

# Package ‘inegiR’

February 2, 2016

**Type** Package

**Title** Integrate INEGI’s (Mexican Stats Office) API with R

**Version** 1.2.0

**Date** 2015-10-19

**Author** Eduardo Flores

**Maintainer** Eduardo Flores <eduardo@enelmargen.org>

**Description** Provides functions to download and parse information from INEGI  
(Official Mexican statistics agency).

**Encoding** UTF-8

**License** CC0

**Imports** zoo,  
XML,  
plyr,  
jsonlite

**Language** es

**RoxygenNote** 5.0.1

## R topics documented:

inegiR-package	2
crecer	2
denu_e_grid	3
denu_e_inegi	4
denu_e_varios_stats	5
hacer_grid	6
inflacion_ciudades	7
inflacion_estudiantes	8
inflacion_general	8
inflacion_tot	9
ordenar_porconteo	10
series_actividad_industrial	10
series_balanza_comercial	11
series_balanza_pagos	12
series_crecimiento_regiones	12
series_exportaciones_pais	13
series_ITAE_estados	14

series_opiniones . . . . .	15
series_PIB_estados . . . . .	15
series_produccion_autos . . . . .	16
series_tipocambio . . . . .	17
serie_inegi . . . . .	17
serie_inegi_json . . . . .	18
tasa_comercio . . . . .	19
tasa_confianza . . . . .	20
tasa_desempleo . . . . .	21
tasa_IGAE . . . . .	21
tasa_PIB . . . . .	22
tasa_sectoresYoY . . . . .	23
ultimos . . . . .	23
YoY . . . . .	24

<b>Index</b>	<b>26</b>
--------------	-----------

---

inegiR-package	<i>Funciones para API de INEGI</i>
----------------	------------------------------------

---

## Description

Funciones para obtener, interactuar y gráficar datos del Instituto Nacional de Estadística y Geografía (México) via el servicio web. La implementación se basa en wrappers de xml, zoo y plyr.

## Details

Package: inegiR  
 Type: Package  
 Version: 1.0  
 Date: 2015-06-19

Para obtener una serie, la función más primitiva es `serie_inegi()`

## Author(s)

Eduardo Flores <eduardo@enelmargen.org>

## References

Se puede encontrar más referencias sobre el API del INEGI, incluyendo generación de un token personal y rutas a más indicadores aquí: <http://www.inegi.org.mx/desarrolladores/indicadores/apiindicadores.aspx>

---

crecer	<i>Crece una serie por tasas</i>
--------	----------------------------------

---

## Description

Al especificar un dato inicial, "crece" una serie de datos usando un vector de tasas de crecimiento. La tasa se hace de periodo en periodo.

**Usage**

```
crecer(tasas, comienzo)
```

**Arguments**

tasas	vector con tasas de crecimiento
comienzo	número inicial

**Value**

Vector numerico

**Author(s)**

Eduardo Flores

**See Also**

series\_crecimiento\_regiones

**Examples**

```
tasas_crecimiento<-c(1.10,1.20,1.05,1.02,1.10)

# Crecer por esas tasas (en cada periodo) el 100:
Resultados<-crecer(tasas = tasas_crecimiento, comienzo = 100)
```

---

denuc\_grid

*Obtiene establecimientos del DENUC en una area mayor a 5kms*

---

**Description**

Regresa data.frame de datos de establecimientos registrados en el DENUC en un grid con dos o mas areas de 5kms (el limite de la llamada a INEGI). Llama a hacer\_grid, ambas posible gracias a Arturo Cardenas <https://github.com/arturocm>.

**Usage**

```
denuc_grid(lat1, lat2, lon1, lon2, token, metros = 5000, keyword = "todos",
  espacio_lat = 0.07, espacio_lon = 0.07, unicos = TRUE)
```

**Arguments**

lat1	Esquina 1 de cuadro o área en latitud.
lat2	Esquina 2 de cuadro o área en latitud.
lon1	Esquina 1 de cuadro o área en longitud.
lon2	Esquina 2 de cuadro o área en longitud.
token	Token emitida por INEGI para acceder a API

metros	Distancia en metros a la redonda para buscar establecimientos. Default = 5000, que es el máximo permitido por INEGI. Considera que si no cambias el espacio entre mediciones (en los otros parámetros), no vas a dibujar un cuadro totalmente cubierto por círculos.
keyword	Palabra clave de establecimiento para buscar, a pasar a denue_inegi. Por default busca todos.
espacio_lat	Espacio entre coordenadas, en latitud, por default = 0.07 grados.
espacio_lon	Espacio entre coordenadas, en longitud, por default = 0.07 grados.

