

# Package ‘enderecobr’

January 14, 2025

**Title** Padronizador de Endereços Brasileiros (Brazilian Addresses Standardizer)

**Version** 0.4.0

**Description** Padroniza endereços brasileiros a partir de diferentes critérios. Os métodos de padronização incluem apenas manipulações básicas de strings, não oferecendo suporte a correspondências probabilísticas entre strings. (Standardizes brazilian addresses using different criteria. Standardization methods include only basic string manipulation, not supporting probabilistic matches between strings.)

**License** MIT + file LICENSE

**URL** <https://github.com/ipeaGIT/enderecobr>,  
<https://ipeagit.github.io/enderecobr/>

**BugReports** <https://github.com/ipeaGIT/enderecobr/issues>

**Depends** R (>= 2.10)

**Imports** checkmate, cli, data.table, rlang, stringi, stringr, tibble

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

**VignetteBuilder** knitr

**Config/testthat/edition** 3

**Encoding** UTF-8

**Language** pt

**RoxygenNote** 7.3.2

**NeedsCompilation** no

**Author** Daniel Herszenhut [aut, cre] (<<https://orcid.org/0000-0001-8066-1105>>),  
Rafael H. M. Pereira [aut] (<<https://orcid.org/0000-0003-2125-7465>>),  
Lucas Mation [aut] (<<https://orcid.org/0000-0002-7461-932X>>)

**Maintainer** Daniel Herszenhut <dhersz@gmail.com>

**Repository** CRAN

**Date/Publication** 2025-01-14 19:00:02 UTC

## Contents

codigos_estados . . . . .	2
codigos_municipios . . . . .	3
correspondencia_campos . . . . .	3
correspondencia_logradouro . . . . .	5
padronizar_bairros . . . . .	6
padronizar_ceps . . . . .	7
padronizar_complementos . . . . .	8
padronizar_enderecos . . . . .	9
padronizar_estados . . . . .	11
padronizar_logradouros . . . . .	12
padronizar_logradouros_completos . . . . .	13
padronizar_municipios . . . . .	14
padronizar_numeros . . . . .	15
padronizar_tipos_de_logradouro . . . . .	16
<b>Index</b>	<b>18</b>

---

codigos_estados	<i>Códigos e nomes dos estados brasileiros (2022)</i>
-----------------	---

---

## Description

Tabela com a relação entre os códigos e nomes dos estados brasileiros. Os códigos foram convertidos para caracteres; os nomes foram convertidos para caracteres ASCII em caixa alta.

## Usage

codigos\_estados

## Format

Um data frame com 27 linhas e 3 colunas:

- codigo\_estado - código do estado;
- nome\_estado - nome do estado;
- abrev\_estado - abreviação do nome do estado.

## Source

<https://www.ibge.gov.br/explica/codigos-dos-municipios.php>

## See Also

[codigos\\_municipios](#)

---

codigos\_municipios      *Códigos e nomes dos municípios brasileiros (2022)*

---

### Description

Tabela com a relação entre os códigos e nomes dos municípios brasileiros. Os códigos foram convertidos para caracteres; os nomes foram convertidos para caracteres ASCII em caixa alta.

### Usage

codigos\_municipios

### Format

Um data frame com 5570 linhas e 3 colunas:

- `codigo_estado` - código do estado em que o município está localizado;
- `codigo_muni` - código do município;
- `nome_muni` - nome do município.

### Source

<https://www.ibge.gov.br/explica/codigos-dos-municipios.php>

### See Also

[codigos\\_estados](#)

---

correspondencia\_campos      *Correspondência entre os campos do endereço e as colunas que os descrevem*

---

### Description

Cria um vetor de caracteres que especifica as colunas que representam cada campo de endereço em um dataframe.

