Package: geoecu (via r-universe)

October 23, 2026

Type Package

Title Maps of the Provinces, Cantons and Parishes of Ecuador (Mapas de Provincias, Cantones y Parroquias del Ecuador)

Version 0.4.0

Description Provides the boundaries of the provinces, cantons and parishes of Ecuador as 'sf' objects, identified by the codes of the national political-administrative division (DPA, ``División Político-Administrativa''), ready to be mapped and joined with health, epidemiological surveillance or other data. The three levels nest exactly and are offered in two resolutions: a light one for maps and dashboards and a detailed one for spatial analysis. The layers are derived from the open data of the Territorial Organization of the State published by the National Committee of Internal Boundaries (CONALI, Ministry of Government of Ecuador) under a CC BY licence <https://www.datosabiertos.gob.ec/>, and are downloaded on first use as 'GeoPackage' files, verified and cached locally. Includes a catalogue of DPA codes, a table to update codes from earlier versions, helpers to clean codes and place names, and tools to relocate the Galapagos Islands in national maps. Documentation is in Spanish.

License MIT + file LICENSE

Copyright see file inst/COPYRIGHTS

URL https://f-slg.github.io/geoecu/, https://github.com/f-slg/geoecu

BugReports https://github.com/f-slg/geoecu/issues

Depends R (>= 4.1.0)

Imports sf, stats, tools, utils

Suggests curl, ggplot2, testthat (>= 3.0.0)

Config/testthat/edition 3

Config/Needs/website rmarkdown, knitr

Encoding UTF-8

Language es

LazyData true

Roxygen list(markdown = TRUE)

Config/roxygen2/version 8.1.1

Config/pak/sysreqs libabsl-dev cmake libgdal-dev gdal-bin libgeos-dev libssl-dev libproj-dev libsqlite3-dev libudunits2-dev

Repository https://f-slg.r-universe.dev

Date/Publication 2026-10-22 18:11:10 UTC

RemoteUrl https://github.com/f-slg/geoecu

RemoteRef HEAD

RemoteSha 4e549b3280b1366f170c4bb6f1d1fcf9510388da

Contents

actualizar_codigo 2
buscar_dpa 3
dpa 4
dpa_equivalencias 5
formatear_codigo 6
geoecu_cache_dir 6
geoecu_datos_disponibles 7
geoecu_limpiar_cache 7
geoecu_unir 8
geoecu_usar_carpeta 8
marco_galapagos 9
normalizar_nombre 10
read_dpa 10
reubicar_galapagos 12

Index **14**

`actualizar_codigo` *Actualizar códigos de parroquia de versiones anteriores*

Description

Convierte códigos de parroquia de los datos de geoecu 0.3.0 y anteriores (por ejemplo, de parroquias urbanas o del antiguo cantón Eloy Alfaro) al código actual, usando la equivalencia principal de dpa_equivalencias. Los códigos vigentes se devuelven igual, aunque sus límites hayan cambiado.

Usage

```
actualizar_codigo(x)
```

Arguments

x	Vector de códigos de parroquia (número o texto).

Details

Cuando menos del 90 % de una parroquia anterior quedó en una sola parroquia actual, se usa la de mayor superposición y se avisa: revisa dpa_equivalencias si necesitas repartir los datos.

Value

Vector de texto con los códigos actuales. Los códigos que no existen en ninguna versión se devuelven como NA con una advertencia.

Examples

```
# Parroquia urbana de Cuenca, Borbón (antes en Eloy Alfaro) y una sin cambios
actualizar_codigo(c("010101", "080253", "110152"))

# Agregar una tabla antigua al código actual
casos <- data.frame(cod = c("010101", "010102", "010150"), casos = c(3, 4, 1))
casos$cod <- actualizar_codigo(casos$cod)
aggregate(casos ~ cod, data = casos, FUN = sum)
```

---

buscar_dpa *Buscar en el catálogo DPA*

Description

Busca por nombre (sin importar tildes ni mayúsculas) o por código en el catálogo dpa.