### Details

Se hace un loop por cada par de coordenadas, que se sobrelapan en círculos alrededor de un cuadro proporcionado por las cuatro esquinas de los parámetros. La función `hacer_grid`, usa máximos y mínimos de latitud y longitud para asignar pares, por lo que debes considerar con cuidado el área a mapear.

### Value

Data.frame

### Author(s)

Arturo Cardenas

### Examples

```
## Not run:
token<-"webservice_token"
latitud1 <- "25.669194"
latitud2 <- "25.169194"
longitud1 <- "-100.30990"
longitud2 <- "-101.20102"
Negocios <- denue_grid(latitud1, latitud2, longitud1, longitud2, token)

## End(Not run)
```

---

denue\_inegi

*Obtiene establecimientos del DENUE*


---

### Description

Regresa data.frame de datos de establecimientos registrados en el DENUE en zona aledaña a las coordenadas. Es una de las funciones primitivas del paquete.

### Usage

```
denue_inegi(latitud, longitud, token, metros = 250, keyword = "todos")
```

**Arguments**

latitud	Vector en caracter de latitud (en decimal) de lugar
longitud	Vector en caracter de longitud (en decimal) de lugar
token	Token emitida por INEGI para acceder a API
metros	Distancia en metros a la redonda para buscar establecimientos. Default = 250
keyword	Palabra clave de establecimiento para buscar. Por default busca todos.

**Value**

Data.frame

**Author(s)**

Eduardo Flores

**Examples**

```
#Traer todos los establecimientos a 1 km de la macro plaza en Monterrey
## Not run:
token<-"webservice_token"
latitud<-"25.669194"
longitud<-"-100.30990"
Negocios <- denue_inegi(latitud, longitud, token, metros = 1000)

## End(Not run)
```

---

denue_varios_stats	<i>Obtiene estadísticas de coordenada</i>
--------------------	---

---

**Description**

Regresa Data.Frame con estadísticas básicas sobre los establecimientos encontrados a la redonda de coordenada, utilizando denue\_inegi(). Trae una función de loop integrada, para que pueda regresar indicadores de muchas coordenadas, utilizando un data.frame.

**Usage**

```
denue_varios_stats(data, col_lat, col_long, token, metros = 250,
  keyword = "todos")
```

**Arguments**

data	Data.frame dónde se encuentran las dos columnas de coordenadas
col_lat	número de columna de "data" en dónde se encuentra la latitud
col_long	número de columna de "data" en dónde se encuentra la longitud
token	Token emitida por INEGI para acceder a API
metros	Distancia en metros a la redonda para buscar establecimientos. Default = 250
keyword	Palabra clave de establecimiento para buscar. Por default busca todos.

**Value**

Data.frame

**Author(s)**

Eduardo Flores

**Examples**

```
#indicadores de 2 lugares
## Not run:
token<-"webservice_token"
df<-as.data.frame(latitud = c(25.669194, 25.121194),
                    longitud = c(-100.30990, -99.81923))
stats<-denue_varios_stats(data = df,
                           col_lat = 1,
                           col_long = 2,
                           metros = 500)

## End(Not run)
```

hacer\_grid

*Proporciona un set de coordenadas***Description**

Regresa un set de coordenadas que juntas se sobrelapan para crear un cuadro mayor a 5 kilómetros.  
Posible gracias a Arturo Cardenas <https://github.com/arturocm>.

**Usage**

```
hacer_grid(lat1, lat2, lon1, lon2, espacio_lat = 0.07, espacio_lon = 0.07)
```

**Arguments**

lat1	Esquina 1 de cuadro o área en latitud.
lat2	Esquina 2 de cuadro o área en latitud.
lon1	Esquina 1 de cuadro o área en longitud.
lon2	Esquina 2 de cuadro o área en longitud.
espacio_lat	Espacio entre coordenadas, en latitud, por default = 0.07 grados.
espacio_lon	Espacio entre coordenadas, en longitud, por default = 0.07 grados.

