

# Package ‘epe4md’

May 8, 2026

**Title** EPE's 4MD Model to Forecast the Adoption of Distributed Generation

**Version** 0.1.4

**Language** pt-BR

**Description** EPE's (Empresa de Pesquisa Energética) 4MD (Modelo de Mercado da Micro e Minigeração Distribuída - Micro and Mini Distributed Generation Market Model) model to forecast the adoption of Distributed Generation. Given the user's assumptions, it is possible to estimate how many consumer units will have distributed generation in Brazil over the next 10 years, for example. In addition, it is possible to estimate the installed capacity, the amount of investments that will be made in the country and the monthly energy contribution of this type of generation. <[https://www.epe.gov.br/sites-pt/publicacoes-dados-abertos/publicacoes/PublicacoesArquivos/publicacao-689/topico-639/NT\\_Metodologia\\_4MD\\_PDE\\_2032\\_VF.pdf](https://www.epe.gov.br/sites-pt/publicacoes-dados-abertos/publicacoes/PublicacoesArquivos/publicacao-689/topico-639/NT_Metodologia_4MD_PDE_2032_VF.pdf)>.

**License** GPL (>= 3)

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.2.3

**Suggests** testthat (>= 3.0.0), knitr, rmarkdown, tidyverse

**Config/testthat/edition** 3

**Config/testthat/parallel** true

**Imports** dplyr (>= 1.1.1), scales, purrr, tidyr, readxl, jrvFinance, janitor, stats, furr, zoo, lubridate, tsibble, stringr, fabletools, feasts, tibble, assertthat, forcats, future, magrittr, readr, utils, ggplot2

**Depends** R (>= 3.5.0)

**URL** <https://epe-gov-br.github.io/epe4md/>

**NeedsCompilation** no

**Author** Gabriel Konzen [aut, cre],  
Bruno Crotman [aut],  
João Santos [aut],  
Leticia Minini [aut],  
Empresa de Pesquisa Energética [cph, fnd]

**Maintainer** Gabriel Konzen <gabriel.konzen@epe.gov.br>

**Repository** CRAN

**Date/Publication** 2023-07-05 14:23:07 UTC

## Contents

epe4md_calcula . . . . .	2
epe4md_calibra_curva_s . . . . .	6
epe4md_casos_payback . . . . .	8
epe4md_fatores_publicacao . . . . .	10
epe4md_graf_geracao_ano . . . . .	10
epe4md_graf_geracao_mes . . . . .	11
epe4md_graf_part_fonte_geracao . . . . .	13
epe4md_graf_part_fonte_potencia . . . . .	14
epe4md_graf_part_segmento . . . . .	15
epe4md_graf_pot_acum . . . . .	16
epe4md_graf_pot_anual . . . . .	17
epe4md_graf_pot_regiao . . . . .	18
epe4md_graf_pot_segmento . . . . .	20
epe4md_investimentos . . . . .	21
epe4md_mercado_potencial . . . . .	22
epe4md_payback . . . . .	23
epe4md_prepara_base . . . . .	26
epe4md_proj_adotantes . . . . .	28
epe4md_proj_geracao . . . . .	29
epe4md_proj_mensal . . . . .	31
epe4md_proj_potencia . . . . .	33
epe4md_sumariza_resultados . . . . .	35
paleta_epe . . . . .	36
<b>Index</b>	<b>37</b>

---

epe4md_calcula	<i>Roda o modelo 4MD</i>
----------------	--------------------------

---

## Description

O resultado do modelo 4MD são projeções de capacidade instalada, número de adotantes e geração de energia em base mensal.

**Usage**

```
epe4md_calcula(  
  premissas_reg,  
  ano_base,  
  sequencial = FALSE,  
  filtro_de_uf = "N",  
  filtro_nome4md = "N",  
  filtro_de_segmento = "N",  
  filtro_de_custo_unitario_max = NULL,  
  ano_max_resultado = 2050,  
  altera_sistemas_existentes = FALSE,  
  ano_decisao_alteracao = 2023,  
  inflacao = 0.0375,  
  taxa_desconto_nominal = 0.13,  
  custo_reforco_rede = 200,  
  ano_troca_inversor = 11,  
  pagamento_disponibilidade = 0.3,  
  disponibilidade_kwh_mes = 100,  
  filtro_renda_domicilio = "maior_3sm",  
  desconto_capex_local = 0,  
  anos_desconto = 0,  
  tx_cresc_grupo_a = 0.016,  
  spb = 0.3,  
  p_max = 0.01,  
  q_max = 1,  
  filtro_comercial = NA_real_,  
  ajuste_ano_corrente = FALSE,  
  ultimo_mes_ajuste = NA_integer_,  
  metodo_ajuste = NA_character_,  
  dir_dados_premissas = NA_character_  
)
```

**Arguments**

- `premissas_reg` data.frame. Input de premissas regulatórias para serem consideradas nos cálculos. O dataframe deve ter as seguintes colunas
- ano, numérico
  - alternativa, numérico. Uma das seguintes opções:
    - 0: Consumidor compensa todas as componentes tarifárias;
    - 1: Paga TUSD Distribuição;
    - 2: Anterior + TUSD Transmissão.
    - 3: Anterior + TUSD Encargos.
    - 4: Anterior + TUSD Perdas.
    - 5: Anterior + TE Encargos. Ou seja, compensa somente a TE Energia.
  - `p_transicao`, numérico. Parcela do custo da alternativa escolhida no parâmetro alternativa a ser pago pelo consumidor

- binomia e, binário. Define se há cobrança de uma tarifa binômica na baixa tensão, em que as componentes TUSD Distribuição e TUSD Transmissão passariam a ser cobradas de forma fixa, não sendo passíveis de compensação
- demanda\_g, binário. Define se há cobrança de TUSDg para a demanda de consumidores do grupo A. Caso seja FALSE, é considerada a cobrança da TUSD consumo.

Um arquivo excel instalado com este pacote, acessível via `system.file("dados_premissas/2021/prem", package = "epe4md")`, contém um exemplo de premissas de entrada.

ano_base	numeric. Ano base da projeção. Define o ano em que a função irá buscar a base de dados. Último ano completo realizado.
sequencial	logic. Parâmetro que define se a projeção deve ser realizada de forma sequencial ou paralela. Para executar a projeção de forma sequencial defina o parâmetro como TRUE. Default FALSE.
filtro_de_uf	string. Parâmetro que define uma unidade federativa (UF) a ser filtrada. Caso uma UF não seja indicada ou seja informado um valor inválido, o resultado será apresentado sem filtros.
filtro_nome4md	string. Parâmetro que define o nome de uma concessionária de energia a ser filtrado. Caso não seja informado um valor ou seja informado um valor inválido o resultado será apresentado sem filtro.
filtro_de_segmento	string. Parâmetro que define um segmento a ser filtrado. Pode se escolher entre "comercial_at", "comercial_at_remoto", "comercial_bt", "residencial" e "residencial_remoto". Caso não seja informado um valor ou seja informado um valor inválido o resultado será apresentado sem filtro.
filtro_de_custo_unitario_max	numeric. Parâmetro que define o valor máximo do custo unitário para ser utilizado no cálculo do payback. Default igual a NULL.
ano_max_resultado	numeric. Ano final para apresentação dos resultados. Máximo igual a 2050. Default igual a 2050.
altera_sistemas_existentes	logic. TRUE se alterações regulatórias afetam investimentos realizados em anos anteriores à revisão da regulação. Default igual a FALSE.
ano_decisao_alteracao	numeric. Ano em que são definidas novas regras e se tornam de conhecimento público. Esse parâmetro só tem efeito caso o anterior seja igual a TRUE. Default igual a 2023.
inflacao	numeric. Taxa anual de inflacao considerada no reajuste das tarifas e para calcular o retorno real de projetos. Default igual a 0.0375.
taxa_desconto_nominal	numeric. Taxa de desconto nominal considerada nos cálculos de payback descontado. Default igual a 0.13.
custo_reforco_rede	numeric. Custo em R\$/kW aplicado a projetos de geracao remota em Alta Tensão. Representa um custo pago pelo empreendedor para reforços na rede. Default igual a 200.