Usage

```
buscar_dpa(texto, nivel = c("parroquia", "canton", "provincia"))
```

Arguments

texto	Texto a buscar: parte de un nombre o un código.
nivel	Nivel en el que buscar: "parroquia", "canton" o "provincia".

Value

Un data.frame con las coincidencias en el nivel indicado.

Examples

```
buscar_dpa("arenal")
buscar_dpa("puyango", nivel = "canton")
buscar_dpa("11", nivel = "provincia")
```

dpa *Catálogo de la División Político-Administrativa (DPA) del Ecuador*

Description

Una fila por parroquia, con sus códigos y nombres en los tres niveles. Sirve para buscar códigos, validar tablas antes de unirlas con las capas o agregar datos parroquiales a cantón o provincia sin necesidad de geometrías.

Usage

```
dpa
```

Format

Un `data.frame` con 1051 filas y 7 columnas:

dpa_parroq Código DPA de la parroquia (6 dígitos, texto).

parroquia Nombre de la parroquia, en mayúsculas.

tipo `"cabecera cantonal"`, `"parroquia rural"` o `"zona en estudio"`.

dpa_canton Código DPA del cantón (4 dígitos, texto).

canton Nombre del cantón.

dpa_provin Código DPA de la provincia (2 dígitos, texto).

provincia Nombre de la provincia.

Details

Incluye 1048 parroquias (223 cabeceras cantonales y 825 parroquias rurales) de 223 cantones y 24 provincias, más 3 zonas en estudio (territorios con límites aún no resueltos): Jaramijó-Manta (código `"139090"`), Sinaí-Cuchaentza (`"140190"`) y Juval (`"900651"`, provincia `"90"`). Las parroquias urbanas forman parte de su cabecera cantonal.

Source

Organización Territorial del Estado, Comité Nacional de Límites Internos (CONALI), Ministerio de Gobierno del Ecuador, versión del 17 de julio de 2026. Datos abiertos, licencia CC BY: https://www.datosabiertos.gob.ec/dataset/organizacion-territorial-parroquial. Ver `data-raw/` en el repositorio para el procesamiento completo.

See Also

dpa_equivalencias para actualizar códigos de versiones anteriores de geoecu.

Examples

```
head(dpa)
# Número de parroquias por provincia
table(dpa$provincia)
```

dpa_equivalencias — *Equivalencias con los códigos de parroquia de versiones anteriores*

Description

Relaciona los códigos de parroquia de los datos de geoecu 0.3.0 y anteriores (release `datos-v0.1.0`) con los de la versión actual, según la superposición de sus áreas. Útil para actualizar tablas codificadas con la versión anterior; para hacerlo directamente, usa `actualizar_codigo()`.

Usage

```
dpa_equivalencias
```

Format

Un `data.frame` con 7 columnas:

dpa_parroq_anterior Código de la parroquia en la versión anterior.

parroquia_anterior Nombre en la versión anterior.

dpa_parroq Código de la parroquia actual.

parroquia Nombre de la parroquia actual.

proporcion Proporción del área de la parroquia anterior que cae en la actual (0 a 1).

principal `TRUE` en la parroquia actual con mayor superposición.

cambio `"sin cambio"`, `"integrada en la cabecera cantonal"` o `"código distinto"`.

Details

Los cambios principales son: las parroquias urbanas pasan a formar parte de su cabecera cantonal; las parroquias del nuevo cantón Borbón (antes en Eloy Alfaro) y la cabecera de Sevilla Don Bosco cambian de código; y algunas parroquias cambian de límites o se integran en otras.

Una parroquia anterior puede repartirse entre varias actuales: hay una fila por cada parte que cubre al menos el 1 % de su área, y `principal` marca la de mayor superposición.

Source

Elaborado en `data-raw/04_equivalencias.R` a partir de las capas de alta resolución de ambas versiones.