**Value**

Data.frame

**Author(s)**

Arturo Cardenas

**See Also**

denu\_e\_grid

**Examples**

```
## Not run:
token<-"webservice_token"
latitud1 <- "25.669194"
latitud2 <- "25.169194"
longitud1 <- "-100.30990"
longitud2 <- "-101.20102"
varias_coordenadas <- hacer_grid(latitud1, latitud2, longitud1, longitud2)

## End(Not run)
```

---

inflacion_ciudades	<i>Obtener inflacion por Ciudad</i>
--------------------	-------------------------------------

---

**Description**

Obtiene la tasa de inflación mensual por ciudad. Es un wrapper de las funciones `serie_inegi()` y `YoY()`.

**Usage**

```
inflacion_ciudades(token)
```

**Arguments**

token	token personal emitido por el INEGI para acceder al API.
-------	--

**Value**

Data.frame

**Author(s)**

Eduardo Flores

**Examples**

```
## Not run:
token<-"webservice_token"
InflacionCiudades<-inflacion_ciudades(token)

## End(Not run)
```

---

inflacion\_estudiantes    *Obtener tasa de inflacion de Estudiantes*

---

**Description**

Obtiene tasa de inflación de estudiantes, inter anual en porcentaje. Es un wrapper de las funciones Serie\_Inegi() y YoY(). La metodología del índice se puede encontrar aquí: <http://enelmargen.org/eem/ipe/> Es un wrapper de las funciones serie\_inegi() y YoY().

**Usage**

```
inflacion_estudiantes(token)
```

**Arguments**

token                      token persona emitido por el INEGI para acceder al API.

**Value**

Data.frame

**Author(s)**

Eduardo Flores

**Examples**

```
## Not run:
token<-"webservice_token"
InflacionEstudiantes<-inflacion_estudiantes(token)

## End(Not run)
```

---

inflacion\_general            *Obtener tasa de inflacion*

---

**Description**

Obtiene tasa de inflación inter anual en porcentaje. La inflación se define como el cambio porcentual en el INPC. Es un wrapper de las funciones serie\_inegi() y YoY().

**Usage**

```
inflacion_general(token)
```

**Arguments**

token                      token persona emitido por el INEGI para acceder al API de indicadores.



**Value**

Data.frame

**Author(s)**

Eduardo Flores

**Examples**

```
## Not run:  
token<-"webservice_token"  
Inflacion<-inflacion_general(token)  
  
## End(Not run)
```

---

inflacion\_tot

*Obtener terminos de intercambio*

---

**Description**

Obtiene la razón de términos de intercambio para México (ToT). Es un wrapper de las funciones `serie_inegi()` y `YoY()`. La razón se define como el índice de precios de exportaciones entre el índice de precios de importaciones. Es un wrapper de las funciones `serie_inegi()` y `YoY()`.

**Usage**

```
inflacion_tot(token)
```

**Arguments**

token                      token personal emitido por el INEGI para acceder al API.

**Value**

Data.frame

**Author(s)**

Eduardo Flores

**Examples**

```
## Not run:  
token<-"webservice_token"  
TerminosIntercambio<-inflacion_tot(token)  
  
## End(Not run)
```

---

ordenar_porconteo	<i>Ordenar por conteo de factores</i>
-------------------	---------------------------------------

---

**Description**

Wrapper para ordenar rapidamente de mayor a menor por grupos un data.frame.

**Usage**

```
ordenar_porconteo(df, col)
```

**Arguments**

df	Data.frame a condensar
col	Columna con factores. Se pone sin parentesis.

**Value**

Data.frame

**Author(s)**

Eduardo Flores

**See Also**

denue\_varios\_stats

**Examples**

```
df<-data.frame(factor=c("A","A","B","C","C","D","A","A"),otros=c(1,3,2,4,5,1,2,7))

#Ordenar, de mayor a menor, por conteo de factores
PorConteo<-ordenar_porconteo(df, factores)
```

---

series\_actividad\_industrial

*Obtener crecimientos de actividad industrial*

---

**Description**

Obtiene principales tasas de crecimiento YoY de componentes de Actividad Industrial (series originales): Construcción, Manufacturas, Minería y Generación de Luz y Agua. Aun y cuando son las mismas series reportadas en el IGAE unas semanas después, estas pueden sufrir ajustes (ver documentación del INEGI así como número de indicador mediante metadata = TRUE). Es un wrapper de las funciones serie\_inegi() y YoY().