ano_troca_inversor	numeric. Ano, a partir do ano de instalação, em que é realizada a troca do inversor fotovoltaico. Default igual a 11.
pagamento_disponibilidade	numeric. Percentual de meses em que o consumidor residencial paga custo de disponibilidade em função da variabilidade da geração FV. Default igual a 0.3.
disponibilidade_kwh_mes	numeric. Consumo de disponibilidade do consumidor em kWh/mês. Default igual a 100, equivalente a um consumidor trifásico.
filtro_renda_domicilio	string. Define o filtro aplicado a consumidores residenciais, de acordo com a renda mensal do responsável, em salários mínimos. Permite: "total", "maior_1sm", "maior_2sm", "maior_3sm" ou "maior_5sm". Default igual a "maior_3sm".
desconto_capex_local	numeric. Percentual de desconto a ser aplicado no CAPEX de sistemas de geração local(ex: 0.1) para simulação de incentivos. Default igual a 0.
anos_desconto	vector. Anos em que há a incidência do desconto no CAPEX. Default igual a 0.
tx_cresc_grupo_a	numeric. Taxa de crescimento anual dos consumidores cativos do Grupo A. Default igual a 0.016 representa crescimento entre 2006 e 2019.
spb	numeric. Fator de Sensibilidade ao Payback (SPB). Default igual a 0.3.
p_max	numeric. Fator de inovação (p) máximo. Default igual a 0.01.
q_max	numeric. Fator de imitação (q) máximo. Default igual a 1.
filtro_comercial	numeric. Fator percentual para definir o nicho do segmento comercial. Default é calculado pelo modelo com base no nicho residencial.
ajuste_ano_corrente	logic. Se TRUE indica que a projeção deverá incorporar o histórico mensal recente, verificado em parte do primeiro ano após o ano base. Default igual a FALSE. O arquivo base_mmgd.xlsx deve incorporar esse histórico.
ultimo_mes_ajuste	numeric. Último mês com dados completos na base_ano_corrente. Default igual a NA. Só tem efeito caso ajuste_ano_corrente seja igual a TRUE.
metodo_ajuste	string. Se igual a "extrapola" o modelo irá extrapolar a potência e o número de adotantes até o final do ano base + 1 com base no verificado até o ultimo_mes_ajuste. Default igual a NA. Só tem efeito caso ajuste_ano_corrente seja igual a TRUE.
dir_dados_premissas	Diretório onde se encontram as premissas. Se esse parâmetro não for passado, a função usa os dados default que são instalados com o pacote. É importante que os nomes dos arquivos sejam os mesmos da pasta default.

### Value

data.frame com os resultados da projeção de capacidade instalada de micro e minigeração distribuída, número de adotantes e geração mensal de energia.

**Examples**

```
premissas_regulatorias <- structure(
  list(ano = 2021,
        alternativa = 0,
        p_transicao = 1,
        binomia = FALSE,
        demanda_g = FALSE),
  class = c("tbl_df", "tbl", "data.frame"),
  row.names = c(NA, -1L)
)

resultado <- epe4md_calcula(premissas_reg = premissas_regulatorias,
                           ano_base = 2021,
                           sequencial = FALSE,
                           filtro_de_uf = "RR",
                           filtro_de_segmento = "comercial_at",
                           filtro_de_custo_unitario_max = 6,
                           ano_max_resultado = 2021)
```

---

epe4md\_calibra\_curva\_s

*Calibra o modelo de Bass com dados históricos e gera curvas S de adoção.*

---

**Description**

Calibra o modelo de Bass com dados históricos e gera curvas S de adoção.

**Usage**

```
epe4md_calibra_curva_s(
  resultado_payback,
  consumidores,
  ano_base,
  ano_max_resultado = 2050,
  spb = 0.3,
  p_max = 0.01,
  q_max = 1,
  dir_dados_premissas = NA_character_
)
```

**Arguments**

`resultado_payback` data.frame. Resultado da função [epe4md\\_payback](#).

`consumidores` list. Resultado da função [epe4md\\_mercado\\_potencial](#).

ano_base	numeric. Ano base da projeção. Define o ano em que a função irá buscar a base de dados. Último ano completo realizado.
ano_max_resultado	numeric. Ano final para apresentação dos resultados. Máximo igual a 2050. Default igual a 2050.
spb	numeric. Fator de Sensibilidade ao Payback (SPB). Default igual a 0.3.
p_max	numeric. Fator de inovação (p) máximo. Default igual a 0.01.
q_max	numeric. Fator de imitação (q) máximo. Default igual a 1.
dir_dados_premissas	Diretório onde se encontram as premissas. Se esse parâmetro não for passado, a função usa os dados default que são instalados com o pacote. É importante que os nomes dos arquivos sejam os mesmos da pasta default.

## Value

data.frame com curvas de difusão e mercado potencial

## Examples

```

resultado_payback <- structure(
  list(segmento = c("comercial_at", "comercial_at", "comercial_at",
    "comercial_at", "comercial_at"),
    fonte_resumo = c("Fotovoltaica", "Fotovoltaica", "Fotovoltaica",
    "Fotovoltaica", "Fotovoltaica"),
    fator_autoconsumo = c(0.8, 0.8, 0.8, 0.8, 0.8),
    oem_anual = c(0.01, 0.01, 0.01, 0.01, 0.01),
    nome_4md = c("RORAIMA", "RORAIMA", "RORAIMA", "RORAIMA", "RORAIMA"),
    fc = c(0.150559213009562, 0.150559213009562, 0.150559213009562,
    0.150559213009562, 0.150559213009562),
    vida_util = c(25, 25, 25, 25, 25),
    degradacao = c(0.005, 0.005, 0.005, 0.005, 0.005),
    pot_sistemas = c(70, 70, 70, 70, 70),
    geracao_1_kwh = c(92322.9094174635, 92322.9094174635,
    92322.9094174635, 92322.9094174635,
    92322.9094174635),
    ano = c(2017, 2018, 2019, 2020, 2021),
    custo_unitario = c(4.6, 4.23, 3.4, 3.62, 3.81),
    custo_inversor = c(0.69, 0.6345, 0.51, 0.543, 0.5715),
    capex_inicial = c(322000, 296100, 238000, 253400, 266700),
    capex_inversor = c(69795.6224290695, 64181.6267119487,
    51588.0687519209, 54926.1202593981,
    57808.9829249466),
    uf = c("RR", "RR", "RR", "RR", "RR"),
    subsistema = c("MAN", "MAN", "MAN", "MAN", "MAN"),
    regioao = c("N", "N", "N", "N", "N"),
    payback = c(9.8778051007831, 6.84888025046529, 5.42743545180765,
    6.58428138751385, 6.57787077623868),
    payback_desc = c(25, 12.9151902647661, 7.76175962915069,
    11.9207871041042, 11.8969973448089),
    tir_nominal = c(0.117295199807788, 0.177811894010469,

```

```

        0.237962995155112, 0.186855919654218,
        0.1870853811094),
    tir_real = c(0.0769110359593141, 0.135240379769127,
        0.193217344727818, 0.143957512919728,
        0.144178680587373)),
    class = c("tbl_df", "tbl", "data.frame"),
    row.names = c(NA, -5L)
)

consumidores <- list(
  consumidores = structure(
    list(nome_4md = c("RORAIMA", "RORAIMA", "RORAIMA", "RORAIMA", "RORAIMA"),
      ano = c(2017, 2018, 2019, 2020, 2021),
      segmento = c("comercial_at", "comercial_at", "comercial_at",
        "comercial_at", "comercial_at"),
      consumidores = c(656, 678, 751, 764, 791)),
    row.names = c(NA, -5L),
    class = c("tbl_df", "tbl", "data.frame")),
  consumidores_totais = structure(
    list(ano = c(2017, 2018, 2019, 2020, 2021),
      total_ucs = c(187922, 187853, 184276, 182048, 181394),
      segmento = c("comercial_at", "comercial_at", "comercial_at",
        "comercial_at", "comercial_at")),
    row.names = c(NA, -5L),
    class = c("tbl_df", "tbl", "data.frame"))
)

casos_otimizados <-
  epe4md_calibra_curva_s(resultado_payback = resultado_payback,
    consumidores = consumidores,
    ano_base = 2021,
    ano_max_resultado = 2050,
    spb = 0.3,
    p_max = 0.01,
    q_max = 1,
    dir_dados_premissas = NA_character_
)

```

---

epe4md\_casos\_payback *Cria a base de casos para serem simulados posteriormente no cálculo do payback.*

---

### Description

Cria a base de casos para serem simulados posteriormente no cálculo do payback.

### Usage

```

epe4md_casos_payback(
  ano_base,

```

```
ano_max_resultado = 2050,  
inflacao = 0.0375,  
ano_troca_inversor = 11,  
fator_custo_inversor = 0.15,  
dir_dados_premissas = NA_character_  
)
```

### Arguments

**ano\_base** numeric. Ano base da projeção. Define o ano em que a função irá buscar a base de dados. Último ano completo realizado.

**ano\_max\_resultado** numeric. Ano final para apresentação dos resultados. Máximo igual a 2050. Default igual a 2050.

**inflacao** numeric. Taxa de inflação considerada no reajuste das tarifas e para calcular o retorno real de projetos.

**ano\_troca\_inversor** numeric. Ano, a partir do ano de instalação, em que é realizada a troca do inversor fotovoltaico. Default igual a 11.

**fator\_custo\_inversor** numeric. Custo do inversor para a substituição em percentual do CAPEX total do sistema de geração. Default igual a 0.15.

**dir\_dados\_premissas** Diretório onde se encontram as premissas. Se esse parâmetro não for passado, a função usa os dados default que são instalados com o pacote. É importante que os nomes dos arquivos sejam os mesmos da pasta default.

### Value

data.frame. Casos para serem simulados posteriormente no cálculo do payback.

