

Package ‘peruocc’

October 6, 2026

Title Query and Standardize Biodiversity Occurrences in Peru

Version 0.1.1

Description Facilitates the retrieval, spatial validation, and integration of flora and fauna occurrence records across administrative units (districts and provinces) in Peru. Retrieves official boundary geometries via 'geoperu', queries and consolidates observations from the Global Biodiversity Information Facility (GBIF, <<https://www.gbif.org/>>) and 'iNaturalist' (<<https://www.inaturalist.org/>>), and standardizes attributes into a unified Darwin Core aligned structure. Designed for biodiversity assessments and spatial workflows within user-defined areas of interest.

License MIT + file LICENSE

Encoding UTF-8

Depends R (>= 4.1.0)

Imports cli, dplyr, geoperu, ggplot2, jsonlite, readr, rgbif, rinat, sf

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

URL <https://paulesantos.github.io/peruocc/>

BugReports <https://github.com/PaulESantos/peruocc/issues>

Config/roxygen2/version 8.1.0

NeedsCompilation no

Author Paul E. Santos Andrade [aut, cre, cph] (ORCID: <<https://orcid.org/0000-0002-6635-0375>>)

Maintainer Paul E. Santos Andrade <paulefrens@gmail.com>

Repository CRAN

Date/Publication 2026-10-06 07:30:02 UTC

Contents

as_peruocc_tbl	2
buscar_especies_distrito	3
buscar_especies_peru	4
buscar_especies_poligono	6
buscar_especies_provincia	8
exportar_resultados	9
graficar_ocurrencias	10
obtener_poligono_distrito	11
obtener_poligono_provincia	12
obtener_poligono_unidad	12
peruocc_data_dir	13
preparar_poligono_usuario	14
verificar_y_configurar_entorno	15
Index	16

as_peruocc_tbl	<i>Coerción a objeto tabular ligero (estilo tibble)</i>
----------------	---

Description

Convierte un `data.frame` u objeto compatible en una estructura tabular con clase `c("peruocc_tbl", "tbl_df", "tbl", "data.frame")`, compatible con el ecosistema tidyverse sin generar conflictos ni dependencias pesadas.

Usage

```
as_peruocc_tbl(x, ...)  
  
## S3 method for class 'data.frame'  
as_peruocc_tbl(x, ...)  
  
## Default S3 method:  
as_peruocc_tbl(x, ...)  
  
## S3 method for class 'peruocc_tbl'  
x[i, j, drop = FALSE]  
  
## S3 method for class 'peruocc_tbl'  
print(x, n = 10L, width = NULL, ...)
```

Arguments

- x Un `data.frame`, lista o matriz a convertir, o un objeto `peruocc_tbl`.
- ... Argumentos adicionales pasados a otros métodos.
- i, j Índices de filas y columnas para extracción o indexación tabular.

drop	Lógico. Si es TRUE, simplifica a vector cuando el resultado es unidimensional.
n	Entero positivo con el número de filas a mostrar en consola.
width	Entero con el ancho de pantalla en caracteres; si es NULL, toma <code>getOption("width")</code> .

Value

Un objeto tabular con clase `c("peruocc_tbl", "tbl_df", "tbl", "data.frame")`.

Examples

```
df <- data.frame(a = 1:5, b = letters[1:5])
tbl <- as_peruocc_tbl(df)
class(tbl)
```

buscar_especies_distrito

Busca ocurrencias en un distrito peruano

Description

Atajo legible de `buscar_especies_peru()` con `nivel = "distrito"`. Acepte los mismos filtros y devuelve la misma estructura de resultado.

Usage

```
buscar_especies_distrito(
  distrito,
  departamento = NULL,
  provincia = NULL,
  nombre_cientifico = NULL,
  grupo = NULL,
  limite_por_api = configuracion_predeterminada()$limite_por_api,
  guardar_resultados = FALSE,
  tolerancia_simplificacion = configuracion_predeterminada()$tolerancia_simplificacion_m,
  ...
)
```

Arguments

distrito	Cadena no vacía con el nombre del distrito.
departamento	NULL o departamento que contiene al distrito.
provincia	NULL o provincia que contiene al distrito; úselo junto a departamento para resolver nombres repetidos.
nombre_cientifico	NULL o nombre de taxón para filtrar resultados.
grupo	NULL, "flora" o "fauna".

`limite_por_api` Entero entre 1 y 10000, o NULL; consulte `buscar_especies_peru()` para sus límites y consecuencias.
`guardar_resultados` Lógico que exporta CSV, GeoJSON y manifiesto al finalizar cuando es TRUE.
`tolerancia_simplificacion` Tolerancia de simplificación para la llamada a GBIF, expresada en metros.
`...` Controles avanzados reenviados a `buscar_especies_peru()`: `estrategia_espacial`, `max_area_ha`, `cache_dir`, `reintentos` y `pausa_entre_lotes_s`.