**Usage**

```
series_actividad_industrial(token)
```

**Arguments**

token                      token personal emitido por el INEGI para acceder al API.

**Value**

Data.frame

**Author(s)**

Eduardo Flores

**Examples**

```
## Not run:
token<-"webservice_token"
ActividadIndustrial<-series_actividad_industrial(token)

## End(Not run)
```

---

series\_balanza\_comercial

*Obtener balanza comercial*

---

**Description**

Obtiene exportaciones, importaciones y balance de los dos en un mismo data.frame por mes. Todos los productos y todos los países. Es un wrapper de las funciones `serie_inegi()` y `YoY()`.

**Usage**

```
series_balanza_comercial(token)
```

**Arguments**

token                      token personal emitido por el INEGI para acceder al API.

**Value**

Data.frame

**Author(s)**

Eduardo Flores

**Examples**

```
## Not run:
token<-"webservice_token"
ComercioExterior<-series_balanza_comercial(token)

## End(Not run)
```

---

series\_balanza\_pagos    *Obtener Balanza de Pagos*

---

### Description

Obtiene principales componentes de la Balanza de Pagos: 2 de la Cuenta Corriente, 3 de la Cuenta Financiera y sus 2 resultados. Es un wrapper de las funciones `serie_inegi()` y `YoY()`.

### Usage

```
series_balanza_pagos(token)
```

### Arguments

token                      token personal emitido por el INEGI para acceder al API.

### Value

Data.frame

### Author(s)

Eduardo Flores

### Examples

```
## Not run:
token<-"webservice_token"
BalanzadePagosMexico<-series_balanza_pagos(token)

## End(Not run)
```

---

series\_crecimiento\_regiones                      *Obtiene indicador de actividad por zona*

---

### Description

Obtiene índice de PIB por zona geográfica. Este es un wrapper de `serie_inegi()`, con una metodología propia de En El Margen, se puede consultar la misma aquí: <http://enelmargen.org/eem/regiones/>. El crecimiento del mismo es una estimación de la dinámica de crecimiento regional nacional, inspirado en <http://www.banxico.org.mx/publicaciones-y-discursos/publicaciones/informes-periodicos/reportes-sobre-las-economias-regionales/{1C8EFC32-C12C-8393-6C29-5AF0A7F456}.pdf>. Los agrupamientos regionales provienen del INEGI <http://www.inegi.org.mx/est/contenidos/proyectos/cn/itaee/default.aspx>

### Usage

```
series_crecimiento_regiones(token)
```

**Arguments**

token                      token personal emitido por el INEGI para acceder al API.

**Value**

Data.frame

**Author(s)**

Eduardo Flores

**See Also**

series\_PIB\_estados, crecer, series\_ITAE\_estados

**Examples**

```
## Not run:
token<-"webservice_token"
CrecimientoZonas<-series_crecimiento_regiones(token)

## End(Not run)
```

---

series\_exportaciones\_pais

*Obtener exportaciones por paises*

---

**Description**

Obtiene exportaciones de principales socios comerciales. Todos los productos y Estados Unidos, Canadá, China, CentroAmerica y América del Sur. Es un wrapper de las funciones `serie_inegi()` y `YoY()`.

**Usage**

```
series_exportaciones_pais(token)
```

**Arguments**

token                      token personal emitido por el INEGI para acceder al API.

**Value**

Data.frame

**Author(s)**

Eduardo Flores

**Examples**

```
## Not run:
token<-"webservice_token"
ExportacionesMx<-series_exportaciones_pais(token)

## End(Not run)
```

---

series_ITAE_estados	<i>Obtiene ITAE</i>
---------------------	---------------------

---

**Description**

Obtiene series originales de Indicador Trimestral de Actividad Económica (ITAE) por estado.