Examples

```
# Parroquias que cambiaron de código
subset(dpa_equivalencias, principal & cambio == "código distinto")
```

formatear_codigo *Formatear códigos DPA*

Description

Devuelve códigos DPA como texto con el número correcto de dígitos, recuperando los ceros a la izquierda que se pierden cuando un código se guarda como número en Excel (por ejemplo, `1101` en lugar de `"1101"`, o `701` en lugar de `"0701"`).

Usage

```
formatear_codigo(x, nivel = c("parroquia", "canton", "provincia"))
```

Arguments

`x`	Vector de códigos (número o texto).
`nivel`	`"provincia"` (2 dígitos), `"canton"` (4) o `"parroquia"` (6).

Value

Vector de texto. Los valores que no son códigos válidos se devuelven como `NA` con una advertencia.

Examples

```
formatear_codigo(c(1, 11, "7"), "provincia")
formatear_codigo(c(701, 1101), "canton")
formatear_codigo(110150, "parroquia")
```

geoecu_cache_dir *Carpeta de caché de geoecu*

Description

Los GeoPackages descargados se guardan aquí para no volver a descargarlos. La ubicación sigue las convenciones de cada sistema operativo (`tools::R_user_dir()`).

Usage

```
geoecu_cache_dir()
```

Value

Ruta de la carpeta de caché (texto).

Examples

```
geoecu_cache_dir()
```

geoecu_datos_disponibles *¿Están disponibles los datos sin descargar?*

Description

Comprueba si el GeoPackage de la resolución indicada ya está en la carpeta local configurada o en la caché.

Usage

```
geoecu_datos_disponibles(resolucion = "baja")
```

Arguments

`resolucion`	`"baja"` o `"alta"`.

Value

`TRUE` o `FALSE`.

Examples

```
geoecu_datos_disponibles()
```

geoecu_limpiar_cache *Borrar los datos descargados*

Description

Elimina los GeoPackages guardados en la caché. Se volverán a descargar la próxima vez que se necesiten.

Usage

```
geoecu_limpiar_cache()
```

Value

`TRUE` de forma invisible.

Examples

```
geoecu_limpiar_cache()
```

geoecu_unir — *Unir datos propios con una capa geográfica*

Description

Une una tabla (casos, tasas, coberturas, etc.) con la capa del nivel indicado usando el código DPA. Los códigos se formatean automáticamente, así que funciona aunque vengan como número desde Excel. Se conservan todos los polígonos de la capa: los territorios sin datos quedan con `NA`, lo que permite mapearlos.

Usage

```
geoecu_unir(datos, codigo, nivel = c("canton", "provincia", "parroquia"), ...)
```

Arguments

`datos`	Un `data.frame` con una columna de código DPA.
`codigo`	Nombre de la columna de `datos` que contiene el código.
`nivel`	`"provincia"`, `"canton"` o `"parroquia"`.
`...`	Otros argumentos para la función `read_*()` correspondiente (por ejemplo, `provincia`, `resolucion` o `crs`).

Value

Un objeto `sf` con las columnas de la capa más las de `datos`. Si hay códigos en `datos` que no existen en la capa, se avisa con una advertencia.

Examples

```
casos <- data.frame(cod = c(1101, 1102, 1103), casos = c(12, 3, 7))
mapa <- geoecu_unir(casos, codigo = "cod", nivel = "canton",
                    provincia = "Loja")
plot(mapa["casos"])
```

geoecu_usar_carpeta — *Usar GeoPackages desde una carpeta local*

Description

Indica a geoecu una carpeta que contiene `geoecu_dpa_baja.gpkg` y/o `geoecu_dpa_alta.gpkg`. Útil para trabajar sin conexión, en servidores institucionales o con datos que aún no se han publicado.

Usage

```
geoecu_usar_carpeta(carpeta)
```

Arguments

carpeta	Ruta a la carpeta. Usa NULL para volver al comportamiento por defecto (caché y descarga).