Value

Lista con límite, ocurrencias, resumen y parámetros. Consulte el valor retornado por `buscar_especies_peru()`.

Examples

```
## Not run:
res <- buscar_especies_distrito("Miraflores", departamento = "Lima", grupo = "flora")
head(res$ocurrencias)

## End(Not run)
```

`buscar_especies_peru` *Busca y consolida ocurrencias en una unidad administrativa del Perú*

Description

Es la función general de consulta. Obtiene el límite oficial, consulta GBIF e iNaturalist, valida localmente que cada coordenada esté dentro del polígono y unifica las columnas en un esquema común. Para áreas extensas puede dividir la consulta en lotes con checkpoint, evitando que un fallo obligue a empezar de nuevo.

Usage

```
buscar_especies_peru(
  nombre,
  nivel = c("distrito", "provincia"),
  departamento = NULL,
  provincia = NULL,
  nombre_cientifico = NULL,
  grupo = NULL,
  limite_por_api = configuracion_predeterminada()$limite_por_api,
  guardar_resultados = FALSE,
  dir_salida = NULL,
  tolerancia_simplificacion = configuracion_predeterminada()$tolerancia_simplificacion_m,
  estrategia_espacial = c("auto", "directa", "segmentada"),
  max_area_ha = configuracion_predeterminada()$max_area_ha_por_lote,
  max_lotes = configuracion_predeterminada()$max_lotes_espaciales,
```

```

cache_dir = ruta_cache("consultas_ocurrencias"),
reintentos = configuracion_predeterminada()$reintentos_api,
pausa_entre_lotes_s = configuracion_predeterminada()$pausa_entre_lotes_s
)

```

Arguments

nombre	Cadena no vacía con el distrito o provincia solicitado, según nivel. La coincidencia ignora tildes y mayúsculas.
nivel	Uno de "distrito" o "provincia"; determina cómo se interpreta nombre. Las provincias se resuelven con sus distritos componentes cuando se usa la estrategia segmentada.
departamento	NULL o cadena con el departamento. Es muy recomendable para resolver nombres repetidos y reduce la descarga de límites.
provincia	NULL o cadena con la provincia. Solo aplica a búsquedas distritales y ayuda a desambiguar homónimos.
nombre_cientifico	NULL o cadena con un taxón, por ejemplo "Panthera onca". GBIF intenta resolverlo en su backbone taxonómico; iNaturalist lo usa como filtro por nombre.
grupo	NULL, "flora" o "fauna" (sin distinguir mayúsculas). Se traduce a Plantae o Animalia. Puede combinarse con nombre_cientifico.
limite_por_api	Entero entre 1 y 10000, o NULL. Es el máximo por fuente y lote, no el máximo final consolidado. NULL solicita descarga completa solo cuando cada API informa un conteo dentro de su capacidad;
guardar_resultados	Lógico. Si es TRUE, ejecuta <code>exportar_resultados()</code> al final. No sobrescribe resultados previos porque genera un identificador temporal nuevo.
dir_salida	Ruta de destino si guardar_resultados = TRUE.
tolerancia_simplificacion	Distancia en metros para simplificar WKT en GBIF si supera el límite de longitud de la API.
estrategia_espacial	Estrategia de particionamiento ("auto", "directa" o "segmentada"). Con "auto", las provincias se particionan por sus distritos y los distritos extensos se dividen en macro-bloques adaptativos.
max_area_ha	Límite de área en hectáreas por lote para teselación cuando se usa estrategia_espacial = "segmentada".
max_lotes	Número máximo de macro-bloques espaciales generados por unidad geográfica para evitar saturar las cuotas de las APIs.
cache_dir	Directorio para guardar checkpoints .rds por lote y fuente.
reintentos	Entero positivo con el número de intentos para llamadas API.
pausa_entre_lotes_s	Pausa en segundos entre lotes consecutivos.

Details

La deduplicación usa source y sourceRecordID; un mismo registro procedente de GBIF e iNaturalist se mantiene, porque son fuentes distintas. Los checkpoints se escriben tras terminar cada fuente/lote. Revise resultado\$resumen\$fallos_lotes antes de interpretar una descarga como completa.

Value

Objeto con clase peruocc_resultado (lista con límite unidad_sf, tibble de ocurrencias, resumen estadístico y parametros).

