

# Package ‘OESolar’

June 15, 2020

**Type** Package

**Title** Diseño de sistemas fotovoltaicos y análisis de datos de energía descargados de Optimum

**Version** 0.1.1

**Author** Daniel G. Paniagua

**Maintainer** Daniel G. Paniagua <dgpaniagua1989@gmail.com>

**Description** Diseño de sistemas fotovoltaicos conectados a red a partir de datos de radiación solar global y parámetros del sistema, incluyendo funciones para descarga directa de datos de radiación solar de base de datos Nasa. El cálculo de los sistemas fotovoltaicos se realiza utilizando el package 'solaR'. Por otra parte, incluye funciones para descargar, de sistema Optimum, datos de generación y consumo de medidores del proyecto PRIER Armstrong, permitiendo el posterior análisis y cálculo de valores de PR diarios y mensuales.

**License** GPL-2

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.1.0

**Imports** dplyr, eplusr, lubridate, nasapower, readxl, reticulate, solaR, tools, xts, zoo, stats

## R topics documented:

agrupar_por . . . . .	2
descarga_optimum . . . . .	3
importar_epw . . . . .	4
importar_optimum . . . . .	5
info_huecos . . . . .	8
pr . . . . .	9
prod_energia . . . . .	10
rad_inclin . . . . .	15
rad_nasa . . . . .	17
rellena_huecos . . . . .	18

<b>Index</b>	<b>20</b>
--------------	-----------

---

`agrupar_por`*Agrupar datos de objeto xts según intervalo de tiempo dado*

---

**Description**

Agrupar datos de objeto xts según intervalo de tiempo dado. El intervalo de tiempo de los datos del xts de salida pueden ser cada una hora "HORA", cada un día "DIARIO" o cada un mes "MENSUAL". El intervalo de datos del xts de entrada debe ser menor que el de salida. Por ejemplo, se puede pasar de datos cada 15 minutos a datos cada una hora, o de datos cada un día a datos cada un mes.

**Usage**

```
agrupar_por(xts, int_salida = "MENSUAL")
```

**Arguments**

<code>xts</code>	xts de entrada. La columna de datos a agrupar debe llamarse "energia".
<code>int_salida</code>	Cadena de caracteres para especificar el nuevo intervalo de tiempo entre datos. Puede ser: "HORA", "DIARIO", "MENSUAL", según lo detallado en la descripción de la función

**Value**

xts con los datos agrupados según el nuevo intervalo de tiempo, con una sola columna llamada "energia".

**Author(s)**

Daniel G. Paniagua

**Examples**

```
# #Importa datos de medidor cada 15 minutos
# datos_15min <- importar_optimum(usuario="usuario_valido", password="password_valido",
#                               medidores="CIR0141449301", fecha_inicio = "2019-03-01",
#                               fecha_fin = "2020-03-15", int_entrada = "15MIN", consumo = TRUE)
# #Agrupar los datos cada 1 mes
# datos_mes <- agrupar_por(datos_15min)
```

---

descarga_optimum	<i>Descarga datos de medidores de energía de Optimum</i>
------------------	--

---

## Description

Descarga datos de medidores de energía de Optimum (Armstrong) y los devuelve como data.frame.

## Usage

```
descarga_optimum(
  usuario,
  password,
  medidores = "GEN1HORA",
  fecha_inicio = "2019-01-01",
  fecha_fin = "2019-01-02",
  channel = "Active Import Inc - 0"
)
```

## Arguments

usuario	Cadena de caracteres. Nombre de usuario para login en Optimum.
password	Cadena de caracteres. Contraseña para login en Optimum.
medidores	<p>Cadena de caracteres. Se puede ingresar el nombre de alguno de los grupos predefinidos que se detallan a continuación. En caso de no querer descargar ninguno de estos grupos predefinidos, se puede detallar una cadena con el/los número/s de medidor/es con una coma delante.</p> <p>Por ejemplo: ',CIR0141437453,CIR0141449390,CIR0141449501'.</p> <p>Los grupos predefinidos son:</p> <ul style="list-style-type: none"> <li>• medidores="TODOS": Todos los medidores Circutor de Optimum. Son 263 medidores. Todos se descargan del canal "Active Import Inc - 0", excepto la planta de piso y un medidor de generación cada una hora (ver detalle en siguientes ítems).</li> <li>• medidores="GEN15MIN": Corresponde a 6 medidores de generación con datos cada 15 minutos. Todos en canal "Active Import Inc - 0".</li> <li>• medidores="GEN1HORA": Corresponde a 5 medidores de generación con datos cada 1 hora con datos en canal "Active Import Inc - 0" y a uno (CIR0141668286) con datos en canal 'Active Export Inc - 0'.</li> <li>• medidores="DEM15MIN": Corresponde a 230 medidores de consumo con datos cada 15 minutos. Todos los datos en se descargan de canal "Active Import Inc - 0" (hasta actual estado de depuración).</li> <li>• medidores="DEM1HORA": Corresponde a 20 medidores de consumo con datos cada 1 hora. Todos los datos en se descargan de canal "Active Import Inc - 0" (hasta actual estado de depuración).</li> <li>• medidores="PLANTAPISO": Corresponde a datos de la planta de piso, los datos están en canal "Sum Li Active Power + (QI+QIV) (time integral 5) - 0".</li> </ul>

fecha_inicio	Cadena de caracteres. Fecha de inicio de los datos a descargar, en formato "YYYY-MM-DD".
fecha_fin	Cadena de caracteres. Fecha final de los datos a descargar, en formato "YYYY-MM-DD".
channel	Cadena de caracteres. Canal del medidor que contiene los datos a descargar. Los más utilizados son: "Active Import Inc - 0" (la mayoría está en este canal), 'Active Export Inc - 0', "Sum Li Active Power + (QI+QIV) (time integral 5) - 0" (solo planta de piso). Ver detalle en descripción de parámetro "medidores".