### Examples

```
casos_payback <-  
  epe4md_casos_payback(  
    ano_base = 2021,  
    ano_max_resultado = 2050,  
    inflacao = 0.0375,  
    ano_troca_inversor = 11,  
    fator_custo_inversor = 0.15,  
    dir_dados_premissas = NA_character_  
  )
```

---

`epe4md_fatores_publicacao`*Altera fatores para facilitar entendimento em publicações*

---

**Description**

Altera fatores para facilitar entendimento em publicações

**Usage**

```
epe4md_fatores_publicacao(dados)
```

**Arguments**

`dados` data.frame. Base de dados para ser alterada.

**Value**

data.frame com nomes dos segmentos alterados.

**Examples**

```
dados <- tibble::tibble(  
  fonte_resumo = c("comercial_bt",  
                  "comercial_at_remoto",  
                  "comercial_at",  
                  "residencial_remoto",  
                  "residencial")  
)  
  
fatores <- epe4md_fatores_publicacao(dados)
```

---

`epe4md_graf_geracao_ano`*Gráfico da geração anual em MWmed*

---

**Description**

Gráfico de área que mostra a geração de energia em MWmed agrupada por ano.

**Usage**

```
epe4md_graf_geracao_ano(  
  dados,  
  ano_inicio = 2013,  
  cor = "#953735",  
  tamanho = 14  
)
```

**Arguments**

dados	data.frame. Resultados mensais de potência e energia.
ano_inicio	numeric. Define o ano em que se inicia o eixo x. Default igual a 2013.
cor	string. Escolha da cor do gráfico. Default igual a Vermelho.
tamanho	numeric. Altera o tamanho da fonte da legenda. Default igual a 14.

**Value**

Plota um gráfico de área que mostra a geração de energia em MWmed agrupada por ano.

**Examples**

```
dados <- structure(
  list(data = structure(c(17928, 20485, 19997), class = "Date"),
    ano = c(2019, 2026, 2024),
    mes = c(2, 2, 10),
    nome_4md = c("EDP ES", "EQUATORIAL AL", "EFLJC"),
    subsistema = c("SE", "NE", "S"),
    uf = c("ES", "AL", "SC"),
    segmento = c("comercial_at", "comercial_at_remoto", "residencial"),
    fonte_resumo = c("Termelétrica", "Termelétrica", "Fotovoltaica"),
    energia_mwh = c(103.488, 624.740392156863, 41.7707091331081),
    energia_autoc_mwh = c(51.744, 0, 16.7082836532433),
    energia_inj_mwh = c(51.744, 624.740392156863, 25.0624254798649),
    energia_mwmed = c(Feb = 0.154, Feb = 0.929673202614379, Oct = 0.0561434262541776),
    pot_mes_mw = c(0, 0, 0.0118333352680657),
    adotantes_mes = c(0, 0, 3),
    p = c(0.000329161453402278, 0.000197086794290392, 0.00025238142355137),
    q = c(1, 1, 0.676434758626127),
    regioao = c("SE", "NE", "S")),
  row.names = c(NA, -3L),
  class = c("tbl_df", "tbl", "data.frame")
)

graf_geracao_ano <- epe4md_graf_geracao_ano(
  dados = dados,
  ano_inicio = 2021,
  cor = "#953735",
  tamanho = 14
)
```

---

epe4md\_graf\_geracao\_mes

*Gráfico da geração mensal em MWmed*

---

**Description**

Gráfico de área que mostra a soma da geração de energia em MWmed agrupada por mês.

**Usage**

```
epe4md_graf_geracao_mes(
  dados,
  ano_inicio = 2013,
  cor = "#953735",
  tamanho = 14
)
```

**Arguments**

<code>dados</code>	data.frame. Resultados mensais de potência e energia.
<code>ano_inicio</code>	numeric. Define o ano em que se inicia o eixo x. Default igual a 2013.
<code>cor</code>	string. Escolha da cor do gráfico. Default igual a Vermelho.
<code>tamanho</code>	numeric. Altera o tamanho da fonte da legenda. Default igual a 14.

**Value**

Plota um gráfico de área que mostra a soma da geração de energia em MWmed agrupada por mês.

**Examples**

```
dados <- structure(
  list(data = structure(c(17928, 20485, 19997), class = "Date"),
    ano = c(2019, 2026, 2024),
    mes = c(2, 2, 10),
    nome_4md = c("EDP ES", "EQUATORIAL AL", "EFLJC"),
    subsistema = c("SE", "NE", "S"),
    uf = c("ES", "AL", "SC"),
    segmento = c("comercial_at", "comercial_at_remoto", "residencial"),
    fonte_resumo = c("Termelétrica", "Termelétrica", "Fotovoltaica"),
    energia_mwh = c(103.488, 624.740392156863, 41.7707091331081),
    energia_autoc_mwh = c(51.744, 0, 16.7082836532433),
    energia_inj_mwh = c(51.744, 624.740392156863, 25.0624254798649),
    energia_mwmed = c(Feb = 0.154, Feb = 0.929673202614379, Oct = 0.0561434262541776),
    pot_mes_mw = c(0, 0, 0.0118333352680657),
    adotantes_mes = c(0, 0, 3),
    p = c(0.000329161453402278, 0.000197086794290392, 0.00025238142355137),
    q = c(1, 1, 0.676434758626127),
    regioao = c("SE", "NE", "S")),
  row.names = c(NA, -3L),
  class = c("tbl_df", "tbl", "data.frame")
)

graf_geracao_mes <- epe4md_graf_geracao_mes(
  dados = dados,
  ano_inicio = 2021,
  cor = "#953735",
  tamanho = 14
)
```

---

epe4md\_graf\_part\_fonte\_geracao

*Gráfico da participação de cada tecnologia na geração de energia*


---

### Description

Gráfico de barras que mostra a participação de cada fonte ('Fotovoltaica', 'Termelétrica', 'Hidro', 'Eólica') na geração de energia somente no parâmetro ano\_max\_resultado passado na função epe4md::epe4md\_calcula.

### Usage

```
epe4md_graf_part_fonte_geracao(dados, cor = "#953735", tamanho = 14)
```

### Arguments

dados	data.frame. Resultados mensais de potência e energia.
cor	string. Escolha da cor do gráfico. Default igual a Vermelho.
tamanho	numeric. Altera o tamanho da fonte da legenda. Default igual a 14.

### Value

plota um gráfico de barras que mostra a participação de cada fonte ('Fotovoltaica', 'Termelétrica', 'Hidro', 'Eólica') na geração de energia somente no parâmetro ano\_max\_resultado passado na função epe4md::epe4md\_calcula.

### Examples

```
dados <- structure(
  list(data = structure(c(17928, 20485, 19997), class = "Date"),
    ano = c(2019, 2026, 2024),
    mes = c(2, 2, 10),
    nome_4md = c("EDP ES", "EQUATORIAL AL", "EFLJC"),
    subsistema = c("SE", "NE", "S"),
    uf = c("ES", "AL", "SC"),
    segmento = c("comercial_at", "comercial_at_remoto", "residencial"),
    fonte_resumo = c("Termelétrica", "Termelétrica", "Fotovoltaica"),
    energia_mwh = c(103.488, 624.740392156863, 41.7707091331081),
    energia_autoc_mwh = c(51.744, 0, 16.7082836532433),
    energia_inj_mwh = c(51.744, 624.740392156863, 25.0624254798649),
    energia_mwmed = c(Feb = 0.154, Feb = 0.929673202614379, Oct = 0.0561434262541776),
    pot_mes_mw = c(0, 0, 0.0118333352680657),
    adotantes_mes = c(0, 0, 3),
    p = c(0.000329161453402278, 0.000197086794290392, 0.00025238142355137),
    q = c(1, 1, 0.676434758626127),
    regioao = c("SE", "NE", "S")),
  row.names = c(NA, -3L),
  class = c("tbl_df", "tbl", "data.frame"))
```

```

)

graf_part_fonte_geracao <- epe4md_graf_part_fonte_geracao(
  dados = dados,
  cor = "#953735",
  tamanho = 14
)

```

---

epe4md\_graf\_part\_fonte\_potencia

*Gráfico da participação de capacidade por fonte*

---

### Description

Gráfico de barras que mostra a participação de cada fonte ('Fotovoltaica', 'Termelétrica', 'Hidro' e 'Eólica') na capacidade instalada.

### Usage

```

epe4md_graf_part_fonte_potencia(
  dados,
  cor = "#112446",
  tamanho = 14
)

```

### Arguments

dados	data.frame. Resultados mensais de potência e energia.
cor	string. Escolha da cor do gráfico.
tamanho	numeric. Altera o tamanho da fonte da legenda. Default igual a 14.

### Value

Plota um gráfico de barras que mostra a participação de cada fonte ('Fotovoltaica', 'Termelétrica', 'Hidro' e 'Eólica') na capacidade instalada.

