

## Value

Un dataframe con las noticias extraidas.

## Examples

```
## Not run:
noticias <- extraer_noticias_fecha_emol("inteligencia artificial", "2025-01-01",
"2025-02-24", fuente="emol")

## End(Not run)
```

extraer\_noticias\_max\_res

*Extraccion de noticias de medios chilenos por cantidad maxima de resultados*

---

## Description

Esta funcion permite realizar una extraccion automatizada de noticias de BioBio y fuentes de El Mercurio.

## Usage

```
extraer_noticias_max_res(  
  search_query,  
  max_results = NULL,  
  subir_a_bd = TRUE,  
  fuentes = "todas"  
)
```

## Arguments

search_query	Una frase de busqueda (obligatoria).
max_results	Numero maximo de resultados a extraer (opcional, por defecto todos).
subir_a_bd	por defecto TRUE, FALSE para test y cosas por el estilo (opcional).
fuentes	por defecto marca todas las fuentes, pero se puede elegir una o varias de las disponibles en el README. (opcional)

## Details

Es importante mencionar que si tiene mas de una fuente seleccionada, la cantidad maxima de resultados se aplicara para cada una de las fuentes, es decir, si pones max\_results = 10 y tienes fuentes = "emol,guioteca,bbcl" tendras como maximo 30 resultados.

## Value

Un dataframe con las noticias extraidas.

## Examples

```
## Not run:  
noticias <- extraer_noticias_max_res("inteligencia artificial",  
max_results = 20, fuentes="bbcl, emol", subir_a_bd = FALSE)  
  
## End(Not run)
```

---

`extraer_noticias_max_res_bbcl`*Extraccion de noticias de BioBio.cl por cantidad maxima de resultados*

---

**Description**

Esta funcion permite realizar una extraccion automatizada de noticias de BioBio.cl entregando como parametro una cantidad maxima de resultados.

**Usage**

```
extraer_noticias_max_res_bbcl(search_query, max_results = NULL)
```

**Arguments**

`search_query` Una frase de busqueda (obligatoria).  
`max_results` Cantidad maxima de resultados (opcional).

**Value**

Un dataframe con las noticias extraidas.

**Examples**

```
## Not run:  
noticias <- extraer_noticias_fecha_bbcl("inteligencia artificial", "2025-01-01",  
"2025-02-24")  
  
## End(Not run)
```

---

`extraer_noticias_max_res_emol`*Extraccion de noticias de Emol.com*

---

**Description**

Esta funcion permite extraer noticias de las fuentes de Emol, tanto de las noticias no pagas de emol, como de quioteca y los medios regionales de El Mercurio

**Usage**

```
extraer_noticias_max_res_emol(search_query, max_results = NULL, fuente)
```

**Arguments**

search_query	Una frase de busqueda (obligatoria).
max_results	Numero maximo de resultados a extraer (opcional, por defecto todos).
fuentes	Fuente de emol para iterar (obligatoria).

**Value**

Un dataframe con las noticias extraidas.

**Examples**

```
## Not run:
noticias <- extraer_noticias_max_res_emol("inteligencia artificial", "2025-01-01",
"2025-02-24", fuentes="mediosregionales")

## End(Not run)
```

---

grafico\_comparacion\_medios

*Grafico de comparacion de medios por periodo (Interactivo)*

---

**Description**

Esta funcion genera un grafico interactivo que compara la cantidad de publicaciones entre diferentes medios de medios, agrupadas por dia o por mes, con opcion de tema dark o light.

**Usage**

```
grafico_comparacion_medios(
  datos,
  titulo,
  fecha_inicio = NULL,
  fecha_fin = NULL,
  medios = NULL,
  agrupar_por = "day",
  tema = "light",
  tipo_grafico = "lineas"
)
```

**Arguments**

datos	Data frame con los datos procesados, que debe incluir las columnas 'fecha' y 'medio'.
titulo	Texto que aparecera en el titulo del grafico.
fecha_inicio	Fecha de inicio para la construccion del grafico en formato YYYY-MM-DD (opcional).

fecha_fin	Fecha de finalizacion para la construccion del grafico en formato YYYY-MM-DD (opcional).
medios	Vector de strings con las medios a comparar. Si es NULL, usa todas las medios disponibles.
agrupar_por	Cadena de texto que especifica el periodo de agrupacion. Valores validos son "day" (por defecto) o "month".
tema	Tema del grafico. Valores validos son "light" (por defecto) o "dark".
tipo_grafico	Tipo de visualizacion. Valores validos son "lineas" (por defecto) o "barras".

**Value**

Un grafico plotly interactivo que muestra la comparacion de publicaciones por medio y periodo.