**Value**

data.frame con los datos descargados, con la misma disposición que la tabla descargada desde la web de Optimum.

**Author(s)**

Daniel G. Paniagua

**Examples**

```
# #Descarga datos de 230 medidores con datos cada 15 minutos, por el período de 13 meses.
# #En la prueba realizada se descargaron 7.841.964 datos en 973 segundos (16min 12seg aprox).
# #Se deben ingresar usuario y password validos (no incluidos en el ejemplo).
#
# datos_dem_190 <- descarga_optimum(usuario="usuario valido", password="password valido",
#                                   medidores="DEM15MIN", fecha_inicio = "2019-03-01",
#                                   fecha_fin = "2020-04-30")
#
# #Descarga datos de 1 medidor cada una hora por un mes.
# datos_1 <- descarga_optimum(usuario="usuario valido", password="password valido",
#                              medidores="CIR0141449301", fecha_inicio = "2019-03-01",
#                              fecha_fin = "2019-03-31")
```

---

importar_epw	<i>Importa datos de radiación y temperatura de archivo meteorológico epw</i>
--------------	--

---

**Description**

Importa datos de radiación solar global en el plano horizontal y de temperatura de un archivo epw.

**Usage**

```
importar_epw(rutaepw)
```

**Arguments**

rutaepw	Cadena de caracteres con ruta del archivo epw.
---------	--

## Details

La función debe asignar un año a los datos, ya que en un archivo epw los datos de los distintos meses generalmente corresponden a distintos años no consecutivos. El año es asignado por el método "data()" de la clase Epw (ver [Epw](#)), que tiene en cuenta no generar errores por años bisiestos. Si bien el método permite asignar el año, se prefiere no hacerlo para no generar este tipo de errores. Puede suceder que quede un dato correspondiente al año siguiente, Esto es porque la hora del dato siempre corresponde a la correspondiente al fin del intervalo de tiempo, generando un dato del año siguiente a las cero horas.

. Por otra parte, la hora en el índice del objeto xts devuelto incluye el huso horario contenido en el archivo epw.

## Value

Un objeto xts con dos columnas:

- G0: Radiación solar global en el plano horizontal (kWh/m2)
- Ta: Temperatura media (°C)

## Author(s)

Daniel G. Paniagua

## References

- Jia H (2020). eplusr: A Toolkit for Using EnergyPlus in R. R package version 0.12.0, <https://CRAN.R-project.org/package=eplusr>.

## Examples

```
# #Importa radiacion global horizontal y temperatua del archivo epw. El archivo especificado
# #debe estar en el directorio de trabajo o bien dar la ruta completa.
#
# rad_dia <- importar_epw("ARG_ROSARIO-AERO_874800_IW2.EPW")
```

---

importar\_optimum

---

*Importa y descarga datos de generación de energía de Optimum*


---

## Description

Importa datos de generación de energía del sistema Optimum, descargándolos directamente, de archivos en formato csv o xls/xlsx descargados desde la web, o de data.frame devuelto por `descarga_optimum()`.

**Usage**

```

importar_optimum(
  file = NULL,
  usuario,
  password,
  medidores = "GEN1HORA",
  fecha_inicio = "2019-01-01",
  fecha_fin = "2019-01-02",
  channel = "Active Import Inc - 0",
  int_entrada = "HORA",
  int_salida = NULL,
  na = "INTERPOLAR",
  mult_us = FALSE,
  consumo = FALSE,
  indicar_na = FALSE
)