**Usage**

```
correspondencia_campos(  
  tipo_de_logradouro = NULL,  
  logradouro = NULL,  
  numero = NULL,  
  complemento = NULL,  
  cep = NULL,  
  bairro = NULL,  
  municipio = NULL,  
  estado = NULL  
)
```

**Arguments**

```
tipo_de_logradouro, logradouro, numero, complemento, cep, bairro,  
municipio, estado
```

Uma string. O nome da coluna que representa o respectivo campo de endereço no dataframe. Pode ser NULL, no caso do campo não estar listado. Ao menos um dos campos deve receber um valor não nulo.

**Value**

Um vetor nomeado de caracteres, em que os nomes representam os campos do endereço e os valores as colunas que os descrevem no dataframe.

**Examples**

```
enderecos <- data.frame(  
  id = 1,  
  tipo = "r",  
  log = "ns sra da piedade",  
  nroLogradouro = 20,  
  compl = "qd 20",  
  cep = 25220020,  
  bairro = "jd botanico",  
  codmun_dom = 3304557,  
  uf_dom = "rj"  
)  
  
# dado o dataframe acima, a seguinte chamada cria a correspondencia entre  
# suas colunas e os campos  
correspondencia_campos(  
  tipo_de_logradouro = "tipo",  
  logradouro = "log",  
  numero = "nroLogradouro",  
  complemento = "compl",  
  cep = "cep",  
  bairro = "bairro",  
  municipio = "codmun_dom",  
  estado = "uf_dom"  
)
```

---

correspondencia\_logradouro

*Correspondência entre os campos do logradouro completo e as colunas que os descrevem*

---

### Description

Cria um vetor de caracteres que especifica as colunas que representam os campos de logradouro (tipo, nome e número) em um dataframe de endereços.

### Usage

```
correspondencia_logradouro(  
  tipo_de_logradouro = NULL,  
  nome_do_logradouro = NULL,  
  numero = NULL  
)
```

### Arguments

tipo\_de\_logradouro, nome\_do\_logradouro, numero

Uma string. O nome da coluna que representa o respectivo campo do logradouro no dataframe. Pode ser NULL, no caso do campo não estar listado. Ao menos um dos campos deve receber um valor não nulo.

### Value

Um vetor nomeado de caracteres, em que os nomes representam os campos do logradouro e os valores as colunas que os descrevem no dataframe.

### Examples

```
enderecos <- data.frame(  
  tipo = "r",  
  log = "ns sra da piedade",  
  nroLogradouro = 20  
)  
  
# dado o dataframe acima, a seguinte chamada cria a correspondencia entre  
# suas colunas e os campos  
correspondencia_logradouro(  
  tipo_de_logradouro = "tipo",  
  nome_do_logradouro = "log",  
  numero = "nroLogradouro"  
)
```

---

padronizar\_bairros      *Padronizar bairros*

---

### Description

Padroniza um vetor de caracteres representando bairros de municípios brasileiros. Veja a seção *Detalhes* para mais informações sobre a padronização.

### Usage

```
padronizar_bairros(bairros)
```

### Arguments

bairros              Um vetor de caracteres. Os bairros a serem padronizados.

### Value

Um vetor de caracteres com os bairros padronizados.

### Detalhes

Operações realizadas durante a padronização:

- remoção de espaços em branco antes e depois das strings e remoção de espaços em excesso entre palavras;
- conversão de caracteres para caixa alta;
- remoção de acentos e caracteres não ASCII;
- adição de espaços após abreviações sinalizadas por pontos;
- expansão de abreviações frequentemente utilizadas através de diversas **expressões regulares (regexes)**;
- correção de alguns pequenos erros ortográficos.

### Examples

```
bairros <- c("PRQ IND", "NSA SEN DE FATIMA", "ILHA DO GOV")
padronizar_bairros(bairros)
```

---

padronizar_ceps	<i>Padronizar CEPs</i>
-----------------	------------------------

---

### Description

Padroniza um vetor de caracteres ou números representando CEPs. Veja a seção *Detalhes* para mais informações sobre a padronização.