**Examples**

```
## Not run:
# Comparar todas las medios por mes
datos <- extraer_noticias_fecha("delincuencia", "2024-01-01", "2025-01-01", subir_a_bd = FALSE)
grafico_comparacion_medios(datos, titulo = "Cobertura de Delincuencia por Medio",
                           agrupar_por = "month", tema = "dark")

# Comparar medios especificas por dia
grafico_comparacion_medios(datos, titulo = "Comparacion BBC1 vs emol",
                           medios = c("bbc1", "emol"),
                           fecha_inicio = "2024-06-01", fecha_fin = "2024-06-30",
                           agrupar_por = "day", tipo_grafico = "barras")

## End(Not run)
```

---

grafico\_notas\_fecha *Grafico de notas por periodo (interactivo)*

---

**Description**

Esta funcion genera un grafico interactivo que muestra la cantidad de publicaciones agrupadas por día o por mes, con opcion de tema dark o light.

**Usage**

```
grafico_notas_fecha(
  datos,
  titulo,
  fecha_inicio = NULL,
  fecha_fin = NULL,
  agrupar_por = "day",
  tema = "light"
)
```

### Arguments

datos	Data frame con los datos procesados, que debe incluir la columna 'fecha' en formato YYYY-MM-DD.
titulo	Texto que aparecera en el titulo del grafico.
fecha_inicio	Fecha de inicio para la construccion del grafico en formato YYYY-MM-DD (opcional).
fecha_fin	Fecha de finalizacion para la construccion del grafico en formato YYYY-MM-DD (opcional).
agrupar_por	Cadena de texto que especifica el periodo de agrupacion. Valores validos son "day" (por defecto) o "month".
tema	Tema del grafico. Valores validos son "light" (por defecto) o "dark".

### Value

Un grafico plotly interactivo que muestra la cantidad de publicaciones por el periodo seleccionado.

### Examples

```
## Not run:
# Ejemplo con tema dark, agrupando por mes
datos <- extraer_noticias_fecha("cambio climatico", "2024-01-01", "2025-01-01", subir_a_bd = FALSE)
grafico_notas_fecha(datos, titulo = "Cambio Climatico (por mes)",
                    agrupar_por = "month", tema = "dark")

# Ejemplo con tema light, agrupando por dia
grafico_notas_fecha(datos, titulo = "Cambio Climatico (por dia)",
                    fecha_inicio = "2024-01-01", fecha_fin = "2024-03-31",
                    tema = "light")

## End(Not run)
```

---

grafico\_notas\_por\_mes *Grafico de notas por mes*

---

### Description

Esta funcion genera un grafico de linea que muestra la cantidad de publicaciones agrupadas por mes.

### Usage

```
grafico_notas_por_mes(datos, titulo, fecha_inicio = NULL, fecha_fin = NULL)
```

**Arguments**

datos	Data frame con los datos procesados, que debe incluir la columna 'fecha' en formato YYYY-MM-DD.
titulo	Texto que aparecera en el titulo del grafico.
fecha_inicio	Fecha de inicio para la construccion del grafico en formato YYYY-MM-DD (opcional).
fecha_fin	Fecha de finalizacion para la construccion del grafico en formato YYYY-MM-DD (opcional).

**Value**

Un grafico ggplot2 que muestra la cantidad de publicaciones por mes.

**Examples**

```
## Not run:
datos <- extraer_noticias_fecha("cambio climatico", "2024-01-01", "2025-01-01", subir_a_bd = FALSE)
grafico_notas_por_mes(datos, titulo = "Cambio Climatico",
fecha_inicio = "2024-01-01", fecha_fin = "2024-06-06")

## End(Not run)
```

---

init_req_bbcl	<i>Inicializa una solicitud a BioBio.cl y retorna el primer caso de busqueda</i>
---------------	--

---

**Description**

Esta funcion permite realizar una consulta inicial a BioBio.cl utilizando una frase de busqueda.

**Usage**

```
init_req_bbcl(search_query)
```

**Arguments**

search\_query Una frase de busqueda (obligatoria).

**Value**

Un dataframe con el primer caso de la busqueda.

**Examples**

```
## Not run:
primer_caso <- init_req_bbcl("inteligencia artificial")

## End(Not run)
```

---

init_req_emol	<i>Inicializa una solicitud a emol.com y retorna el primer caso de busqueda</i>
---------------	---

---

**Description**

Esta funcion permite realizar una consulta inicial a emol.com utilizando una frase de busqueda.

**Usage**

```
init_req_emol(search_query, fuentes = "emol-todas")
```

**Arguments**

search_query	Una frase de busqueda (obligatoria).
fuentes	Un string donde se ponen las fuentes de emol a consultar

**Value**

Un dataframe con el primer caso de la busqueda.

**Examples**

```
## Not run:
primer_caso <- init_req_emol("Boric", fuentes="emol")

## End(Not run)
```

---

iteracion_emol	<i>Inicializa una solicitud a emol.com y retorna maximo 10 noticias</i>
----------------	---

---

**Description**

Esta funcion auxiliar llama a emol.com utilizando una frase de busqueda. Entrega como maximo 10 resultados. Se debe llamar desde otras funciones solo con una fuente a la vez, es decir, sin llamar a emol-todas.