```

**Arguments**

file	Cadena de caracteres, data.frame o NULL. Ruta del archivo a importar o nombre del data.frame. Puede ser xls, xlsx o csv. Para csv, el caracter separador debe ser coma. Para descargar datos directamente de Optimum debe ser NULL (por defecto).
usuario	Cadena de caracteres. Nombre de usuario para login en Optimum. Solo se utiliza si los datos se descargan directamente de Optimum. Parámetro de <a href="#">descarga_optimum</a> .
password	Cadena de caracteres. Contraseña para login en Optimum. Solo se utiliza si los datos se descargan directamente de Optimum. Parámetro de <a href="#">descarga_optimum</a> .
medidores	Ver detalles en <a href="#">descarga_optimum</a> .
fecha_inicio	Cadena de caracteres. Fecha de inicio de los datos a descargar, en formato "YYYY-MM-DD". Solo se utiliza si los datos se descargan directamente de Optimum. Parámetro de <a href="#">descarga_optimum</a> .
fecha_fin	Cadena de caracteres. Fecha final de los datos a descargar, en formato "YYYY-MM-DD". Solo se utiliza si los datos se descargan directamente de Optimum. Parámetro de <a href="#">descarga_optimum</a> .
channel	Cadena de caracteres. Canal del medidor que contiene los datos a descargar. Solo se utiliza si los datos se descargan directamente de Optimum. Parámetro de <a href="#">descarga_optimum</a> .
int_entrada	Cadena de caracteres. "15MIN" u "HORA". Corresponde al intervalo de tiempo entre los datos de entrada. No utilizar datos que mezclen estos valores.
int_salida	Cadena de caracteres. "15MIN", "HORA", "DIARIO" o "MENSUAL". Corresponde al intervalo de tiempo entre los datos que se desee para el objeto xts de salida.
na	Cadena de caracteres. NULL, "CERO" o "INTERPOLAR", según cómo se quieran rellenar los huecos: <ul style="list-style-type: none"> <li>• NULL: no se rellenan los huecos.</li> </ul>



```
#
# #####Descargando datos directamente#####
# #El ejemplo anterior se puede ejecutar usando directamente la funcion importar_optimum.
# #Para este ejemplo, la funcion se ejecuto en 1433 segundos (23min 53seg aprox).:
# list.prueba_dem_year <- importar_optimum(usuario="usuario_valido", password = "password_valido",
#                                           medidores = "DEM15MIN", fecha_inicio = "2019-03-01",
#                                           fecha_fin = "2020-04-30",int_entrada = "15MIN",
#                                           mult_us = TRUE, consumo = TRUE)
```

---

info\_huecos

*Devuelve data.frame con cantidad de datos por intervalo de tiempo*


---

### Description

Devuelve data.frame con cantidad de datos del xts segmentado por intervalo de tiempo, indicando si existen huecos y detallando su cantidad y longitud.

### Usage

```
info_huecos(xts)
```

### Arguments

xts                      Objeto xts a examinar.

### Value

data.frame con dos columnas:

- "dif": contiene todos los intervalos de tiempo (en minutos) entre datos, incluyendo el estándar. Si solo hay una fila, quiere decir que no hay huecos.
- "cant": cantidad de datos faltantes (huecos) del intervalo de tiempo correspondiente.

### Author(s)

Daniel G. Paniagua

### Examples

```
# #####DATOS CON HUECOS#####
# #Descarga datos de 1 medidor con datos cada 15 minutos de un mes, sin rellenar huecos (na=NULL)
# datos_con_huecos <- importar_optimum(usuario="usuario_valido", password="password_valido",
#                                       medidores="CIR0141449301", fecha_inicio = "2019-03-01",
#                                       fecha_fin = "2019-03-31", int_entrada = "15MIN",
#                                       consumo = TRUE, na=NULL)
#
# #Ejecuta info_huecos
# huecos <- info_huecos(datos_con_huecos)
#
```



```
# #Viendo el data.frame "huecos", puede verse que los datos descargados tienen 62 huecos de
# #135 minutos. Esto es, dos huecos de 135 minutos por dia.
#
# #####DATOS SIN HUECOS#####
# #Descarga datos de 1 medidor con datos cada una hora de un mes, sin rellenar huecos (na=NULL)
# datos_sin_huecos <- importar_optimum(usuario="usuario_valido", password="password_valido",
#                                     medidores="CIR0141437453", fecha_inicio = "2019-03-01",
#                                     fecha_fin = "2019-03-31", int_entrada = "HORA", na=NULL)
#
# #Ejecuta info_huecos
# huecos <- info_huecos(datos_sin_huecos)
#
# #Viendo el data.frame "huecos", puede verse que los datos descargados no tienen huecos, ya que el
# #data.frame solo tiene una fila, correspondiente al intervalo de tiempo estandar entre datos.
```

pr

*Calcula PR a partir de datos de generación real y radiación solar efectiva en el plano del panel*

## Description

Calcula PR a partir de datos de generación real y radiación solar efectiva en el plano del panel

## Usage

```
pr(
  Gef,
  int_entrada = "MENSUAL",
  Pn = 1560,
  year = as.POSIXlt(Sys.Date())$year + 1900 - 1
)
```

## Arguments

Gef	Objeto xts con al menos una columna "Gef" con datos de radiación en el plano del panel FV y otra columna "gen" con datos de generación de energía de ese sistema. Las unidades de ambas columnas se deben corresponder.
int_entrada	Cadena de caracteres. "MENSUAL" o "DIARIO", según lo sean los datos de entrada.
Pn	Numérico. Potencia nominal del sistema en Wp.
year	Numérico. Año al que corresponden los datos.

## Value

El objeto xts original con una columna adicional con el PR.

## Author(s)

Daniel G. Paniagua

## References

- ""van Sark, Wilfried & Reich, Nils & Müller, Björn & Armbruster, Alfons & Kiefer, Klaus & Reise, Christian. (2012). Review of PV performance ratio development. 10.13140/2.1.2138.7204.