### Usage

```
padronizar_ceps(ceps)
```

### Arguments

`ceps` Um vetor de caracteres ou números. Os CEPs a serem padronizados.

### Value

Um vetor de caracteres com os CEPs padronizados.

### Detalhes

Operações realizadas durante a padronização:

- conversão para caracter, se o input for numérico;
- adição de zeros à esquerda, se o input contiver menos de 8 dígitos;
- remoção de espaços em branco, pontos e vírgulas;
- adição de traço separando o radical (5 primeiros dígitos) do sufixo (3 últimos dígitos).

### Examples

```
ceps <- c("22290-140", "22.290-140", "22290 140", "22290140")
padronizar_ceps(ceps)
```

```
ceps <- c(22290140, 1000000, NA)
padronizar_ceps(ceps)
```

---

padronizar\_complementos

*Padronizar complementos*

---

## Description

Padroniza um vetor de caracteres representando complementos de logradouros. Veja a seção *Detalhes* para mais informações sobre a padronização.

## Usage

```
padronizar_complementos(complementos)
```

## Arguments

complementos    Um vetor de caracteres. Os complementos a serem padronizados.

## Value

Um vetor de caracteres com os complementos padronizados.

## Detalhes

Operações realizadas durante a padronização:

- remoção de espaços em branco antes e depois das strings e remoção de espaços em excesso entre palavras;
- conversão de caracteres para caixa alta;
- remoção de acentos e caracteres não ASCII;
- adição de espaços após abreviações sinalizadas por pontos;
- expansão de abreviações frequentemente utilizadas através de diversas **expressões regulares (regexes)**;
- correção de alguns pequenos erros ortográficos.

## Examples

```
complementos <- c("", "QD1 LT2 CS3", "APTO. 405")  
padronizar_complementos(complementos)
```



---

padronizar\_enderecos *Padronizar endereços*

---

## Description

Padroniza simultaneamente os diversos campos de um endereço listados em um dataframe.

## Usage

```
padronizar_enderecos(  
    enderecos,  
    campos_do_endereco = correspondencia_campos(),  
    formato_estados = "por_extenso",  
    formato_numeros = "character",  
    manter_cols_extras = TRUE,  
    combinar_logradouro = FALSE,  
    checar_tipos = FALSE  
)
```

## Arguments

- enderecos** Um dataframe. Os endereços a serem padronizados. Cada uma de suas colunas deve corresponder a um campo do endereço (e.g. logradouro, cidade, bairro, etc).
- campos\_do\_endereco** Um vetor nomeado de caracteres. A correspondência entre os campos a serem padronizados (nomes do vetor) e as colunas que os representam no dataframe (valores em si). A função `correspondencia_campos()` facilita a criação deste vetor, fazendo também algumas verificações do conteúdo imputado. Argumentos dessa função com valor NULL são ignorados, e ao menos um valor diferente de nulo deve ser fornecido. Caso deseje criar o vetor manualmente, note que seus nomes devem ser os mesmos nomes dos parâmetros da função `correspondencia_campos()`.
- formato\_estados** Uma string. Como o estado padronizado deve ser formatado. Por padrão, "por\_extenso", fazendo com que a função retorne o nome dos estados por extenso. Se "sigla", a função retorna a sigla dos estados.
- formato\_numeros** Uma string. Como o número padronizado deve ser formatado. Por padrão, "character", fazendo com que a função retorne o número como caractere. Se "integer", a função retorna o número como inteiro.
- manter\_cols\_extras** Um logical. Se colunas não especificadas em `campos_do_endereco` devem ser mantidas ou não (por exemplo, uma coluna de id do conjunto de dados sendo padronizado). Por padrão, TRUE.

**combinar\_logradouro**

Um logical. Se os campos que descrevem o logradouro (tipo, nome e número, por exemplo) devem ser combinados em um único campo de logradouro completo. Nesse caso, o parâmetro `logradouro` da `correspondencia_campos()` deve ser interpretado como o nome do logradouro. Por padrão, FALSE.