**Usage**

```
series_ITAE_estados(token)
```

**Arguments**

token	token personal emitido por el INEGI para acceder al API.
-------	--

**Value**

Data.frame

**Author(s)**

Eduardo Flores

**See Also**

series\_PIB\_estados, series\_crecimiento\_regiones

**Examples**

```
## Not run:
token<-"webservice_token"
ITAE<-series_ITAE_estados(token)

## End(Not run)
```

---

series_opiniones	<i>Obtener opiniones empresariales por sector</i>
------------------	---

---

**Description**

Obtiene principales componentes de encuestas de Opinión Empresarial del INEGI dividido en 3 sectores: Comercio, Manufacturas y Construcción. Es un wrapper de las funciones `serie_inegi()` y `YoY()`.

**Usage**

```
series_opiniones(token)
```

**Arguments**

token	token personal emitido por el INEGI para acceder al API.
-------	--

**Value**

Data.frame

**Author(s)**

Eduardo Flores

**Examples**

```
## Not run:  
token<-"webservice_token"  
OpinionMexicanos<-series_opiniones(token)  
  
## End(Not run)
```

---

series_PIB_estados	<i>Obtener PIB por Entidad Federativa</i>
--------------------	---

---

**Description**

Obtiene series originales del PIB a Precios 2008 por Entidad Federativa. Unidades: millones de pesos a precios de 2008. Es un wrapper de `serie_inegi()`.

**Usage**

```
series_PIB_estados(token)
```

**Arguments**

token	token personal emitido por el INEGI para acceder al API.
-------	--

**Value**

Data.frame

**Author(s)**

Eduardo Flores

**See Also**

series\_crecimiento\_regiones

**Examples**

```
## Not run:
token<-"webservice_token"
Estados<-series_PIB_estados(token)

## End(Not run)
```

---

series\_produccion\_autos

*Obtener Produccion de Autos*

---

**Description**

Obtiene producción automotriz en México y cambio porcentual anual. Es un wrapper de las funciones `serie_inegi()` y `YoY()`.

**Usage**

```
series_produccion_autos(token)
```

**Arguments**

token                      token personal emitido por el INEGI para acceder al API.

**Value**

Data.frame

**Author(s)**

Eduardo Flores

**Examples**

```
## Not run:
token<-"webservice_token"
ProduccionAutos<-series_produccion_autos(token)

## End(Not run)
```



---

series_tipocambio	<i>Obtener Tipo de Cambio Peso - USD</i>
-------------------	--

---

**Description**

Obtiene tipo de cambio interbancario (venta) histórico de pesos a dólares. Es un wrapper de las funciones `serie_inegi()` y `YoY()`.

**Usage**

```
series_tipocambio(token)
```

**Arguments**

token	token personal emitido por el INEGI para acceder al API.
-------	--

**Value**

Data.frame

**Author(s)**

Eduardo Flores

**Examples**

```
## Not run:
token<-"webservice_token"
USD<-series_tipocambio(token)

## End(Not run)
```

---

serie_inegi	<i>Obtiene serie de tiempo de INEGI</i>
-------------	---

---

**Description**

Regresa data.frame con la serie de tiempo escogida, al buscar en el webservice del INEGI y parsear via XML y ZOO. Si parametro Metadata=TRUE, regresa lista con indicadores meta y datos. Es una de las funciones primitivas del paquete.

**Usage**

```
serie_inegi(serie, token, metadata = FALSE, coercion = TRUE)
```

**Arguments**

serie	Vector en caracter de url de dirección. Este es un método directo (se requiere de URL en formato XML, con token)
token	token personal emitido por el INEGI para acceder al API.
metadata	Default = FALSE, si TRUE, parsea una lista con metadatos de serie.
coercionar	Por default (TRUE), los indicadores quincenales serán coercionados a mensuales. Aparecerán todas las observaciones pero en el mismo día del mes a pesar de estar en diferentes quincenas. Para usar días = FALSE.

**Value**

Dataframe o lista

**Note**

La instancia "?callback?", requerida por la documentación del INEGI para series JSON no es necesaria.

**Author(s)**

Eduardo Flores

**Examples**

```
## Not run:
#Serie de INPC General
token<-"webservice_token"
url <- "http://www3.inegi.org.mx/sistemas/api/indicadores/v1//Indicador/216064/00000/es/false/xml/"
Serie <- serie_inegi(url, token)

## End(Not run)
```

---

serie\_inegi\_json

*Obtiene serie de tiempo de INEGI en formato JSON*


---

**Description**

Regresa data.frame con la serie de tiempo escogida, al buscar en el webservice del INEGI y parsear via Jsonlite. Si parametro Metadata=TRUE, regresa lista con indicadores meta y datos.