## Examples

```
# library(OESolar)
#
# #importa datos de generacion cinco medidores de sistema Optimum.
# gen_dia <- importar_optimum(usuario = "usuario_valido", password = "password_valido",
#                             medidores = "GEN1HORA", fecha_inicio = "2019-01-01",
#                             fecha_fin = "2019-12-31", int_entrada = "HORA",
#                             int_salida = "DIARIO", na="INTERPOLAR", mult_us = TRUE)
# nombres <- names(gen_dia)
#
# #Convierte el objeto "list" a una unica tabla donde cada medidor queda en una columna distinta.
# xts.gen_dia <- do.call(merge, gen_dia)
# names(xts.gen_dia) <- nombres
#
# #Agrega una columna con el promedio de los cinco usuarios para cada dia
# xts.gen_dia$gen <- rowMeans(xts.gen_dia)
#
# #Descarga datos de radiacion global en el plano horizontal de Nasa
# rad_dia <- rad_nasa(intervalo = "DIARIO", lonlat = c(-61.60,-32.78),
#                   fechas = c("2019-01-01","2019-12-31"))
# #Obtiene los valores de radiacion en el plano inclinado
# rad_dia_inclin <- rad_inclin(rad_dia, lat = -32.78, beta = 30, int_entrada = "DIARIO")
#
# #Combina la tabla con los datos de generacion con la tabla de datos de radiacion.
# pr <- merge(xts.gen_dia, rad_dia_inclin)
#
# #Aplica la funcion pr, que agrega una columna a la tabla con el valor de PR.
# pr <- pr(pr, int_entrada="DIARIO", Pn=1560)
```

---

prod\_energia

*Calcula la producción de energía de un generador fotovoltaico conectado a red.*

---

## Description

Calcula la producción de energía de un generador fotovoltaico conectado a red a partir de datos de radiación solar global y temperatura media en el plano horizontal y de parámetros característicos de los paneles FV, inversor y otros componentes. Los datos de radiación pueden ser intradiarios, diarios o mensuales.

## Usage

```
prod_energia(
  G0,
```

```

int_entrada = "MENSUAL",
int_salida = "MENSUAL",
lon = -62.8,
lat = -32.9,
beta = 30,
alfa = 0,
Pn = 1.5,
iS = 2,
year = as.POSIXlt(Sys.Date())$year + 1900 - 1,
modulo = list(Vocn = 37.7, Iscn = 8.95, Vmn = 30.3, Imn = 8.58, Ncs = 60, Ncp = 1,
  CoefVT = 0.00292, TONC = 45),
generador = list(Nms = 6, Nmp = 1),
inversor = list(Ki = c(0.0095, 0.015, 0.02), Pinv = 1500, Vmin = 160, Vmax = 500,
  Gumb = 20),
rendimientos = list(ModQual = 3, ModDisp = 2, OhmDC = 1.5, OhmAC = 1.5, MPP = 1,
  TrafoMT = 0, Disp = 0.5)
)

```

### Arguments

G0	xts con dos columnas: "G0" con radiación en el plano horizontal en kW/m <sup>2</sup> y "Ta" con temperatura media en °C. Para datos mensuales, el índice de fechas en formato yearmon. Para datos intradiarios, el índice de fechas debe incluir el huso horario correspondiente. Puede ser el resultado de rad_nasa o importar_epw.
int_entrada	Cadena de caracteres: "MENSUAL", "DIARIO" o "INTRADIARIO" según corresponda a los datos de radiación y temperatura provistos en GO.
int_salida	Cadena de caracteres: "MENSUAL", "DIARIO" o "INTRADIARIO" según se prefieran los datos de salida de la función. Para int_entrada="MENSUAL", los datos de salida intradiarios corresponden a un día por mes (día medio mensual). Además, para int_entrada="MENSUAL", el objeto xts devuelto para int_salida="MENSUAL" y para int_salida="DIARIO" es el mismo.
lon	Numérico. Longitud de la ubicación del lugar en grados. Solo utilizado para datos de entrada intradiarios.
lat	Numérico. Latitud de la ubicación del lugar en grados. Negativo para hemisferio sur.
beta	Numérico. Inclinación del panel en grados.
alfa	Numérico. Azimuth en grados. Para hemisferio norte, el sur es cero, es negativo hacia el este y positivo hacia el oeste. El package solaR no especifica para el hemisferio sur, pero se verificó que el norte es cero. Tomar mismo criterio que para hemisferio norte para este y oeste.
Pn	Numérico. Potencia nominal del sistema en kWp.
iS	Numérico. Grado de suciedad. Su valor puede ser 0,1,2,3,4. iS=0 no considera el efecto de suciedad y reflexión. iS=1 a iS=4 tiene en cuenta el efecto de suciedad y reflexión, donde iS=1 corresponde a una superficie limpia e iS=4 corresponde a una superficie sucia. El valor por defecto es 2.
year	Numérico. Indicar año al que corresponden los datos.

modulo	Objeto "list". Datos según package solaR: Ver sección "Details" y "Usage" para valores por defecto. Ver además argumento module en <a href="#">prodGCPV</a> .
generador	Objeto "list". Datos según package solaR. Ver sección "Details" y "Usage" para valores por defecto. Ver además argumento generator en <a href="#">prodGCPV</a> .
inversor	Objeto "list". Datos según package solaR. Ver sección "Details" y "Usage" para valores por defecto. Ver además argumento inverter en <a href="#">prodGCPV</a> .
rendimientos	Objeto "list". Datos según package solaR. Ver sección "Details" y "Usage" para valores por defecto. Ver además argumento effSys en <a href="#">prodGCPV</a> .