**checar\_tipos**

Um logical. Apenas tem efeito quando `combinar_logradouro` é TRUE. Se a ocorrência de duplicatas entre os tipos e nomes dos logradouros deve ser verificada ao combiná-los (por exemplo, quando o tipo é descrito como "RUA" e o nome é descrito como "RUA BOTAFOGO"). Por padrão, FALSE.

**Value**

Um dataframe com colunas adicionais, representando os campos de endereço padronizados.

**Examples**

```
enderecos <- data.frame(
  id = 1,
  logradouro = "r ns sra da piedade",
  nroLogradouro = 20,
  complemento = "qd 20",
  cep = 25220020,
  bairro = "jd botanico",
  codmun_dom = 3304557,
  uf_dom = "rj"
)

campos <- correspondencia_campos(
  logradouro = "logradouro",
  numero = "nroLogradouro",
  complemento = "complemento",
  cep = "cep",
  bairro = "bairro",
  municipio = "codmun_dom",
  estado = "uf_dom"
)

padronizar_enderecos(enderecos, campos)

padronizar_enderecos(enderecos, campos, manter_cols_extras = FALSE)

padronizar_enderecos(enderecos, campos, combinar_logradouro = TRUE)

ends_tipo_duplicado <- data.frame(tipo = "r", nome = "r ns sra da piedade")

padronizar_enderecos(
  ends_tipo_duplicado,
  campos_do_endereco = correspondencia_campos(
    tipo_de_logradouro = "tipo",
    logradouro = "nome"
  ),
  combinar_logradouro = TRUE,
```

```
    checar_tipos = TRUE
  )
```

---

padronizar\_estados      *Padronizar estados*

---

## Description

Padroniza um vetor de caracteres ou números representando estados brasileiros. Veja a seção *Detalhes* para mais informações sobre a padronização.

## Usage

```
padronizar_estados(estados, formato = "por_extenso")
```

## Arguments

estados	Um vetor de caracteres ou números. Os estados a serem padronizados.
formato	Uma string. Como o resultado padronizado deve ser formatado. Por padrão, "por_extenso", fazendo com que a função retorne o nome dos estados por extenso. Se "sigla", a função retorna a sigla dos estados.

## Value

Um vetor de caracteres com os estados padronizados.

## Detalhes

Operações realizadas durante a padronização:

- conversão para caracter, se o input for numérico;
- remoção de espaços em branco antes e depois dos valores e remoção de espaços em excesso entre palavras;
- conversão de caracteres para caixa alta;
- remoção de zeros à esquerda;
- busca, a partir do código numérico ou da abreviação da UF, do nome completo de cada estado;
- caso a busca não tenha encontrado determinado valor, remoção de acentos e caracteres não ASCII.

## Examples

```
estados <- c("21", "021", "MA", " 21", " MA ", "ma", "", NA)
padronizar_estados(estados)
```

```
estados <- c(21, NA)
padronizar_estados(estados)
padronizar_estados(estados, formato = "sigla")
```

---

padronizar\_logradouros

*Padronizar logradouros*

---

### Description

Padroniza um vetor de caracteres representando logradouros de municípios brasileiros. Veja a seção *Detalhes* para mais informações sobre a padronização.

### Usage

```
padronizar_logradouros(logradouros)
```

### Arguments

logradouros      Um vetor de caracteres. Os logradouros a serem padronizados.

### Value

Um vetor de caracteres com os logradouros padronizados.

### Detalhes

Operações realizadas durante a padronização:

- remoção de espaços em branco antes e depois das strings e remoção de espaços em excesso entre palavras;
- conversão de caracteres para caixa alta;
- remoção de acentos e caracteres não ASCII;
- adição de espaços após abreviações sinalizadas por pontos;
- expansão de abreviações frequentemente utilizadas através de diversas **expressões regulares (regexes)**;
- correção de alguns pequenos erros ortográficos.