Details

También puedes definir la variable de entorno GEOECU_CARPETA (por ejemplo, en tu .Renviron) para que la carpeta se use siempre.

Value

La carpeta anterior, de forma invisible.

Examples

```
# Usar una carpeta propia y luego volver a la configuración anterior
anterior <- geoecu_usar_carpeta(tempdir())
geoecu_datos_disponibles()
geoecu_usar_carpeta(anterior)
```

---

marco_galapagos *Recuadro para Galápagos reubicado*

Description

Devuelve un rectángulo alrededor de Galápagos, pensado para dibujarlo como un recuadro tenue después de reubicar_galapagos().

Usage

```
marco_galapagos(x, margen = 0.06)
```

Arguments

x	Objeto sf de geoecu, normalmente ya pasado por reubicar_galapagos().
margen	Margen alrededor de las islas, como proporción del ancho del archipiélago. Por defecto 0.06.

Value

Un objeto sfc con un polígono, en el mismo sistema de coordenadas que x.

Examples

```
prov <- reubicar_galapagos(read_provincias(crs = 32717))
marco_galapagos(prov)
```

`normalizar_nombre` *Normalizar nombres geográficos*

Description

Convierte nombres a una forma comparable: mayúsculas, sin tildes ni diéresis, Ñ como N, sin signos de puntuación y con espacios simples. Útil para unir tablas que escriben los nombres de forma distinta ("Cañar", "CANAR", "cañar ").

Usage

```
normalizar_nombre(x)
```

Arguments

x	Vector de texto.

Value

Vector de texto normalizado.

Examples

```
normalizar_nombre(c("Cañar", "  san  José de Chimbo", "CHECA (JIDCAY)"))
```

`read_dpa` *Leer las capas de provincias, cantones o parroquias*

Description

Devuelven un objeto `sf` con los límites del nivel solicitado y sus códigos DPA, según la Organización Territorial del Estado (CONALI). Los tres niveles encajan exactamente: cada cantón es la unión de sus parroquias y cada provincia la unión de sus cantones.

Usage

```
read_provincias(provincia = NULL, resolucion = c("baja", "alta"), crs = 4326)

read_cantones(
  provincia = NULL,
  canton = NULL,
  resolucion = c("baja", "alta"),
  crs = 4326
)

read_parroquias(
  provincia = NULL,
  canton = NULL,
  tipo = NULL,
  resolucion = c("baja", "alta"),
  crs = 4326,
  area = NULL
)
```

Arguments

`provincia`	Provincias a incluir, por nombre (sin importar tildes ni mayúsculas) o código: `"Loja"`, `"11"` o `11`. `NULL` (por defecto) devuelve todo el país.
`resolucion`	`"baja"` (por defecto; liviana, para mapas y tableros) o `"alta"` (detallada, para análisis espacial).
`crs`	Sistema de coordenadas de salida. Por defecto WGS84 (EPSG:4326). Usa `32717` (UTM 17S) para calcular áreas o distancias en metros.
`canton`	Cantones a incluir, por nombre o código. Si un nombre se repite en varias provincias (por ejemplo, Bolívar u Olmedo), usa el código o indica también `provincia`.
`tipo`	Solo para parroquias: `"cabecera"` (cabeceras cantonales), `"rural"` (parroquias rurales), `"zona"` (zonas en estudio) o `NULL` (todas). Se pueden combinar, por ejemplo `c("cabecera", "rural")`.
`area`	Obsoleto. Usa `tipo`. `"rural"` equivale a `tipo = "rural"` y `"urbana"` a `tipo = "cabecera"`.

Details

Además de las circunscripciones, las capas incluyen las zonas en estudio (territorios con límites aún no resueltos), identificadas en la columna `tipo`, para que el mapa nacional no tenga huecos. Para excluirlas, filtra `tipo != "zona en estudio"`. Las parroquias son las de la división político-administrativa: cabeceras cantonales y parroquias rurales. Las parroquias urbanas forman parte de su cabecera cantonal.