## Details

Descripción adicional de parámetros. Elementos de los siguientes objetos "list".

modulo (valores por defecto para módulo LV Energy LVE60PS):

- Vocn: open-circuit voltage of the module at Standard Test Conditions (V).
- Iscn: short circuit current of the module at Standard Test Conditions (A).
- Vmn: maximum power point voltage of the module at Standard Test Conditions (V).
- Imn: Maximum power current of the module at Standard Test Conditions (A).
- Ncs: number of cells in series inside the module.
- Ncp: number of cells in parallel inside the module.
- CoefVT: coefficient of decrement of voltage of each cell with the temperature (volts per celsius degree).
- TONC: nominal operational cell temperature (°C).

generador (valores por defecto para techos solares PRIER Armstrong):

- Nms: number of modules in series.
- Nmp: number of modules in parallel.

inversor (valores por defecto para inversor SMA SB 1.5):

- Ki: vector of three values, coefficients of the efficiency curve of the inverter (default c(0.0095, 0.015, 0.02), SMA SB 1.5), or a matrix of nine values (3x3) if there is dependence with the voltage (see references solaR). Estos coeficientes corresponden a una función que modela el rendimiento de los inversores. Se pueden ajustar graficando en planilla de cálculos la tabla de valores que dá el fabricante en la hoja de datos del inversor y la dicha función modificando los coeficientes hasta que coincidan (es sencillo partiendo de los valores por defecto dados aquí). La función es:  $n = p_0 / (p_0 + k_0 + k_1 \cdot p_0 + k_2 \cdot p_0)$ , donde "n" es el rendimiento del inversor,  $p_0$  es la potencia de salida normalizada en por unidad (eje x de la gráfica) y  $k_0$ ,  $k_1$  y  $k_2$  son los tres coeficientes respectivamente (en los valores por defecto:  $k_0=0.0095$ ,  $k_1=0.015$  y  $k_2=0.02$ ). Se puede ver detalle en libro de O. Perpiñan (ver referencias).
- Pinv: nominal inverter power (W).
- Vmin, Vmax: minimum and maximum voltages of the MPP range of the inverter (V).
- Gumb: minimum irradiance for the inverter to start (W/m<sup>2</sup>) (default value 20W/m<sup>2</sup>).

rendimientos (valores por defecto de package solaR ajustados con NREL):

- ModQual: average tolerance of the set of modules (porciento), default value is 3.
- ModDisp: module parameter dispersion losses (porciento), default value is 2.
- OhmDC: Joule losses due to the DC wiring (porciento), default value is 1.5.
- OhmAC: Joule losses due to the AC wiring (porciento), default value is 1.5.
- MPP: average error of the MPP algorithm of the inverter (porciento), default value is 1.
- TrafoMT: losses due to the MT transformer (porciento), default value is 0.
- Disp: losses due to stops of the system (porciento), default value is 0.5.

## Value

Devuelve objeto xts con las columnas detalladas a continuación:

- Para `int_salida="MENSUAL"`:  $G0_{dm}$  (kW/m<sup>2</sup>),  $T_{adm}$  (°C),  $G_{efdm}$  (o  $G_{dm}$  si  $iS=0$ ) (kW/m<sup>2</sup>),  $Eac_{dm}$  (energía producida en un día en kWh),  $Eac_m$  (energía producida en el mes en kWh) y  $PR_{dm}$  (PR diario mensual del sistema calculado).
- Para `int_salida="DIARIO"`:  $G0_d$  (kW/m<sup>2</sup>),  $T_{ad}$  (°C),  $G_{efd}$  (o  $G_d$  si  $iS=0$ ) (kW/m<sup>2</sup>),  $Eac_d$  (energía producida en un día en kWh) y  $PR_d$  (PR diario del sistema calculado).
- Para `int_salida="INTRADIARIO"`:  $G0$  (kW/m<sup>2</sup>),  $T_a$  (°C),  $G_{ef}$  (o  $G$  si  $iS=0$ ) (kW/m<sup>2</sup>),  $Pac$  (potencia producida en el intervalo de tiempo en kW) y  $PR$  (PR en el intervalo de tiempo del sistema calculado).

## Author(s)

Daniel G. Paniagua

## References

- Perpiñán, O, Energía Solar Fotovoltaica, 2015. (<http://oscarperpinan.github.io/esf/>)
- Perpiñán O (2012). "solaR: Solar Radiation and Photovoltaic Systems with R." Journal of Statistical Software, 50(9), 1–32. <http://www.jstatsoft.org/v50/i09/>.