### Examples

```
logradouros <- c("r. gen.. glicério")  
padronizar_logradouros(logradouros)
```

---

`padronizar_logradouros_completos`*Padronizar logradouros completos*

---

## Description

Padroniza o logradouro completo a partir de diversos campos (tipo de logradouro, nome do logradouro e número), garantindo a consistência da informação.

## Usage

```
padronizar_logradouros_completos(  
  enderecos,  
  campos_do_logradouro = correspondencia_logradouro(),  
  manter_cols_extras = TRUE,  
  checar_tipos = FALSE  
)
```

## Arguments

- |                                   |   |
|-----------------------------------|---|
| <code>enderecos</code>            | Um dataframe. Os endereços a serem padronizados. Ao menos uma de suas colunas deve corresponder a um campo do logradouro.   |
| <code>campos_do_logradouro</code> | Um vetor nomeado de caracteres. A correspondência entre os campos a serem padronizados (nomes do vetor) e as colunas que os representam no dataframe (valores do vetor). A função <code>correspondencia_logradouro()</code> facilita a criação deste vetor, fazendo também algumas verificações do conteúdo imputado. Caso deseje criar o vetor manualmente, note que seus nomes devem ser os mesmos nomes dos parâmetros da função <code>correspondencia_logradouro()</code> . |
| <code>manter_cols_extras</code>   | Um logical. Se colunas não especificadas em <code>campos_do_logradouro</code> devem ser mantidas no output ou não (por exemplo, uma coluna com a informação de bairro ou com o id do conjunto de dados sendo padronizado). Por padrão, TRUE.  |
| <code>checar_tipos</code>         | Um logical. Se a ocorrência de duplicatas entre os tipos e nomes dos logradouros deve ser verificada ao combiná-los (por exemplo, quando o tipo é descrito como "RUA" e o nome é descrito como "RUA BOTAFOGO"). Por padrão, FALSE.  |

## Value

Caso `manter_cols_extras` seja TRUE, o mesmo dataframe de input, mas sem as colunas descrevendo o logradouro e com uma coluna padronizada adicional `logradouro_completo`. Caso `manter_cols_extras` seja FALSE, um dataframe de apenas uma coluna, `logradouro_completo`.

**Examples**

```
enderecos <- data.frame(
  id = 1,
  tipoLogradouro = "r",
  logradouro = "ns sra da piedade",
  nroLogradouro = 20,
  complemento = "qd 20",
  cep = 25220020,
  bairro = "jd botanico",
  codmun_dom = 3304557,
  uf_dom = "rj"
)

campos <- correspondencia_logradouro(
  tipo_de_logradouro = "tipoLogradouro",
  nome_do_logradouro = "logradouro",
  numero = "nroLogradouro"
)

padronizar_logradouros_completos(enderecos, campos)

padronizar_logradouros_completos(
  enderecos,
  campos,
  manter_cols_extras = FALSE
)

enderecos <- data.frame(
  tipoLogradouro = "r",
  logradouro = "r ns sra da piedade",
  nroLogradouro = 20
)
padronizar_logradouros_completos(enderecos, campos, checar_tipos = TRUE)
```

---

padronizar\_municipios *Padronizar municípios*

---

**Description**

Padroniza um vetor de caracteres ou números representando municípios brasileiros. Veja a seção *Detalhes* para mais informações sobre a padronização.

**Usage**

```
padronizar_municipios(municipios)
```

**Arguments**

`municipios` Um vetor de caracteres ou números. Os municípios a serem padronizados.

**Value**

Um vetor de caracteres com os municípios padronizados.