La primera vez se descarga el GeoPackage de la resolución pedida y se guarda en caché; las siguientes veces se lee directamente del disco.

Value

Un objeto `sf` de polígonos con las columnas:

- `read_provincias()`: `dpa_provin`, `provincia`, `tipo`, `area_km2`.
- `read_cantones()`: `dpa_canton`, `canton`, `tipo`, `dpa_provin`, `provincia`, `area_km2`.
- `read_parroquias()`: `dpa_parroq`, `parroquia`, `tipo`, `dpa_canton`, `canton`, `dpa_provin`, `provincia`, `area_km2`.

`tipo` vale `"provincia"`, `"cantón"`, `"cabecera cantonal"` o `"parroquia rural"` según el nivel, o `"zona en estudio"`.

Source

Organización Territorial del Estado, Comité Nacional de Límites Internos (CONALI), Ministerio de Gobierno del Ecuador. Datos abiertos, licencia CC BY: https://www.datosabiertos.gob.ec/dataset/organizacion-territorial-parroquial.

Examples

```
provincias <- read_provincias()
plot(sf::st_geometry(provincias))

# Cantones de una provincia, en UTM 17S
loja <- read_cantones(provincia = "Loja", crs = 32717)

# Parroquias rurales de un cantón
read_parroquias(canton = "Puyango", tipo = "rural")
```

---

reubicar_galapagos *Reubicar Galápagos junto al continente para mapas nacionales*

Description

Traslada las islas Galápagos (provincia `"20"`) a la esquina superior izquierda del mapa, junto al Ecuador continental, como se hace en la cartografía nacional para no desperdiciar espacio. Funciona con cualquier capa de geoecu (provincias, cantones o parroquias) y con capas unidas con `geoecu_unir()`, porque usa la columna `dpa_provin`.

Usage

```
reubicar_galapagos(x, separacion = 0.06)
```

Arguments

`x`	Objeto `sf` de geoecu con la columna `dpa_provin` y polígonos del continente y de Galápagos. Se recomienda un sistema proyectado, como UTM 17S (`crs = 32717`).
`separacion`	Separación entre Galápagos y el continente, como proporción del ancho del Ecuador continental. Por defecto `0.06`.

Details

El traslado es **solo para visualizar**: no uses el resultado para medir áreas o distancias de Galápagos. Si reubicas varias capas para superponerlas (por ejemplo, parroquias y límites provinciales), todas quedan alineadas porque comparten la misma geometría de origen.

Value

El mismo objeto `sf`, con Galápagos trasladado. Si `x` no contiene Galápagos o no contiene el continente, se devuelve sin cambios.

See Also

`marco_galapagos()` para dibujar el recuadro.

Examples

```
library(ggplot2)
prov <- reubicar_galapagos(read_provincias(crs = 32717))
ggplot(prov) +
  geom_sf() +
  geom_sf(data = marco_galapagos(prov), fill = NA, color = "grey70") +
  theme_void()
```

Index

∗ **datasets**
    dpa, 4
    dpa_equivalencias, 5

actualizar_codigo, 2
actualizar_codigo(), *5*

buscar_dpa, 3

dpa, *3*, 4
dpa_equivalencias, *2–4*, 5

formatear_codigo, 6

geoecu_cache_dir, 6
geoecu_datos_disponibles, 7
geoecu_limpiar_cache, 7
geoecu_unir, 8
geoecu_unir(), *12*
geoecu_usar_carpeta, 8

marco_galapagos, 9
marco_galapagos(), *13*

normalizar_nombre, 10

read_cantones *(*read_dpa*)*, 10
read_dpa, 10
read_parroquias *(*read_dpa*)*, 10
read_provincias *(*read_dpa*)*, 10
reubicar_galapagos, 12
reubicar_galapagos(), *9*

tools::R_user_dir(), *6*