### Examples

```

dados <- structure(
  list(data = structure(c(17928, 20485, 19997), class = "Date"),
    ano = c(2019, 2026, 2024),
    mes = c(2, 2, 10),
    nome_4md = c("EDP ES", "EQUATORIAL AL", "EFLJC"),
    subsistema = c("SE", "NE", "S"),
    uf = c("ES", "AL", "SC"),
    segmento = c("comercial_at", "comercial_at_remoto", "residencial"),
    fonte_resumo = c("Termelétrica", "Termelétrica", "Fotovoltaica"),
    energia_mwh = c(103.488, 624.740392156863, 41.7707091331081),

```

```

energia_autoc_mwh = c(51.744, 0, 16.7082836532433),
energia_inj_mwh = c(51.744, 624.740392156863, 25.0624254798649),
energia_mwmed = c(Feb = 0.154, Feb = 0.929673202614379, Oct = 0.0561434262541776),
pot_mes_mw = c(0, 0, 0.0118333352680657),
adotantes_mes = c(0, 0, 3),
p = c(0.000329161453402278, 0.000197086794290392, 0.00025238142355137),
q = c(1, 1, 0.676434758626127),
regiao = c("SE", "NE", "S")),
row.names = c(NA, -3L),
class = c("tbl_df", "tbl", "data.frame")
)

graf_part_fonte_potencia <- epe4md_graf_part_fonte_potencia(
  dados = dados,
  cor = "#112446",
  tamanho = 14
)

```

---

epe4md\_graf\_part\_segmento

*Gráfico da evolução da participação por segmento*

---

## Description

Gráfico de colunas que mostra a participação de cada segmento 'Comercial (AT)', 'Comercial(BT)', 'Comercial Remoto (AT/BT)', 'Residencial' e 'Residencial Remoto' na geração de energia em cada ano.

## Usage

```
epe4md_graf_part_segmento(dados, ano_inicio = 2013, tamanho = 14)
```

## Arguments

dados	data.frame. Resultados mensais de potência e energia.
ano_inicio	numeric. Define o ano em que se inicia o eixo x. Default igual a 2013.
tamanho	numeric. Altera o tamanho da fonte da legenda. Default igual a 14.

## Value

Plota um gráfico de colunas que mostra a participação de cada segmento 'Comercial (AT)', 'Comercial(BT)', 'Comercial Remoto (AT/BT)', 'Residencial' e 'Residencial Remoto' na geração de energia em cada ano.

**Examples**

```

dados <- structure(
  list(
    data = structure(c(18932, 17956, 18201), class = "Date"),
    ano = c(2021, 2019, 2019),
    mes = c(11, 3, 11),
    nome_4md = c("RORAIMA", "RORAIMA", "RORAIMA"),
    subsistema = c("MAN", "MAN", "MAN"),
    uf = c("RR", "RR", "RR"),
    segmento = c("comercial_at", "comercial_at", "comercial_at"),
    fonte_resumo = c("Fotovoltaica", "Fotovoltaica", "Fotovoltaica"),
    energia_mwh = c(627.959044043851, 42.0777734281515, 42.5086240693568),
    energia_autoc_mwh = c(502.36723523508, 33.6622187425212, 34.0068992554854),
    energia_inj_mwh = c(125.59180880877, 8.41555468563031, 8.50172481387135),
    energia_mwmed = c(Nov = 0.872165338949792, Mar = 0.0565561470808488,
                      Nov = 0.0590397556518844),
    pot_mes_mw = c(0.01199, 0, 0),
    adotantes_mes = c(1, 0, 0),
    p = c(0.000122298434419402, 0.000122298434419402, 0.000122298434419402),
    q = c(1, 1, 1),
    regioao = c("N", "N", "N")),
  row.names = c(NA, -3L),
  class = c("tbl_df", "tbl", "data.frame")
)

graf_part_segmento <- epe4md_graf_part_segmento(
  dados = dados,
  ano_inicio = 2021,
  tamanho = 14
)

```

---

epe4md\_graf\_pot\_acum *Gráfico da capacidade instalada acumulada*

---

**Description**

Mostra a soma acumulada das potências entre os parâmetros ano\_inicio e ano\_max\_resultado passado na função epe4md::epe4md\_calcula.

**Usage**

```
epe4md_graf_pot_acum(dados, ano_inicio = 2013, cor = "#13475d", tamanho = 14)
```

**Arguments**

dados	data.frame. Resultados mensais de potência e energia.
ano_inicio	numeric. Define o ano em que se inicia o eixo x. Default igual a 2013.
cor	string. Escolha da cor do gráfico. Default igual a Azul Escuro.
tamanho	numeric. Altera o tamanho da fonte da legenda. Default igual a 14.

**Value**

Plota um gráfico que mostra a soma acumulada das potências entre os parâmetros `ano_inicio` e `ano_max_resultado` passado na função `epe4md::epe4md_calcula`.

**Examples**

```
dados <- structure(
  list(data = structure(c(17928, 20485, 19997), class = "Date"),
    ano = c(2019, 2026, 2024),
    mes = c(2, 2, 10),
    nome_4md = c("EDP ES", "EQUATORIAL AL", "EFLJC"),
    subsistema = c("SE", "NE", "S"),
    uf = c("ES", "AL", "SC"),
    segmento = c("comercial_at", "comercial_at_remoto", "residencial"),
    fonte_resumo = c("Termelétrica", "Termelétrica", "Fotovoltaica"),
    energia_mwh = c(103.488, 624.740392156863, 41.7707091331081),
    energia_autoc_mwh = c(51.744, 0, 16.7082836532433),
    energia_inj_mwh = c(51.744, 624.740392156863, 25.0624254798649),
    energia_mwmed = c(Feb = 0.154, Feb = 0.929673202614379, Oct = 0.0561434262541776),
    pot_mes_mw = c(0, 0, 0.0118333352680657),
    adotantes_mes = c(0, 0, 3),
    p = c(0.000329161453402278, 0.000197086794290392, 0.00025238142355137),
    q = c(1, 1, 0.676434758626127),
    regioao = c("SE", "NE", "S")),
  row.names = c(NA, -3L),
  class = c("tbl_df", "tbl", "data.frame")
)

graf_pot_acum <- epe4md_graf_pot_acum(
  dados = dados,
  ano_inicio = 2021,
  cor = "#13475d",
  tamanho = 14
)
```

---

`epe4md_graf_pot_anual` *Gráfico da capacidade instalada anual*

---

**Description**

Gráfico de colunas que mostra a soma simples da potência em cada ano, entre os parâmetros `ano_inicio` e `ano_max_resultado` passado na função `epe4md::epe4md_calcula`.

**Usage**

```
epe4md_graf_pot_anual(dados, ano_inicio = 2013, cor = "#13475d", tamanho = 14)
```

**Arguments**

dados	data.frame. Resultados mensais de potência e energia.
ano_inicio	numeric. Define o ano em que se inicia o eixo x. Default igual a 2013.
cor	string. Escolha da cor do gráfico. Default igual a Azul Escuro.
tamanho	numeric. Altera o tamanho da fonte da legenda. Default igual a 14.

**Value**

Plota um gráfico de colunas que mostra a soma simples da potência em cada ano, entre os parâmetros ano\_inicio e ano\_max\_resultado passado na função epe4md::epe4md\_calcula.

**Examples**

```
dados <- structure(
  list(data = structure(c(17928, 20485, 19997), class = "Date"),
    ano = c(2019, 2026, 2024),
    mes = c(2, 2, 10),
    nome_4md = c("EDP ES", "EQUATORIAL AL", "EFLJC"),
    subsistema = c("SE", "NE", "S"),
    uf = c("ES", "AL", "SC"),
    segmento = c("comercial_at", "comercial_at_remoto", "residencial"),
    fonte_resumo = c("Termelétrica", "Termelétrica", "Fotovoltaica"),
    energia_mwh = c(103.488, 624.740392156863, 41.7707091331081),
    energia_autoc_mwh = c(51.744, 0, 16.7082836532433),
    energia_inj_mwh = c(51.744, 624.740392156863, 25.0624254798649),
    energia_mwmed = c(Feb = 0.154, Feb = 0.929673202614379, Oct = 0.0561434262541776),
    pot_mes_mw = c(0, 0, 0.0118333352680657),
    adotantes_mes = c(0, 0, 3),
    p = c(0.000329161453402278, 0.000197086794290392, 0.00025238142355137),
    q = c(1, 1, 0.676434758626127),
    regioao = c("SE", "NE", "S")),
  row.names = c(NA, -3L),
  class = c("tbl_df", "tbl", "data.frame")
)

graf_pot_anual <- epe4md_graf_pot_anual(
  dados = dados,
  ano_inicio = 2021,
  cor = "#13475d",
  tamanho = 14
)
```

---

epe4md\_graf\_pot\_regiao

*Gráfico da capacidade instalada acumulada por região*

---

**Description**

Gráfico da capacidade instalada acumulada por região.

**Usage**

```
epe4md_graf_pot_regiao(dados, ano_inicio = 2013, tamanho = 14)
```

**Arguments**

`dados` data.frame. Resultados mensais de potência e energia.

`ano_inicio` numeric. Define o ano em que se inicia o eixo x. Default igual a 2013.

`tamanho` numeric. Altera tamanho da fonte da legenda. Default igual a 14.

**Value**

Plota um gráfico da capacidade instalada acumulada por região.