**Usage**

```
serie_inegi_json(serie, token, metadata = FALSE, coercionar = TRUE)
```

**Arguments**

serie	Vector en caracter de url de dirección. Este es un método directo (se requiere de URL en formato XML, con token)
token	token personal emitido por el INEGI para acceder al API.
metadata	Default = FALSE, si TRUE, parsea una lista con metadatos de serie.
coercionar	Por default (TRUE), los indicadores quincenales serán coercionados a mensuales. Aparecerán todas las observaciones pero en el mismo día del mes a pesar de estar en diferentes quincenas. Para usar días = FALSE.

**Details**

Esta función se llama directamente en `serie_inegi()`, cuando el parametro "serie" termina en "json/".

**Value**

Dataframe o lista

**Note**

La instancia "?callback?" requerida por la documentación del INEGI no es necesaria.

**Author(s)**

Eduardo Flores

**See Also**

`serie_inegi`

**Examples**

```
## Not run:
#Serie de INPC General
token<-"webservice_token"
url <- "http://www3.inegi.org.mx/sistemas/api/indicadores/v1//Indicador/216064/00000/es/false/xml/"
Serie <- serie_inegi(url,token)

## End(Not run)
```

---

tasa\_comercio

*Obtener Tasa de Crecimiento de Comercio*

---

**Description**

Obtiene tasa de crecimiento del Comercio (Actividad Terciaria), por mes. Es un wrapper de las funciones `serie_inegi()` y `YoY()`.

**Usage**

```
tasa_comercio(token)
```

**Arguments**

token                      token personal emitido por el INEGI para acceder al API.

**Value**

Data.frame

**Author(s)**

Eduardo Flores

**Examples**

```
## Not run:
token<-"webservice_token"
Comercio<-tasa_comercio(token)

## End(Not run)
```

---

tasa\_confianza

*Obtener Confianza del Consumidor*

---

**Description**

Obtiene Tasas de Cambio de Confianza del Consumidor Devuelve tasas de serie desestacionalizada anual, desestacionalizada contra mes previo y serie original anual. Es un wrapper de las funciones `serie_inegi()` y `YoY()`.

**Usage**

```
tasa_confianza(token)
```

**Arguments**

token                      token personal emitido por el INEGI para acceder al API.

**Value**

Data.frame

**Author(s)**

Eduardo Flores

**Examples**

```
## Not run:
token<-"webservice_token"
ConfianzaEconomia<-tasa_confianza(token)

## End(Not run)
```

---

tasa_desempleo	<i>Obtener Desempleo Urbano</i>
----------------	---------------------------------

---

**Description**

Obtiene tasa de desocupación (serie unificada) urbana (agregado de 32 ciudades) Es un wrapper de las funciones `serie_inegi()` y `YoY()`.

**Usage**

```
tasa_desempleo(token)
```

**Arguments**

token	token personal emitido por el INEGI para acceder al API.
-------	--

**Value**

Data.frame

**Author(s)**

Eduardo Flores

**Examples**

```
## Not run:  
token<-"webservice_token"  
Desempleo<-tasa_desempleo(token)  
  
## End(Not run)
```

---

tasa_IGAE	<i>Obtener IGAE</i>
-----------	---------------------

---

**Description**

Obtiene Tasas de Crecimiento de Indicador Global de Actividad Económica Devuelve tasas de serie desestacionalizada anual, desestacionalizada contra mes previo y serie original anual.

**Usage**

```
tasa_IGAE(token)
```

**Arguments**

token	token personal emitido por el INEGI para acceder al API.
-------	--

**Value**

Data.frame

**Author(s)**

Eduardo Flores

**Examples**

```
## Not run:
token<-"webservice_token"
ActividadEconomica<-tasa_IGAE(token)

## End(Not run)
```

---

tasa\_PIB

*Obtener tasa de crecimiento del PIB*

---

**Description**

Obtiene tasa de crecimiento vs. mismo periodo de un año antes en porcentaje. Es un wrapper de las funciones `serie_inegi()` y `YoY()`.