Examples

```
## Not run:
resultado <- buscar_especies_peru(
  nombre = "Tambopata", nivel = "provincia", departamento = "Madre de Dios",
  grupo = "fauna", limite_por_api = 1000, max_area_ha = 1000
)
head(resultado$ocurrencias)

## End(Not run)
```

buscar_especies_poligono

Busca ocurrencias en un polígono personalizado

Description

Consulta GBIF e iNaturalist sobre un límite aportado por el usuario. El área se normaliza mediante [preparar_poligono_usuario\(\)](#), se consulta por WKT o caja delimitadora y, finalmente, los puntos se recortan contra la geometría exacta localmente. Soporta áreas de estudio, buffers, ANP y polígonos de múltiples partes.

Usage

```
buscar_especies_poligono(
  poligono,
  nombre = NULL,
  nombre_cientifico = NULL,
  grupo = NULL,
  limite_por_api = configuracion_predeterminada()$limite_por_api,
  guardar_resultados = FALSE,
  dir_salida = NULL,
  tolerancia_simplificacion = configuracion_predeterminada()$tolerancia_simplificacion_m,
  estrategia_espacial = c("auto", "directa", "segmentada"),
  max_area_ha = configuracion_predeterminada()$max_area_ha_por_lote,
  max_lotes = configuracion_predeterminada()$max_lotes_espaciales,
```

```
cache_dir = ruta_cache("consultas_ocurrencias"),
reintentos = configuracion_predeterminada()$reintentos_api,
pausa_entre_lotes_s = configuracion_predeterminada()$pausa_entre_lotes_s
)
```

Arguments

poligono	Objeto sf, sfc o Spatial, o ruta de longitud uno a un archivo .shp, .geojson, .gpkg o .kml. Debe tener geometría poligonal; los elementos múltiples se disuelven en un único límite.
nombre	NULL o etiqueta de texto para identificar la consulta. Si es NULL, se usa el nombre del archivo o "Poligono_Personalizado".
nombre_cientifico	NULL o nombre de especie/grupo taxonómico. Se usa para filtrar ambas fuentes, aunque cada API puede resolver sinónimos de forma distinta.
grupo	NULL, "flora" o "fauna"; filtra por Plantae o Animalia.
limite_por_api	Entero entre 1 y 10000, o NULL. Es un máximo por API y tesela; no es el máximo global. NULL intenta recuperar todas las filas solamente si los límites técnicos de ambas APIs lo permiten.
guardar_resultados	Lógico. Con TRUE exporta los resultados en la carpeta processed/ configurada con <code>peruocc_data_dir()</code> .
dir_salida	Ruta de destino si guardar_resultados = TRUE.
tolerancia_simplificacion	Número no negativo, en metros, usado para acortar la geometría WKT de GBIF. El filtro espacial final usa siempre la geometría original.
estrategia_espacial	Una de "auto", "segmentada" o "directa". "auto" equivale a "segmentada" y divide polígonos grandes; "directa" evita teselas y solo es aconsejable para áreas pequeñas.
max_area_ha	Área positiva, en hectáreas, objetivo de cada tesela para estrategia segmentada. El valor 1000 equilibra tamaño de petición y número de llamadas; reduzca este valor ante errores por volumen.
max_lotes	Número máximo de macro-bloques espaciales generados por unidad geográfica para evitar saturar las cuotas de las APIs.
cache_dir	Directorio escribible para checkpoints de resultados por fuente/lote. Conservarlo permite reanudar una extracción interrumpida.
reintentos	Entero positivo con el número máximo de reintentos de llamadas remotas transitorias.
pausa_entre_lotes_s	Número no negativo de segundos de espera entre lotes. Aumentarlo es útil ante respuestas de límite de tasa.

Value

Lista con unidad_sf, ocurrencias, resumen y parametros. resumen\$fallos_lotes indica si alguna fuente/lote no pudo completarse.

Examples

```
coords <- matrix(c(-77.05, -12.10, -77.01, -12.10, -77.01, -12.05,
                  -77.05, -12.05, -77.05, -12.10), ncol = 2, byrow = TRUE)
zona <- sf::st_as_sf(sf::st_sfc(sf::st_polygon(list(coords))), crs = 4326))
## Not run:
resultado <- buscar_especies_poligono(zona, nombre = "Zona de prueba",
                                     grupo = "flora", limite_por_api = 500)

## End(Not run)
```

buscar_especies_provincia

Busca ocurrencias en una provincia peruana

Description

Atajo de [buscar_especies_peru\(\)](#) con nivel = "provincia". Con la estrategia predeterminada procesa los distritos de forma independiente y consolida al final, una opción más recuperable que consultar la provincia disuelta en una sola petición.