**Examples**

```
dados <- structure(
  list(
    data = structure(c(16587, 16953, 17348), class = "Date"),
    ano = c(2015, 2016, 2017),
    mes = c(6, 6, 7),
    nome_4md = c("EQUATORIAL PA", "AME", "AME"),
    subsistema = c("N", "MAN", "MAN"),
    uf = c("PA", "AM", "AM"),
    segmento = c("comercial_at", "comercial_at", "residencial"),
    fonte_resumo = c("Fotovoltaica", "Fotovoltaica", "Fotovoltaica"),
    energia_mwh = c(2.91887385534686, 1.47730302655558, 6.59899395709866),
    energia_autoc_mwh = c(2.33509908427749, 1.18184242124447, 2.63959758283946),
    energia_inj_mwh = c(0.583774771069372, 0.295460605311116, 3.9593963742592),
    energia_mwmed = c(Jun = 0.00405399146575953, Jun = 0.00205180975910497,
                      Jul = 0.00886961553373476),
    pot_mes_mw = c(0, 0, 0),
    adotantes_mes = c(0, 0, 0),
    p = c(0.000218228349678747, 0.000336532886429228, 0.000109941017035186),
    q = c(1, 0.677467408094237, 0.68352139010882),
    regioao = c("N", "N", "N")),
  row.names = c(NA, -3L),
  class = c("tbl_df", "tbl", "data.frame")
)

graf_pot_regiao <- epe4md_graf_pot_regiao(
  dados = dados,
  ano_inicio = 2013,
  tamanho = 14
)
```

---

epe4md\_graf\_pot\_segmento

*Gráfico da capacidade instalada acumulada por segmento*

---

### Description

Mostra a soma acumulada da potência entre o intervalo de anos solicitado, agrupada entre os segmentos : ' Comercial (AT) ', ' Comercial (BT) ', ' Comercial Remoto (AT/BT) ', ' Residencial ' e ' Residencial Remoto '.

### Usage

```
epe4md_graf_pot_segmento(dados, ano_inicio = 2013, tamanho = 14)
```

### Arguments

dados	data.frame. Resultados mensais de potência e energia.
ano_inicio	numeric. Define o ano em que se inicia o eixo x. Default igual a 2013.
tamanho	numeric. Altera tamanho da fonte da legenda. Default igual a 14.

### Value

Plota um gráfico que mostra a soma acumulada da potência entre o intervalo de anos solicitado, agrupada entre os segmentos : ' Comercial (AT) ', ' Comercial (BT) ', ' Comercial Remoto (AT/BT) ', ' Residencial ' e ' Residencial Remoto '.

### Examples

```
dados <- structure(
  list(data = structure(c(16953, 17167, 17532), class = "Date"),
    ano = c(2016, 2017, 2018),
    mes = c(6, 1, 1),
    nome_4md = c("RORAIMA", "RORAIMA", "RORAIMA"),
    subsistema = c("MAN", "MAN", "MAN"),
    uf = c("RR", "RR", "RR"),
    segmento = c("residencial", "residencial", "comercial_at"),
    fonte_resumo = c("Fotovoltaica", "Fotovoltaica", "Fotovoltaica"),
    energia_mwh = c(0.14901698211027, 0.31989258307487, 21.4846247221392),
    energia_autoc_mwh = c(0.0596067928441078, 0.127957033229948,
      17.1876997777114),
    energia_inj_mwh = c(0.0894101892661617, 0.191935549844922,
      4.29692494442784),
    energia_mwmed = c(Jun = 0.000206968030708708, Jan = 0.00042996314929418,
      Jan = 0.0288771837663161),
    pot_mes_mw = c(0.00297, 0, 0),
    adotantes_mes = c(1, 0, 0),
    p = c(1e-04, 1e-04, 0.000122345127598721),
    q = c(0.779421829099793, 0.779421829099793, 1),
```

```
      regioao = c("N", "N", "N")),
    row.names = c(NA, -3L),
    class = c("tbl_df", "tbl", "data.frame")
  )

graf_pot_segmento <- epe4md_graf_pot_segmento(
  dados = dados,
  ano_inicio = 2013,
  tamanho = 14
)
```

---

epe4md\_investimentos *Calcula o montante investido*

---

## Description

Calcula o montante investido

## Usage

```
epe4md_investimentos(
  resultados_mensais,
  ano_base,
  ano_max_resultado = 2050,
  dir_dados_premissas = NA_character_
)
```

## Arguments

**ano\_base** numeric. Ano base da projeção. Define o ano em que a função irá buscar a base de dados. Último ano completo realizado.

**ano\_max\_resultado** numeric. Ano final para apresentação dos resultados. Máximo igual a 2050. Default igual a 2050.

**dir\_dados\_premissas** Diretório onde se encontram as premissas. Se esse parâmetro não for passado, a função usa os dados default que são instalados com o pacote. É importante que os nomes dos arquivos sejam os mesmos da pasta default.

**resultados\_mensais** Resultado da função [epe4md\\_calcula](#).

## Value

data.frame com o montante estimado de investimentos relativos à expansão da capacidade instalada de micro e minigeração distribuída.

**Examples**

```

resultados_mensais <- tibble::tibble(
  data = c('2021-01-01', '2021-01-01', '2021-01-01'),
  ano = 2021,
  nome_4md = c("ENEL RJ", "ENF", "LIGHT"),
  sistema = "SE",
  uf = "RJ",
  segmento = "comercial_at",
  fonte_resumo = "Fotovoltaica",
  energica_mwh = c(2019.6225, 69.06433, 2303.18754),
  energica_autoc_mwh = c(1615.698, 55.25146, 1842.55003),
  energia_inj_mwh = c(403.92450, 13.81287, 460.63751),
  energia_mwmed = c(2.714546380, 0.092828390, 3.095682180),
  pot_mes_mw = c(1.5275, 0.0000, 3.09568218),
  adotantes_mes = c(21, 0, 4),
  p = c(0.00027826110, 0.00060785180, 0.00137570060),
  q = c(1.0000000, 1.0000000, 0.5282319)
)

investimentos <- epe4md_investimentos(
  resultados_mensais = resultados_mensais,
  ano_base = 2021,
  ano_max_resultado = 2050,
  dir_dados_premissas = NA_character_
)

```

---

epe4md\_mercado\_potencial

*Cria a base do mercado potencial inicial para a adoção.*

---

**Description**

Cria a base do mercado potencial inicial para a adoção.

**Usage**

```

epe4md_mercado_potencial(
  ano_base,
  filtro_renda_domicilio = "maior_3sm",
  filtro_comercial = NA,
  tx_cresc_grupo_a = 0.016,
  dir_dados_premissas = NA_character_
)

```

**Arguments**

**ano\_base** numeric. Ano base da projeção. Define o ano em que a função irá buscar a base de dados. Último ano completo realizado.

`filtro_renda_domicilio`  
string. Define o filtro aplicado a consumidores residenciais, de acordo com a renda mensal do responsável, em salários mínimos. Permite: "total", "maior\_1sm", "maior\_2sm", "maior\_3sm" ou "maior\_5sm". Default igual a "maior\_3sm".

`filtro_comercial`  
numeric. Fator percentual para definir o nicho do segmento comercial. Default é calculado pelo modelo com base no nicho residencial.

`tx_cresc_grupo_a`  
numeric. Taxa de crescimento anual dos consumidores cativos do Grupo A. Default igual a 0.016 representa crescimento entre 2006 e 2019.

`dir_dados_premissas`  
Diretório onde se encontram as premissas. Se esse parâmetro não for passado, a função usa os dados default que são instalados com o pacote. É importante que os nomes dos arquivos sejam os mesmos da pasta default.

### Value

list com dois data.frames. "consumidores" possui o mercado potencial inicial. "consumidores\_totais" possui dados de mercado total.

### Examples

```
mercado_potencial <- epe4md_mercado_potencial(  
  ano_base = 2021,  
  filtro_renda_domicilio = "maior_3sm",  
  filtro_comercial = NA,  
  tx_cresc_grupo_a = 0.016,  
  dir_dados_premissas = NA_character_  
)
```

---

epe4md\_payback

*Roda um fluxo de caixa para cada caso e retorna métricas financeiras.*

---

### Description

Roda um fluxo de caixa para cada caso e retorna métricas financeiras.

