

# Package ‘datamedios’

February 18, 2025

**Type** Package

**Title** Scraping Chilean Media

**Version** 1.1.0

**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, tidyverse, wordcloud2, tidytext, lubridate, rvest, stringr, xml2, purrr, DT, ggplot2

**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** 2025-02-18 10:10:10 UTC

## Contents

agregar_datos_unicos . . . . .	2
extraccion_parrafos . . . . .	3

extraer_noticias_fecha . . . . .	3
extraer_noticias_max_res . . . . .	4
grafico_notas_por_mes . . . . .	5
init_req_bbcl . . . . .	5
limpieza_notas . . . . .	6
tabla_frecuencia_palabras . . . . .	7
word_cloud . . . . .	8
<b>Index</b>	<b>9</b>

---

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

---

### Description

Esta funcion agrega datos a una tabla MySQL utilizando una API que espera 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

```
# Agregar datos unicos
noticias <- extraer_noticias_max_res("tesla", max_results=10, subir_a_bd = FALSE)
agregar_datos_unicos(noticias)
```

---

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 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(  
  search_query,  
  fecha_inicio,  
  fecha_fin,  
  subir_a_bd = TRUE  
)
```

**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).

**Value**

Un dataframe con las noticias extraídas.

**Examples**

```
noticias <- extraer_noticias_fecha("inteligencia artificial", "2025-01-01",  
"2025-02-24", subir_a_bd = FALSE)
```

---

```
extraer_noticias_max_res
```

*Extraccion de noticias de BioBio.cl*

---

**Description**

Esta funcion permite realizar una extraccion automatizada de noticias de BioBio.cl.

**Usage**

```
extraer_noticias_max_res(search_query, max_results = NULL, subir_a_bd = TRUE)
```

**Arguments**

search_query	Una frase de búsqueda (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).

**Value**

Un dataframe con las noticias extraídas.

**Examples**

```
noticias <- extraer_noticias_max_res("inteligencia artificial",  
max_results = 100, subir_a_bd = FALSE)
```

---

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

```
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")
```

---

init\_req\_bbcl *Inicializa una solicitud a BioBio.cl y retorna el primer caso de busqueda*

---

### 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 búsqueda (obligatoria).

**Value**

Un dataframe con el primer caso de la búsqueda.

**Examples**

```
primer_caso <- init_req_bbcl("inteligencia artificial")
```

---

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 y con la funcion extraccion\_parrafos ya operada.

sinonimos        Una lista 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 <- extraccion_parrafos(datos)
datos_proc <- limpieza_notas(datos, sinonimos = c("IA", "AI"))
```

---

`tabla_frecuencia_palabras`*Generar una tabla estilizada con las palabras mas frecuentes*

---

### 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

<code>datos</code>	Data frame que contiene la columna 'contenido_limpio'.
<code>max_words</code>	Numero maximo de palabras que se mostraran en la tabla.
<code>stop_words</code>	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**

<code>datos</code>	data frame que incluye la columna <code>contenido_limpio</code> .
<code>max_words</code>	Cantidad maxima de palabras que apareceran en la nube.
<code>stop_words</code>	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**

```
datos <- extraer_noticias_fecha("Monsalve", "2024-01-01", "2025-01-01", subir_a_bd = FALSE)
datos_proc <- limpieza_notas(datos)
word_cloud(datos_proc, max_words = 50, stop_words = c("es", "la"))
```



# Index

agregar\_datos\_unicos, 2

extraccion\_parrafos, 3

extraer\_noticias\_fecha, 3

extraer\_noticias\_max\_res, 4

grafico\_notas\_por\_mes, 5

init\_req\_bbcl, 5

limpieza\_notas, 6

tabla\_frecuencia\_palabras, 7

word\_cloud, 8