**Detalhes**

Operações realizadas durante a padronização:

- conversão para caracter, se o input for numérico;
- remoção de espaços em branco antes e depois dos valores e remoção de espaços em excesso entre palavras;
- conversão de caracteres para caixa alta;
- remoção de zeros à esquerda;
- busca, a partir do código numérico, do nome completo de cada município;
- caso a busca não tenha encontrado determinado valor, remoção de acentos e caracteres não ASCII, correção de erros ortográficos frequentes e atualização de nomes conforme listagem de municípios do IBGE de 2022.

**Examples**

```
municipios <- c(
  "3304557", "003304557", " 3304557 ", "RIO DE JANEIRO", "rio de janeiro",
  "SÃO PAULO",
  "", NA
)
padronizar_municipios(municipios)

municipios <- c(3304557, NA)
padronizar_municipios(municipios)

municipios <- c("PARATI", "AUGUSTO SEVERO", "SAO VALERIO DA NATIVIDADE")
padronizar_municipios(municipios)
```

---

padronizar\_numeros      *Padronizar números de logradouros*

---

**Description**

Padroniza um vetor de caracteres ou números representando números de logradouros. Veja a seção *Detalhes* para mais informações sobre a padronização.

**Usage**

```
padronizar_numeros(numeros, formato = "character")
```

**Arguments**

numeros	Um vetor de caracteres ou números. Os números de logradouro a serem padronizados.
formato	Uma string. Como o resultado padronizado deve ser formatado. Por padrão, "character", fazendo com que a função retorne um vetor de caracteres. Se "integer", a função retorna um vetor de números inteiros.

**Value**

Um vetor de caracteres com os números de logradouros padronizados.

**Detalhes**

Operações realizadas durante a padronização:

- conversão para character, se o input for numérico;
- remoção de espaços em branco antes e depois dos números e de espaços em branco em excesso entre números;
- remoção de zeros à esquerda;
- substituição de números vazios e de variações de SN (SN, S N, S.N., S./N., etc) por S/N.

**Examples**

```
numeros <- c("0210", "001", "1", "", "S N", "S/N", "SN", "0180 0181")
padronizar_numeros(numeros)
```

```
numeros <- c(210, 1, 10000, NA)
padronizar_numeros(numeros)
```

---

padronizar\_tipos\_de\_logradouro

*Padronizar tipos de logradouro*

---

**Description**

Padroniza um vetor de caracteres representando tipos de logradouro. Veja a seção *Detalhes* para mais informações sobre a padronização.

**Usage**

```
padronizar_tipos_de_logradouro(tipos)
```

**Arguments**

tipos	Um vetor de caracteres. Os tipos de logradouro a serem padronizados.
-------	--



### **Value**

Um vetor de caracteres com os tipos de logradouro padronizados.

### **Detalhes**

Operações realizadas durante a padronização:

- remoção de espaços em branco antes e depois das strings e remoção de espaços em excesso entre palavras;
- conversão de caracteres para caixa alta;
- remoção de acentos e caracteres não ASCII;
- adição de espaços após abreviações sinalizadas por pontos;
- expansão de abreviações frequentemente utilizadas através de diversas **expressões regulares (regexes)**;
- correção de alguns pequenos erros ortográficos.

### **Examples**

```
tipos <- c("R", "AVE", "QDRA")
padronizar_tipos_de_logradouro(tipos)
```

# Index

## \* datasets

- codigos\_estados, [2](#)
- codigos\_municipios, [3](#)

- codigos\_estados, [2](#), [3](#)
- codigos\_municipios, [2](#), [3](#)
- correspondencia\_campos, [3](#)
- correspondencia\_logradouro, [5](#)

- padronizar\_bairros, [6](#)
- padronizar\_ceps, [7](#)
- padronizar\_complementos, [8](#)
- padronizar\_enderecos, [9](#)
- padronizar\_estados, [11](#)
- padronizar\_logradouros, [12](#)
- padronizar\_logradouros\_completos, [13](#)
- padronizar\_municipios, [14](#)
- padronizar\_numeros, [15](#)
- padronizar\_tipos\_de\_logradouro, [16](#)