### Usage

```
epe4md_payback(  
  casos_payback,  
  premissas_reg,  
  ano_base,  
  sequencial,  
  filtro_de_uf,  
  filtro_de_segmento,  
  filtro_de_custo_unitario_max,  
  altera_sistemas_existentes = TRUE,
```

```

ano_decisao_alteracao = 2023,
inflacao = 0.0375,
taxa_desconto_nominal = 0.13,
custo_reforco_rede = 200,
ano_troca_inversor = 11,
pagamento_disponibilidade = 0.3,
disponibilidade_kwh_mes = 100,
desconto_capex_local = 0,
anos_desconto = 0,
dir_dados_premissas = NA_character_
)

```

### Arguments

<code>casos_payback</code>	data.frame. Base gerada pela função <code>epe4md_casos_payback</code>
<code>premissas_reg</code>	data.frame. Input de premissas regulatórias para serem consideradas nos cálculos.
<code>ano_base</code>	numeric. Ano base da projeção. Define o ano em que a função irá buscar a base de dados. Último ano completo realizado.
<code>sequencial</code>	logic. Parâmetro que define se a projeção deve ser realizada de forma sequencial ou paralela. Para executar a projeção de forma sequencial defina o parâmetro como TRUE. Para executar a projeção de forma paralela, defina o parâmetro como FALSE.
<code>filtro_de_uf</code>	string. Parâmetro que define uma unidade federativa (UF) a ser filtrada. Caso uma UF não seja indicada ou seja informado um valor inválido, o resultado será apresentado sem filtros.
<code>filtro_de_segmento</code>	string. Parâmetro que define um segmento a ser filtrado. Pode se escolher entre "comercial_at", "comercial_at_remoto", "comercial_bt", "residencial" e "residencial_remoto". Caso não seja informado um valor ou seja informado um valor inválido o resultado será apresentado sem filtro.
<code>filtro_de_custo_unitario_max</code>	numeric. Parâmetro que define o valor máximo do custo unitário para ser utilizado no cálculo do payback. Default igual a NULL.
<code>altera_sistemas_existentes</code>	logic. TRUE se alterações regulatórias afetam investimentos realizados em anos anteriores à revisão da regulação. Default igual a FALSE.
<code>ano_decisao_alteracao</code>	numeric. Ano em que são definidas novas regras e se tornam de conhecimento público. Esse parâmetro só tem efeito caso o anterior seja igual a TRUE. Default igual a 2021.
<code>inflacao</code>	numeric. Taxa anual de inflacao considerada no reajuste das tarifas e para calcular o retorno real de projetos. Default igual a 0.0375.
<code>taxa_desconto_nominal</code>	numeric. Taxa de desconto nominal considerada nos cálculos de payback descontado. Default igual a 0.13.

custo_reforco_rede	numeric. Custo em R\$/kW aplicado a projetos de geracao remota em Alta Tensão. Representa um custo pago pelo empreendedor para reforços na rede. Default igual a 200.
ano_troca_inversor	numeric. Ano, a partir do ano de instalação, em que é realizada a troca do inversor fotovoltaico. Default igual a 11.
pagamento_disponibilidade	numeric. Percentual de meses em que o consumidor residencial paga custo de disponibilidade em função da variabilidade da geração FV. Default igual a 0.3. Tem efeito somente até o ano de 2022.
disponibilidade_kwh_mes	numeric. Consumo de disponibilidade do consumidor em kWh/mês. Default igual a 100, equivalente a um consumidor trifásico. Tem efeito somente até o ano de 2022.
desconto_capex_local	numeric. Percentual de desconto a ser aplicado no CAPEX de sistemas de geração local(ex: 0.1) para simulação de incentivos. Default igual a 0.
anos_desconto	vector. Anos em que há a incidência do desconto no CAPEX. Default igual a 0.
dir_dados_premissas	Diretório onde se encontram as premissas. Se esse parâmetro não for passado, a função usa os dados default que são instalados com o pacote. É importante que os nomes dos arquivos sejam os mesmos da pasta default.

## Value

data.frame. Métricas financeiras para cada caso.

## Examples

```
casos_payback <- tibble::tibble(
  segmento = "comercial_at",
  fonte_resumo = "Fotovoltaica",
  fator_autoconsumo = 0.8,
  oem_anual = 0.01,
  nome_4md = c("OUTRA", "RGE", "RORAIMA", "SULGIPE", "UHENPAL"),
  fc = c(0.1633388, 0.1513311, 0.1505592, 0.1649651, 0.1481121),
  vida_util = 25,
  degradacao = 0.005,
  pot_sistemas = 70,
  geracao_1_kwh = c(100159.36, 92796.22, 92322.91, 101156.62, 90822.36),
  ano = 2021,
  custo_unitario = 3.81,
  custo_inversor = 0.5715,
  capex_inicial = 266700,
  capex_inversor = 57808.98
)

premissas_regulatorias <- tibble::tibble(
  ano = 2021,
```

```

alternativa = 0,
p_transicao = 1.00,
binomia = FALSE,
demanda_g = FALSE
)

payback <- epe4md_payback(
  casos_payback = casos_payback,
  premissas_reg = premissas_regulatorias,
  ano_base = 2021,
  sequencial = TRUE,
  filtro_de_uf = "RR",
  filtro_de_segimento = "comercial_at",
  filtro_de_custo_unitario_max = 6,
  altera_sistemas_existentes = TRUE,
  ano_decisao_alteracao = 2023,
  inflacao = 0.0375,
  taxa_desconto_nominal = 0.13,
  custo_reforco_rede = 200,
  ano_troca_inversor = 11,
  pagamento_disponibilidade = 0.3,
  disponibilidade_kwh_mes = 100,
  desconto_capex_local = 0,
  anos_desconto = 0,
  dir_dados_premissas = NA_character_
)

```

---

epe4md\_prepara\_base *Prepara a base de dados da ANEEL com geradores existentes para ser utilizada nas funções seguintes.*

---

### Description

Prepara a base de dados da ANEEL com geradores existentes para ser utilizada nas funções seguintes.

### Usage

```

epe4md_prepara_base(
  base_aneel,
  ano_base,
  resumida = TRUE,
  dir_dados_premissas = NA_character_
)

```

### Arguments

ano\_base            numeric. Ano base da projeção. Define o ano em que a função irá buscar a base de dados. Último ano completo realizado.

dir_dados_premissas	Diretório onde se encontram as premissas. Se esse parâmetro não for passado, a função usa os dados default que são instalados com o pacote. É importante que os nomes dos arquivos sejam os mesmos da pasta default.
base_aneel	Dataframe com a base de dados disponibilizada pela ANEEL na sua página de dados abertos com dados individualizados dos micro e minigeradores distribuídos. Sugestão ler base com read.csv2
resumida	Logic. Se TRUE, retorna a base resumida (Default). Se FALSE retorna base com mais desagregações.

### Value

data.frame. Base tratada e resumida.

### Examples

```
base <- tibble::tibble(
  DatGeracaoConjuntoDados = "2023-03-22",
  AnmPeriodoReferencia = "03/2023",
  NumCNPJDistribuidora = c(3.467321e+12, 8.644416e+13, 2.508603e+13),
  SigAgente = c("EMT", "CEGERO", "ETO"),
  NomAgente = c("Energisa Mato Grosso - Distribuidora de Energia S.A.",
    "COOPERATIVA DE ELETRICIDADE DE S<c3>O LUDGERO",
    "ENERGISA TOCANTINS DISTRIBUIDORA DE ENERGIA S.A."),
  CodClasseConsumo = c(1, 3, 1),
  DscClasseConsumo = c("Residencial", "Rural", "Residencial"),
  CodSubGrupoTarifario = c(9, 10, 9),
  DscSubGrupoTarifario = c("B1", "B2", "B1"),
  codUFibge = c(51, 42, 17),
  SigUF = c("MT", "SC", "TO"),
  codRegiao = c(5102, 4206, 1702),
  NomRegiao = c("Centro Oeste", "Sul", "Norte"),
  CodMunicipioIbge = c(5101803, 4211702, 1721000),
  NomMunicipio = c("Barra do Gar<e7>as", "Orleans", "Palmas"),
  CodCEP = c("78600***", "88870***", "77000***"),
  SigTipoConsumidor = "PF",
  NumCPFNPJ = c("***.529.008-**", "***.075.129-**", "***.730.731-**"),
  NomeTitularEmpreendimento = "***",
  CodEmpreendimento = c("GD.MT.000.001.703", "GD.SC.000.051.177", "GD.TO.000.000.763"),
  DthAtualizaCadastralEmpreend = c("2016-04-08", "2018-12-04", "2016-09-29"),
  SigModalidadeEmpreendimento = c("R", "P", "P"),
  DscModalidadeHabilitado = c("Caracterizada como Autoconsumo remoto",
    "Com Microgeracao ou Minigeracao distribuida",
    "Com Microgeracao ou Minigeracao distribuida"),
  QtdUCRecebeCredito = c(2, 1, 1),
  SigTipoGeracao = "UFV",
  DscFonteGeracao = "Radia<e7><e3>o solar",
  DscPorte = "Microgeracao",
  MdaPotenciaInstaladaKW = c(3.00, 5.00, 4.00),
  NumCoordNEmpreendimento = NA,
  NumCoordEEmpreendimento = NA,
  NomSubEstacao = NA,
```

```
    NumCoordESub = NA,  
    NumCoordNSub = NA  
  )  
  
  base <- epe4md_prepara_base(  
    base_aneel = base,  
    ano_base = 2021,  
    resumida = TRUE,  
    dir_dados_premissas = NA_character_  
  )
```

---

epe4md\_proj\_adotantes *Realiza a projeção do número de adotantes de micro e minigeração distribuída*

---

### Description

Realiza a projeção do número de adotantes de micro e minigeração distribuída

### Usage

```
epe4md_proj_adotantes(  
  casos_otimizados,  
  consumidores,  
  ano_base,  
  dir_dados_premissas = NA_character_  
)
```

### Arguments

**casos\_otimizados** data.frame. Resultado da função [epe4md\\_calibra\\_curva\\_s](#).

**consumidores** list. Resultado da função [epe4md\\_mercado\\_potencial](#).

**ano\_base** numeric. Ano base da projeção. Define o ano em que a função irá buscar a base de dados. Último ano completo realizado.