## Examples

```
# #Calculo de las distintas combinaciones de intervalos de datos de entrada y salida.
#
# rad_intra <- importar_epw("ARG_MARCOS-JUAREZ-AERO_874670_IW2.EPW")
#
# rad_dia <- rad_nasa(intervalo = "DIARIO", lonlat = c(-62.15,-32.7),
#                     fechas = c("2019-01-01","2019-12-31"))
#
# rad_mes <- rad_nasa(intervalo = "MENSUAL", lonlat = c(-62.15,-32.7), fechas = 2019)
#
# #####1)DATOS DE ENTRADA INTRADIARIO#####
#
# prod1_intra_intra_is0 <- prod_energia(G0=rad_intra, int_entrada="INTRADIARIO",
#                                     int_salida="INTRADIARIO",
#                                     lon=-62.15, lat=-32.7, beta=30, alfa=0, Pn=1.56, iS = 0,
#                                     year=as.POSIXlt(Sys.Date())$year+1900-1)
```

```

#
# prod2_intra_intra_is2 <- prod_energia(G0=rad_intra, int_entrada="INTRADIARIO",
#                                     int_salida="INTRADIARIO",
#                                     lon=-62.15, lat=-32.7, beta=30, alfa=0, Pn=1.56, iS = 2,
#                                     year=as.POSIXlt(Sys.Date())$year+1900-1)
#
# prod3_intra_dia_is0 <- prod_energia(G0=rad_intra, int_entrada="INTRADIARIO",
#                                     int_salida="DIARIO",
#                                     lon=-62.15, lat=-32.7, beta=30, alfa=0, Pn=1.56, iS = 0,
#                                     year=as.POSIXlt(Sys.Date())$year+1900-1)
#
# prod4_intra_dia_is2 <- prod_energia(G0=rad_intra, int_entrada="INTRADIARIO",
#                                     int_salida="DIARIO",
#                                     lon=-62.15, lat=-32.7, beta=30, alfa=0, Pn=1.56, iS = 2,
#                                     year=as.POSIXlt(Sys.Date())$year+1900-1)
#
# prod5_intra_mes_is0 <- prod_energia(G0=rad_intra, int_entrada="INTRADIARIO",
#                                     int_salida="MENSUAL",
#                                     lon=-62.15, lat=-32.7, beta=30, alfa=0, Pn=1.56, iS = 0,
#                                     year=as.POSIXlt(Sys.Date())$year+1900-1)
#
# prod6_intra_mes_is2 <- prod_energia(G0=rad_intra, int_entrada="INTRADIARIO",
#                                     int_salida="MENSUAL",
#                                     lon=-62.15, lat=-32.7, beta=30, alfa=0, Pn=1.56, iS = 2,
#                                     year=as.POSIXlt(Sys.Date())$year+1900-1)
#
# #####2)DATOS DE ENTRADA DIARIO#####
#
# prod7_dia_intra_is0 <- prod_energia(G0=rad_dia, int_entrada="DIARIO", int_salida="INTRADIARIO",
#                                     lon=-62.15, lat=-32.7, beta=30, alfa=0, Pn=1.56, iS = 0,
#                                     year=as.POSIXlt(Sys.Date())$year+1900-1)
#
# prod8_dia_intra_is2 <- prod_energia(G0=rad_dia, int_entrada="DIARIO", int_salida="INTRADIARIO",
#                                     lon=-62.15, lat=-32.7, beta=30, alfa=0, Pn=1.56, iS = 2,
#                                     year=as.POSIXlt(Sys.Date())$year+1900-1)
#
# prod9_dia_dia_is0 <- prod_energia(G0=rad_dia, int_entrada="DIARIO", int_salida="DIARIO",
#                                     lon=-62.15, lat=-32.7, beta=30, alfa=0, Pn=1.56, iS = 0,
#                                     year=as.POSIXlt(Sys.Date())$year+1900-1)
#
# prod10_dia_dia_is2 <- prod_energia(G0=rad_dia, int_entrada="DIARIO", int_salida="DIARIO",
#                                     lon=-62.15, lat=-32.7, beta=30, alfa=0, Pn=1.56, iS = 2,
#                                     year=as.POSIXlt(Sys.Date())$year+1900-1)
#
# prod11_dia_mes_is0 <- prod_energia(G0=rad_dia, int_entrada="DIARIO", int_salida="MENSUAL",
#                                     lon=-62.15, lat=-32.7, beta=30, alfa=0, Pn=1.56, iS = 0,
#                                     year=as.POSIXlt(Sys.Date())$year+1900-1)
#
# prod12_dia_mes_is2 <- prod_energia(G0=rad_dia, int_entrada="DIARIO", int_salida="MENSUAL",
#                                     lon=-62.15, lat=-32.7, beta=30, alfa=0, Pn=1.56, iS = 2,
#                                     year=as.POSIXlt(Sys.Date())$year+1900-1)
#
#
#