**Usage**

```
tasa_PIB(token)
```

**Arguments**

token                      token persona emitido por el INEGI para acceder al API.

**Value**

Data.frame

**Note**

Ruta tematica BIE: Indicadores económicos de coyuntura ... Producto interno bruto trimestral, base 2008 ... Series originales ... Valores a precios de 2008 ... Producto interno bruto, a precios de mercado

**Author(s)**

Eduardo Flores

**Examples**

```
## Not run:
token<-"webservice_token"
CrecimientoMex<-tasa_PIB(token)

## End(Not run)
```

---

tasa_sectoresYoY	<i>Obtener cambios porcentuales por sector</i>
------------------	--

---

**Description**

Obtiene Tasas de Crecimiento de Indicador Global de Actividad Económica por subsector. Todas las tasas son con series originales. Cambio porcentual anual.

**Usage**

```
tasa_sectoresYoY(token)
```

**Arguments**

token	token personal emitido por el INEGI para acceder al API.
-------	--

**Value**

Data.frame

**Author(s)**

Eduardo Flores

**Examples**

```
## Not run:  
token<-"webservice_token"  
Sectores<-tasa_sectoresYoY(token)  
  
## End(Not run)
```

---

ultimos	<i>Traer n datos mas recientes</i>
---------	------------------------------------

---

**Description**

Wrapper para ordenar de mayor a menor serie y traer solamente últimos 13 periodos. Prefente para series mensuales.

**Usage**

```
ultimos(serie, col = "Fechas", n = 12)
```

**Arguments**

serie	serie en data.frame
col	Columna con fechas
n	cantidad de periodos a traer

**Value**

Data.frame

**Author(s)**

Eduardo Flores

**See Also**

denue\_varios\_stats

**Examples**

```
#Ver solamente ultimos 13 meses
## Not run:
Ultimos<-ultimos(Inflacion, n = 12)

## End(Not run)
```

YoY

*Calcular tasas de crecimiento***Description**

Calcula tasas de crecimiento de una serie.

**Usage**

```
YoY(serie, lapso, decimal = TRUE)
```

**Arguments**

serie	vector o serie de tiempo con datos numéricos
lapso	separaciones por año a contemplar (12 = datos mensuales, 4 = datos trimestrales)
decimal	¿Quieres que el resultado este en decimales? Default = TRUE. False obtiene el decimal x 100.

**Value**

Vector numerico

**Note**

La serie debe estar en orden ascendiente (Posición inicial es la más antigua). La función de `Serie_Inegi()` guarda en ese orden.

**Author(s)**

Eduardo Flores



**Examples**

```
#Calcular la inflación (Ver Inflacion_Inegi() para un método más directo)
## Not run:
token<-"webservice_token"
INPC<-serie_inegi(INPC, token)
Inflacion<-YoY(INPC$Valores,12)

## End(Not run)
```

# Index

## \*Topic **package**

inegiR-package, [2](#)

crecer, [2](#)

denue\_grid, [3](#)

denue\_inegi, [4](#)

denue\_varios\_stats, [5](#)

hacer\_grid, [6](#)

inegiR (inegiR-package), [2](#)

inegiR-package, [2](#)

inflacion\_ciudades, [7](#)

inflacion\_estudiantes, [8](#)

inflacion\_general, [8](#)

inflacion\_tot, [9](#)

ordenar\_porconteo, [10](#)

serie\_inegi, [17](#)

serie\_inegi\_json, [18](#)

series\_actividad\_industrial, [10](#)

series\_balanza\_comercial, [11](#)

series\_balanza\_pagos, [12](#)

series\_crecimiento\_regiones, [12](#)

series\_exportaciones\_pais, [13](#)

series\_ITAE\_estados, [14](#)

series\_opiniones, [15](#)

series\_PIB\_estados, [15](#)

series\_produccion\_autos, [16](#)

series\_tipocambio, [17](#)

tasa\_comercio, [19](#)

tasa\_confianza, [20](#)

tasa\_desempleo, [21](#)

tasa\_IGAE, [21](#)

tasa\_PIB, [22](#)

tasa\_sectoresYoY, [23](#)

ultimos, [23](#)

YoY, [24](#)