**dir\_dados\_premissas** Diretório onde se encontram as premissas. Se esse parâmetro não for passado, a função usa os dados default que são instalados com o pacote. É importante que os nomes dos arquivos sejam os mesmos da pasta default.

### Value

list com dois data.frames. "proj\_adotantes" possui os resultados da projeção de adotantes de micro e minigeração distribuída. "part\_adotantes" possui o resultado em termos de participação do número de adotantes frente ao total de unidades consumidoras.

**Examples**

```

casos_otimizados <-
  structure(
    list(nome_4md = c("MUXENERGIA", "EPB", "EFLJC"),
          segmento = c("comercial_bt", "comercial_at", "residencial"),
          p = c(0.01, 0.000316252847258369, 0.00025238142355137),
          q = c(1, 1, 0.676434758626127),
          spb = c(0.3, 0.3, 0.3),
          ano = c(2029, 2025, 2026),
          Ft = c(0.999996472365327, 0.992930172901814, 0.829136572246132),
          consumidores = c(257, 1871, 847),
          payback = c(4.38236607048146, 7.216158855616, 4.25745937192023),
          mercado_potencial = c(69, 215, 236)),
    row.names = c(NA, -3L),
    class = c("tbl_df", "tbl", "data.frame"))

consumidores <-
  list(
    structure(
      list(nome_4md = c("CPFL PIRATININGA", "EMS", "RGE"),
            ano = c(2020, 2028, 2034),
            segmento = c("comercial_at_remoto", "residencial", "comercial_at"),
            consumidores = c(6736, 113412, 13515)),
      class = c("tbl_df", "tbl", "data.frame"),
      row.names = c(NA, -3L)),
    structure(
      list(ano = c(2017, 2035, 2031),
            total_ucs = c(67352162.358959, 88092914.3204848, 83828643.0017198),
            segmento = c("residencial", "residencial", "residencial")),
      row.names = c(NA, -3L),
      class = c("tbl_df", "tbl", "data.frame"))
  )

names(consumidores) <- c("consumidores", "consumidores_totais")

proj_adotantes <- epe4md_proj_adotantes(
  casos_otimizados = casos_otimizados,
  consumidores = consumidores,
  ano_base = 2021
)

```

---

epe4md\_proj\_geracao      *Estima a geração de eletricidade a partir da projeção de potência*

---

**Description**

Estima a geração de eletricidade a partir da projeção de potência

**Usage**

```
epe4md_proj_geracao(
  proj_mensal,
  ano_base,
  filtro_de_uf,
  filtro_de_segmento,
  dir_dados_premissas = NA_character_
)
```

**Arguments**

**proj\_mensal** dataframe. Resultado da função [epe4md\\_proj\\_mensal](#).

**ano\_base** numeric. Ano base da projeção. Define o ano em que a função irá buscar a base de dados. Último ano completo realizado.

**filtro\_de\_uf** string. Parâmetro que define uma unidade federativa (UF) a ser filtrada. Caso uma UF não seja indicada ou seja informado um valor inválido, o resultado será apresentado sem filtros.

**filtro\_de\_segmento** string. Parâmetro que define um segmento a ser filtrado. Pode se escolher entre "comercial\_at", "comercial\_at\_remoto", "comercial\_bt", "residencial" e "residencial\_remoto". Caso não seja informado um valor ou seja informado um valor inválido o resultado será apresentado sem filtro.

**dir\_dados\_premissas** Diretório onde se encontram as premissas. Se esse parâmetro não for passado, a função usa os dados default que são instalados com o pacote. É importante que os nomes dos arquivos sejam os mesmos da pasta default.

**Value**

data.frame com os resultados da projeção de capacidade instalada de micro e minigeração distribuída, número de adotantes e geração mensal de energia.

**Examples**

```
projecao_mensal <- structure(
  list(ano = c(2018, 2019, 2020),
       nome_4md = c("COSERN", "EQUATORIAL AL", "ERO"),
       fonte_resumo = c("Fotovoltaica", "Fotovoltaica", "Fotovoltaica"),
       segmento = c("residencial", "comercial_bt", "residencial_remoto"),
       adotantes_mes = c(29, 38, 4),
       pot_mes_mw = c(0.16773, 0.42818, 0.01418),
       mes_ano = c("2018 jul 1", "2019 jun 1", "2020 ago 1"),
       mes = c(7, 6, 8),
       p = c(1e-04, 0.000321082460336051, 1e-04),
       q = c(0.999129117163193, 1, 0.788982959999952),
       Ft = c(0.0385425368844212, 0.260610809377481, 0.0652161787454779)),
  row.names = c(NA, -3L),
  class = c("tbl_df", "tbl", "data.frame")
)
```

```
projecao_mensal <- dplyr::mutate(projecao_mensal, mes_ano = strptime(mes_ano, "%Y %b %d"))

proj_geracao <- epe4md_proj_geracao(
  proj_mensal = projecao_mensal,
  ano_base = 2021,
  filtro_de_uf = "N",
  filtro_de_segmento = "comercial_at",
  dir_dados_premissas = NA_character_
)
```

---

epe4md\_proj\_mensal      *Faz a abertura mensal da projeção de potência*

---

### Description

Faz a abertura mensal da projeção de potência

### Usage

```
epe4md_proj_mensal(
  lista_potencia,
  ano_base,
  filtro_nome4md,
  filtro_de_segmento,
  ano_max_resultado = 2050,
  ajuste_ano_corrente = FALSE,
  ultimo_mes_ajuste = NA,
  metodo_ajuste = NA,
  dir_dados_premissas = "inst/dados_premissas"
)
```

### Arguments

**lista\_potencia** list. Resultado da função [epe4md\\_proj\\_potencia](#).

**ano\_base** numeric. Ano base da projeção. Define o ano em que a função irá buscar a base de dados. Último ano completo realizado.

**filtro\_nome4md** string. Parâmetro que define o nome de uma concessionária de energia a ser filtrado. Caso não seja informado um valor ou seja informado um valor inválido o resultado será apresentado sem filtro.

**filtro\_de\_segmento** string. Parâmetro que define um segmento a ser filtrado. Pode se escolher entre "comercial\_at", "comercial\_at\_remoto", "comercial\_bt", "residencial" e "residencial\_remoto". Caso não seja informado um valor ou seja informado um valor inválido o resultado será apresentado sem filtro.

**ano\_max\_resultado** numeric. Ano final para apresentação dos resultados. Máximo igual a 2050. Default igual a 2050.

ajuste_ano_corrente	logic. Se TRUE indica que a projeção deverá incorporar o histórico mensal recente, verificado em parte do primeiro ano após o ano base. Default igual a FALSE. O arquivo base_mmgd.xlsx deve incorporar esse histórico.
ultimo_mes_ajuste	numeric. Último mês com dados completos na base_ano_corrente. Default igual a NA. Só tem efeito caso ajuste_ano_corrente seja igual a TRUE.
metodo_ajuste	string. Se igual a "extrapola" o modelo irá extrapolar a potência e o número de adotantes até o final do ano base + 1 com base no verificado até o ultimo_mes_ajuste. Se igual a "substitui", o modelo substitui a projeção até o ultimo_mes_ajuste e mantém o restante do ano com a projeção normal. Só tem efeito caso ajuste_ano_corrente seja igual a TRUE.
dir_dados_premissas	Diretório onde se encontram as premissas. Se esse parâmetro não for passado, a função usa os dados default que são instalados com o pacote. É importante que os nomes dos arquivos sejam os mesmos da pasta default.