```

```

# #####3)DATOS DE ENTRADA MENSUAL#####
# prod13_mes_intra_is0 <- prod_energia(G0=rad_mes, int_entrada="MENSUAL",
#                                     int_salida="INTRADIARIO",
#                                     lon=-62.15, lat=-32.7, beta=30, alfa=0, Pn=1.56, iS = 0,
#                                     year=as.POSIXlt(Sys.Date())$year+1900-1)
#
# prod14_mes_intra_is2 <- prod_energia(G0=rad_mes, int_entrada="MENSUAL",
#                                     int_salida="INTRADIARIO",
#                                     lon=-62.15, lat=-32.7, beta=30, alfa=0, Pn=1.56, iS = 2,
#                                     year=as.POSIXlt(Sys.Date())$year+1900-1)
#
# prod15_mes_dia_is0 <- prod_energia(G0=rad_mes, int_entrada="MENSUAL", int_salida="DIARIO",
#                                    lon=-62.15, lat=-32.7, beta=30, alfa=0, Pn=1.56, iS = 0,
#                                    year=as.POSIXlt(Sys.Date())$year+1900-1)
#
# prod16_mes_dia_is2 <- prod_energia(G0=rad_mes, int_entrada="MENSUAL", int_salida="DIARIO",
#                                    lon=-62.15, lat=-32.7, beta=30, alfa=0, Pn=1.56, iS = 2,
#                                    year=as.POSIXlt(Sys.Date())$year+1900-1)
#
# prod17_mes_mes_is0 <- prod_energia(G0=rad_mes, int_entrada="MENSUAL", int_salida="MENSUAL",
#                                    lon=-62.15, lat=-32.7, beta=30, alfa=0, Pn=1.56, iS = 0,
#                                    year=as.POSIXlt(Sys.Date())$year+1900-1)
#
# prod18_mes_mes_is2 <- prod_energia(G0=rad_mes, int_entrada="MENSUAL", int_salida="MENSUAL",
#                                    lon=-62.15, lat=-32.7, beta=30, alfa=0, Pn=1.56, iS = 2,
#                                    year=as.POSIXlt(Sys.Date())$year+1900-1)

```

---

rad\_inclin

*Calcula el valor de radiación solar en el plano inclinado.*


---

## Description

Calcula el valor de radiación solar en el plano inclinado a partir de la radiación solar en el plano horizontal y la temperatura. Funciona para datos diarios y datos mensuales.

## Usage

```

rad_inclin(
  horiz,
  lon = -62.8,
  lat = -32.9,
  beta = 45,
  alfa = 0,
  int_entrada = "MENSUAL",
  iS = 2,
  year = as.POSIXlt(Sys.Date())$year + 1900 - 1
)

```

**Arguments**

horiz	xts con dos columnas: "G0" con radiación en el plano horizontal en kW/m <sup>2</sup> y "Ta" con temperatura media en °C. Para datos mensuales, el índice de fechas en formato yearmon. Para datos intradiarios, el índice de fechas debe incluir el huso horario correspondiente.
lon	Numérico. Longitud de la ubicación del lugar en grados. Solo utilizado para datos de entrada intradiarios.
lat	Numérico. Latitud de la ubicación del lugar en grados. Negativo para hemisferio sur.
beta	Numérico. Inclinación del panel en grados.
alfa	Numérico. Azimuth en grados. Para hemisferio norte, el sur es cero, es negativo hacia el este y positivo hacia el oeste. El package solaR no especifica para el hemisferio sur, pero se verificó que el norte es cero. Tomar mismo criterio que para hemisferio norte para este y oeste.
int_entrada	Cadena de caracteres: "MENSUAL", "DIARIO" o "INTRADIARIO" según corresponda a los datos de radiación y temperatura provistos en horiz.
iS	Numérico. Grado de suciedad. Su valor puede ser 0,1,2,3,4. iS=0 no considera el efecto de suciedad y reflexión. iS=1 a iS=4 tiene en cuenta el efecto de suciedad y reflexión, donde iS=1 corresponde a una superficie limpia e iS=4 corresponde a una superficie sucia. El valor por defecto es 2.
year	Numérico. Año al que corresponden los datos. Solo utilizado para datos mensuales. Valor por defecto: año anterior al actual.

**Value**

Devuelve el xts horiz con una columna adicional con la radiación en el plano del panel.

**Author(s)**

Daniel G. Paniagua

**References**

- Perpiñán, O, Energía Solar Fotovoltaica, 2015. (<http://oscarperpinan.github.io/esf/>)
- Perpiñán O (2012). "solaR: Solar Radiation and Photovoltaic Systems with R." Journal of Statistical Software, 50(9), 1–32. <http://www.jstatsoft.org/v50/i09/>.

**Examples**

```
# #Calcula valores de radiacion en el plano inclinado para datos diarios de
# #de radiacion solar de Rosario del año 2019.
#
# rad <- rad_nasa(intervalo = "DIARIO", fechas=c("2019-01-01","2019-12-31"))
# rad_incl <- rad_inclin(rad, beta=28, alfa=11, int_entrada = "DIARIO")
#
# #Calcula valores de radiacion en el plano inclinado para datos mensuales
# #(promediados de los últimos 30 años) de radiacion de la ciudad de Rosario, Argentina.
```



```
#
# rad <- rad_nasa(intervalo = "MENSUAL", fechas = 2018)
# rad_incl <- rad_inclin(rad, beta=28, alfa=11)
```

rad\_nasa

*Obtiene datos de radiación de Nasa Power***Description**

Obtiene datos de radiación solar global en el plano horizontal y de temperatura de la base de datos Nasa Power.