**Usage**

```
iteracion_emol(search_query, page = 0, fuentes = "emol-todas")
```

**Arguments**

search_query	Una frase de busqueda (obligatoria).
page	La pagina de busqueda para iterar, es un int
fuentes	Es un string que deberia tener solo fuentes de emol posibles separadas por comas.

**Value**

Un dataframe con el caso de la búsqueda, incluyendo solo columnas específicas.

**Examples**

```
## Not run:  
primer_caso <- iteracion_emol("Boric", fuentes="emol-todas")  
  
## End(Not run)
```

---

limpieza_notas	<i>Funcion para limpiar notas de contenido HTML</i>
----------------	---

---

**Description**

Esta funcion permite limpiar por completo las notas eliminando codigos y secciones irrelevantes. Verifica que el input sea un data frame con una columna llamada 'contenido'.

**Usage**

```
limpieza_notas(datos, sinonimos = c())
```

**Arguments**

datos	Data frame donde estan almacenadas las notas
sinonimos	Un vector de character

**Value**

Un dataframe con el contenido limpio en la columna contenido\_limpio

**Examples**

```
datos <- extraer_noticias_max_res("inteligencia artificial", max_results= 150, subir_a_bd = FALSE)  
datos_proc <- limpieza_notas(datos, sinonimos = c("IA", "AI"))
```

---

parserFuentes	<i>Parser de Fuentes</i>
---------------	--------------------------

---

**Description**

Esta funcion toma un string que contiene nombres de fuentes separados por comas y devuelve una lista con cada fuente como un elemento separado, sin espacios en blanco adicionales.

**Usage**

```
parserFuentes(cadena)
```

**Arguments**

cadena	Un string que contiene nombres de fuentes separados por comas.
--------	--

**Value**

Una lista de strings, cada uno representando una fuente sin espacios en blanco adicionales.

**Examples**

```
parserFuentes("bbcl, emol, mediosregionales, ")
parserFuentes(" emol-todas, bbcl")
```

---

tabla_frecuencia_palabras	<i>Generar una tabla estilizada con las palabras mas frecuentes</i>
---------------------------	---

---

**Description**

Esta funcion procesa la columna 'contenido\_limpio' de un dataframe, tokeniza el texto, cuenta la frecuencia de cada palabra y genera una tabla con las palabras mas frecuentes.

**Usage**

```
tabla_frecuencia_palabras(datos, max_words, stop_words = NULL)
```

**Arguments**

datos	Data frame que contiene la columna 'contenido_limpio'.
max_words	Numero maximo de palabras que se mostraran en la tabla.
stop_words	Vector opcional de palabras que se deben excluir del conteo.

**Value**

Una tabla con las palabras mas frecuentes.

**Examples**

```
datos <- data.frame(  
  contenido_limpio = c(  
    "La ministra de Defensa Maya Fernandez enfrenta cuestionamientos  
    el presidente Gabriel Boric solicita transparencia en los procesos.  
    Renovacion Nacional pide la renuncia de Maya Fernandez debido a la polemica.  
    La transparencia es fundamental en la politica y la gestion publica"  
  ),  
  stringsAsFactors = FALSE  
)  
  
# Probar la funcion con el dataframe de ejemplo  
tabla_frecuencia_palabras(datos, max_words = 5, stop_words = c())
```

---

word\_cloud

*Funcion de nube de palabras*

---

**Description**

Esta funcion permite realizar una nube de palabras con las palabras más frecuentes del corpus de noticias.

**Usage**

```
word_cloud(datos, max_words, stop_words = NULL)
```

**Arguments**

datos	data frame que incluye la columna contenido_limpio.
max_words	Cantidad maxima de palabras que apareceran en la nube.
stop_words	Definir las palabras que seran ignoradas en la visualizacion. Debe ser un vector de caracteres.

**Value**

Una nube de palabras con las palabras mas frecuentes.

**Examples**

```
## Not run:
datos <- extraer_noticias_fecha("Boric",
"2025-03-01",
"2025-04-01",
fuentes="bbcl",
subir_a_bd = FALSE)
datos_proc <- limpieza_notas(datos)
word_cloud(datos_proc, max_words = 50, stop_words = c("es", "la"))

## End(Not run)
```

# Index

agregar\_datos\_unicos, [2](#)

extraccion\_parrafos, [3](#)

extraer\_noticias\_fecha, [3](#)

extraer\_noticias\_fecha\_bbcl, [4](#)

extraer\_noticias\_fecha\_emol, [5](#)

extraer\_noticias\_max\_res, [6](#)

extraer\_noticias\_max\_res\_bbcl, [7](#)

extraer\_noticias\_max\_res\_emol, [7](#)

grafico\_comparacion\_medios, [8](#)

grafico\_notas\_fecha, [9](#)

grafico\_notas\_por\_mes, [10](#)

init\_req\_bbcl, [11](#)

init\_req\_emol, [12](#)

iteracion\_emol, [12](#)

limpieza\_notas, [13](#)

parserFuentes, [14](#)

tabla\_frecuencia\_palabras, [14](#)

word\_cloud, [15](#)