### Value

data.frame com os resultados da projeção de capacidade instalada de micro e minigeração distribuída e número de adotantes mensal

### Examples

```
lista_potencia <- list(
  proj_potencia = structure(
    list(nome_4md = c("RORAIMA", "RORAIMA", "RORAIMA", "RORAIMA", "RORAIMA"),
          segmento = c("comercial_at", "comercial_at", "comercial_at",
                      "comercial_at", "comercial_at"),
          p = c(0.000122298434419402, 0.000122298434419402,
                0.000122298434419402, 0.000122298434419402, 0.000122298434419402),
          q = c(1, 1, 1, 1, 1),
          spb = c(0.3, 0.3, 0.3, 0.3, 0.3),
          ano = c(2017, 2018, 2019, 2020, 2021),
          Ft = c(0.0177177153816275, 0.0469352065606947, 0.118237854106428,
                0.267268529395706, 0.497952210651889),
          consumidores = c(656, 678, 751, 764, 791),
          payback = c(25, 12.9151902647661, 7.76175962915069,
                    11.9207871041042, 11.8969973448089),
          mercado_potencial = c(0.25, 3.5, 18.25, 5.25, 5.5),
          adotantes_ano = c(3, 1, 1, 3, 6),
          adotantes_acum = c(3, 4, 5, 8, 14),
          fonte_resumo = c("Fotovoltaica", "Fotovoltaica", "Fotovoltaica",
                          "Fotovoltaica", "Fotovoltaica"),
          part_fonte = c(1, 1, 1, 1, 1),
          adotantes_hist = c(3, 1, 1, 3, 6),
          pot_media = c(353.694375, 353.694375, 353.694375, 353.694375,
                       353.694375),
          pot_ano = c(199, 80, 94, 4602, 529.11),
          pot_hist = c(199, 80, 94, 4602, 529.11),
          pot_ano_mw = c(0.199, 0.08, 0.094, 4.602, 0.52911),
```

```

        pot_acum_mw = c(0.199, 0.279, 0.373, 4.975, 5.50411)),
class = c("tbl_df", "tbl", "data.frame"),
row.names = c(NA, -5L)
),
part_adotantes = structure(
  list(ano = c(2017, 2018, 2019, 2020, 2021),
        segmento = c("comercial_at", "comercial_at", "comercial_at",
"comercial_at", "comercial_at"),
        adotantes = c(3, 4, 5, 8, 14),
        mercado_potencial = c(1, 14, 73, 21, 22),
        total_ucs = c(187922, 187853, 184276, 182048, 181394),
        penetracao_total = c(1.5964070199338e-05, 2.12932452502755e-05,
2.71332132236428e-05, 4.39444542098787e-05, 7.71800610825055e-05),
        mercado_nicho = c(187922, 187853, 184276, 182048, 181394),
        penetracao_nicho = c(1.5964070199338e-05, 2.12932452502755e-05,
2.71332132236428e-05, 4.39444542098787e-05, 7.71800610825055e-05),
        penetracao_potencial = c(3, 0.285714285714286, 0.0684931506849315,
0.380952380952381, 0.636363636363636)),
  row.names = c(NA, -5L),
  class = c("tbl_df", "tbl", "data.frame")
)
)

proj_mensal <- epe4md_proj_mensal(
  lista_potencia = lista_potencia,
  ano_base = 2021,
  filtro_nome4md = "RORAIMA",
  filtro_de_segmento = "comercial_at",
  ano_max_resultado = 2021
)

```

---

epe4md\_proj\_potencia *Realiza a projecao da capacidade instalada de micro e minigeracao distribuida*

---

### Description

Realiza a projecao da capacidade instalada de micro e minigeracao distribuida

### Usage

```

epe4md_proj_potencia(
  lista_adotantes,
  ano_base,
  dir_dados_premissas = "inst/dados_premissas"
)

```

**Arguments**

`lista_adotantes`  
 data.frame. Data.frame que contém dados relativos aos adotantes.

`ano_base`  
 numeric. Ano base da projeção. Define o ano em que a função irá buscar a base de dados. Último ano completo realizado.

`dir_dados_premissas`  
 Diretório onde se encontram as premissas. Se esse parâmetro não for passado, a função usa os dados default que são instalados com o pacote. É importante que os nomes dos arquivos sejam os mesmos da pasta default.

**Value**

list com dois data.frames. "proj\_potencia" possui os resultados da projeção de capacidade instalada de micro e minigeração distribuída. "part\_adotantes" possui o resultado em termos de participação do número de adotantes frente ao total de unidades consumidoras.

**Examples**

```
lista_adotantes <- list(
  structure(
    list(nome_4md = c("CEB", "EQUATORIAL MA", "LIGHT"),
         segmento = c("comercial_at_remoto", "comercial_bt", "comercial_at_remoto"),
         p = c(0.000314088483901221, 0.000326288160094334, 0.01),
         q = c(0.860602105239079, 0.796618861208451, 0.0771175699573262),
         spb = c(0.3, 0.3, 0.3),
         ano = c(2024, 2028, 2022),
         Ft = c(0.917955265426745, 0.992968921614641, 0.137575817915978),
         consumidores = c(12146, 47955, 22878),
         payback = c(5.00924084667372, 4.40202323859729, 8.53878219211533),
         mercado_potencial = c(675.75, 3200.75, 441.5),
         adotantes_ano = c(309, 0, 4),
         adotantes_acum = c(2361, 0, 42),
         fonte_resumo = c("Fotovoltaica", "Eólica", "Fotovoltaica"),
         part_fonte = c(1, 0, 0.186666666666667),
         adotantes_hist = c(NA_real_, NA_real_, NA_real_)),
    row.names = c(NA, -3L),
    class = c("tbl_df", "tbl", "data.frame")),
  structure(
    list(ano = c(2030, 2016, 2020),
         segmento = c("comercial_bt", "residencial", "residencial"),
         adotantes = c(694432, 7214, 357386),
         mercado_potencial = c(620539, 857786, 2931122),
         total_ucs = c(14858570, 66148593.0147109, 71001451.6735913),
         penetracao_total = c(0.0467361260202025, 0.000109057497238009, 0.00503350271826806),
         mercado_nicho = c(2541888, 10668970, 11014026),
         penetracao_nicho = c(0.273195357151849, 0.000676166490298501, 0.0324482618798975),
         penetracao_potencial = c(1.11907873638885, 0.00841002301273278, 0.121928053489415)),
    row.names = c(NA, -3L),
    class = c("tbl_df", "tbl", "data.frame"))
)
```

```
names(lista_adotantes) <- c("proj_adotantes", "part_adotantes")

proj_potencia <- epe4md_proj_potencia(
  lista_adotantes = lista_adotantes,
  ano_base = 2021
)
```

---

epe4md\_sumariza\_resultados

*Resume os resultados de capacidade instalada*

---

### Description

Resume os resultados de capacidade instalada

### Usage

```
epe4md_sumariza_resultados(resultados_mensais)
```

### Arguments

```
resultados_mensais
  data.frame. Saída da função epe4md::epe4md_calcula
```

### Value

data.frame com projeção de capacidade instalada nacional, em GW e geração de energia, em GWh e MW méd.

### Examples

```
resultados_mensais <- structure(
  list(data = structure(c(18628, 18628, 18628),
    class = "Date"),
    ano = c(2021, 2021, 2021),
    mes = c(1, 1, 1),
    nome_4md = c("RORAIMA", "SULGIPE", "UHENPAL"),
    subsistema = c("MAN", "NE", "S"),
    uf = c("RR", "SE", "RS"),
    segmento = c("comercial_at", "comercial_at", "comercial_bt"),
    fonte_resumo = c("Fotovoltaica", "Fotovoltaica", "Fotovoltaica"),
    energia_mwh = c(536.436102870736, 83.2181016179793, 128.993567632682),
    energia_autoc_mwh = c(429.148882296589, 66.5744812943834, 64.4967838163411),
    energia_inj_mwh = c(107.287220574147, 16.6436203235959, 64.4967838163411),
    energia_mwmed = c(Jan = 0.721016267299377, Jan = 0.11185228712094, Jan = 0.173378451119196),
    pot_mes_mw = c(0, 0, 0.05441),
    adotantes_mes = c(0, 0, 5),
    p = c(0.000122345127598721, 0.000242743031345856, 0.00368794049368357),
    q = c(1, 1, 0.550183036478482)),
  row.names = c(NA, -3L),
```

```
    class = c("tbl_df", "tbl", "data.frame")
  )
  resultado <- epe4md_sumariza_resultados(resultados_mensais)
```

---

paleta\_epe

*Paleta de cores da EPE*

---

**Description**

Vetor contendo 11 cores diferentes referentes às cores utilizadas pela EPE.

**Usage**

paleta\_epe

**Format**

Um vetor com 11 cores.

# Index

epe4md\_calcula, [2](#), [21](#)  
epe4md\_calibra\_curva\_s, [6](#), [28](#)  
epe4md\_casos\_payback, [8](#), [24](#)  
epe4md\_fatores\_publicacao, [10](#)  
epe4md\_graf\_geracao\_ano, [10](#)  
epe4md\_graf\_geracao\_mes, [11](#)  
epe4md\_graf\_part\_fonte\_geracao, [13](#)  
epe4md\_graf\_part\_fonte\_potencia, [14](#)  
epe4md\_graf\_part\_segmento, [15](#)  
epe4md\_graf\_pot\_acum, [16](#)  
epe4md\_graf\_pot\_anual, [17](#)  
epe4md\_graf\_pot\_regiao, [18](#)  
epe4md\_graf\_pot\_segmento, [20](#)  
epe4md\_investimentos, [21](#)  
epe4md\_mercado\_potencial, [6](#), [22](#), [28](#)  
epe4md\_payback, [6](#), [23](#)  
epe4md\_prepara\_base, [26](#)  
epe4md\_proj\_adotantes, [28](#)  
epe4md\_proj\_geracao, [29](#)  
epe4md\_proj\_mensal, [30](#), [31](#)  
epe4md\_proj\_potencia, [31](#), [33](#)  
epe4md\_sumariza\_resultados, [35](#)

paleta\_epe, [36](#)