**Usage**

```
rad_nasa(
  intervalo = "HISTORICO",
  lonlat = c(-60.8, -32.9),
  fechas = as.POSIXlt(Sys.Date())$year + 1900 - 1
)
```

**Arguments**

intervalo	Una cadena de caracteres. intervalo="DIARIO" para datos diarios. intervalo="MENSUAL" para datos mensuales de un año especificado en "fechas". intervalo="HISTORICO" para promedio mensual sobre período de 30 años.
lonlat	Vector con longitud y latitud de la localidad.
fechas	Para: intervalo="DIARIO", vector con fecha inicio y fin. Ej: c("2018-01-01", "2018-12-31"). intervalo="MENSUAL", especificar un año con número. intervalo="HISTORICO", especificar un año con número. Los datos no corresponderán al año especificado, sino que será un promedio sobre los últimos 30 años. Este valor se utilizará para asignar un año a los datos, que será útil para futuras comparaciones con datos de generación y cálculo de PR. Valor por defecto: año anterior al año en curso.

**Value**

Un xts con dos columnas: G0 y Ta (radiación global en el plano horizontal y temperatura ambiente media). Para "HISTORICO" y "MENSUAL" el formato del índice de fechas es yearmon.

**Author(s)**

Daniel G. Paniagua

## References

- Sparks A (2019). nasapower: NASA-POWER Data from R. R package version 1.1.3, <https://CRAN.R-project.org/package=nasapower>.
- Sparks AH (2018). “nasapower: A NASA POWER Global Meteorology, Surface Solar Energy and Climatology Data Client for R.” The Journal of Open Source Software, 3(30), 1035. doi: 10.21105/joss.01035.

## Examples

```
# #Devuelve dataframe con datos mensuales (promediados de los ultimos
# #30 anos) de radiacion y temperatura de la ciudad de Rosario, Argentina.
#
# rad <- rad_nasa()
#
# #Devuelve dataframe con datos mensuales de radiacion y temperatura
# #de Rosario, Argentina del 2018.
#
# rad <- rad_nasa(intervalo = "MENSUAL", fechas = 2018)
#
# #Devuelve dataframe con datos diarios de radiacion y temperatura
# #de Marcos Juarez, Argentina del 2019.
#
# rad <- rad_nasa(intervalo = "DIARIO",
#                 lonlat = c(-62.15,-32.7),
#                 fechas = c("2019-01-01","2019-12-31"))
```

---

rellena\_huecos

*Rellena huecos de datos faltantes en xts con datos de energía*


---

## Description

Rellena huecos de datos faltantes en xts con datos de energía cada 15 minutos o cada 1 hora.

## Usage

```
rellena_huecos(
  xts,
  int_entrada = "HORA",
  na = "INTERPOLAR",
  consumo = FALSE,
  indicar_na = FALSE
)
```

## Arguments

**xts**                      Objeto xts con datos de energía a rellenar. La columna con los datos debe llamarse "energía".

int_entrada	Cadena de caracteres. "15MIN" O "HORA". Corresponde al intervalo de tiempo entre los datos de entrada. No utilizar datos que mezclen estos valores.
na	Cadena de caracteres. "CERO" o "INTERPOLAR", según cómo se quieran rellenar los huecos: <ul style="list-style-type: none"> <li>• "CERO": rellena todos los huecos con ceros.</li> <li>• "INTERPOLAR": rellena huecos interpolando los datos faltantes con los datos anterior y posterior al hueco. Si consumo=TRUE, interpola para todos los horarios. Si consumo=FALSE, interpola en horarios diurnos y completa con cero en horarios nocturnos.</li> </ul>
consumo	Lógico. TRUE si los datos son de consumo. Si es FALSE (cuando los datos son de generación), la interpolación para NA en horario nocturno no se hace y se completa con ceros.
indicar_na	Lógico. Si es TRUE, los datos agregados se identifican con un "1" en una columna adicional llamada "rellenado".

### Value

xts estrictamente regular según el intervalo de tiempo ingresado en int\_entrada.

### Author(s)

Daniel G. Paniagua

### Examples

```
# #####DATOS CON HUECOS#####
# #Descarga datos de 1 medidor con datos cada 15 minutos de un mes, sin rellenar huecos (na=NULL)
# datos_con_huecos <- importar_optimum(usuario="usuario_valido",
#                                     password="password_valido",
#                                     medidores="CIR0141449301", fecha_inicio = "2019-03-01",
#                                     fecha_fin = "2019-03-31", int_entrada = "15MIN",
#                                     consumo = TRUE, na=NULL)
#
# datos_rellenados <- rellena_huecos(xts=datos_con_huecos, int_entrada="15MIN", na="INTERPOLAR",
#                                   consumo=TRUE, indicar_na=TRUE)
#
# info_huecos(datos_con_huecos)
# #62 huecos de 135 minutos (8 datos faltantes de 15 minutos por hueco)
#
# length(datos_rellenados$energia)-length(datos_con_huecos$energia)
# #[1] 496
# #Puede verse que se rellenaron 496 datos (8x62)
```

# Index

agrupar\_por, [2](#)

descarga\_optimum, [3](#), [6](#)

Epw, [5](#)

importar\_epw, [4](#)

importar\_optimum, [5](#)

info\_huecos, [8](#)

pr, [9](#)

prod\_energia, [10](#)

prodGCPV, [12](#)

rad\_inclin, [15](#)

rad\_nasa, [17](#)

rellena\_huecos, [18](#)