Usage

```
buscar_especies_provincia(
  provincia,
  departamento = NULL,
  nombre_cientifico = NULL,
  grupo = NULL,
  limite_por_api = configuracion_predeterminada()$limite_por_api,
  guardar_resultados = FALSE,
  tolerancia_simplificacion = configuracion_predeterminada()$tolerancia_simplificacion_m,
  ...
)
```

Arguments

provincia	Cadena no vacía con el nombre de la provincia.
departamento	NULL o el departamento que contiene la provincia. Es necesario cuando el nombre es ambiguo.
nombre_cientifico	NULL o nombre de taxón para filtrar.
grupo	NULL, "flora" o "fauna".
limite_por_api	Entero entre 1 y 10000, o NULL; se aplica a cada fuente y lote.
guardar_resultados	Lógico; si es TRUE exporta los resultados finales.
tolerancia_simplificacion	Tolerancia para simplificación de WKT de GBIF, medida en metros.
...	Controles avanzados reenviados a buscar_especies_peru() . Destacan max_area_ha para ajustar las teselas y cache_dir para reanudar.

Value

Lista con límite provincial disuelto, ocurrencias consolidadas, resumen de lotes y parámetros de la ejecución.

Examples

```
## Not run:
res <- buscar_especies_provincia("Urubamba", departamento = "Cusco", grupo = "fauna")
head(res$ocurrencias)

## End(Not run)
```

exportar_resultados	<i>Exporta un resultado de búsqueda a formatos interoperables</i>
---------------------	---

Description

Escribe las ocurrencias consolidadas como tabla CSV, capa GeoJSON y/o un manifiesto JSON de reproducibilidad. El manifiesto registra parámetros, geometría, versiones de paquetes y las rutas creadas. No se genera ningún archivo si resultado\$ocurrencias no contiene filas.

Usage

```
exportar_resultados(
  resultado,
  dir_salida = NULL,
  prefijo = NULL,
  formatos = c("csv", "geojson", "manifiesto")
)
```

Arguments

resultado	Lista producida por una función buscar_especies_*. Debe contener al menos ocurrencias, resumen, parametros y unidad_sf.
dir_salida	Ruta del directorio de destino. Si es NULL, usa processed/ dentro de peruocc_data_dir() . Si no se ha configurado un directorio, debe especificarse explícitamente (por ejemplo, tempdir()).
prefijo	Cadena opcional para el identificador de archivos. Con NULL se forma uno con fecha UTC, nivel, unidad y grupo. No incluya extensión: esta función añade .csv, .geojson o .json.
formatos	Vector no vacío formado por "csv", "geojson" y/o "manifiesto". El CSV mantiene todas las filas; el GeoJSON omite filas sin longitud o latitud finitas.

Value

Invisiblemente, una lista nombrada con las rutas creadas. Los nombres posibles son csv, geojson y manifiesto.

Examples

```
## Not run:
resultado <- buscar_especies_distrito("Miraflores", departamento = "Lima")
exportar_resultados(resultado, dir_salida = tempdir(), formatos = c("csv", "manifiesto"))

## End(Not run)
```

graficar_ocurrencias *Grafica ocurrencias sobre su área de consulta*

Description

Construye un mapa ggplot2 con el polígono consultado y los registros que quedaron después del filtro espacial exacto. Es apropiada para inspección exploratoria y control de calidad de coordenadas, no para cartografía final.

Usage

```
graficar_ocurrencias(
  resultado_lista,
  color_por = "source",
  guardar_mapa = FALSE,
  ruta_salida = NULL
)
```

Arguments

resultado_lista	Lista devuelta por buscar_especies_peru() , buscar_especies_distrito() , buscar_especies_provincia() o buscar_especies_poligono() . Debe contener unidad_sf, ocurrencias y resumen.
color_por	Cadena con la columna usada para colorear puntos. Los valores admitidos son "source" (GBIF/iNaturalist, predeterminado) y "kingdom" (Plantae/Animalia cuando está disponible).
guardar_mapa	Lógico de longitud uno. Si es TRUE, además devuelve el gráfico y lo guarda como PNG. Requiere especificar ruta_salida o haber configurado peruocc_data_dir() .
ruta_salida	Ruta completa de archivo donde guardar la imagen PNG cuando guardar_mapa = TRUE. Si es NULL y peruocc_data_dir() está configurado, se genera automáticamente en la subcarpeta results/.

Value

Un objeto de clase ggplot. Puede añadirse capas o temas de ggplot2 antes de imprimirlo.

Examples

```
## Not run:
resultado <- buscar_especies_distrito("Miraflores", departamento = "Lima")
graficar_ocurrencias(resultado, color_por = "source")

## End(Not run)
```

obtener_poligono_distrito

Obtiene el límite oficial de un distrito peruano

Description

Descarga o recupera del caché la capa distrital de geoperu, localiza la unidad solicitada sin distinguir mayúsculas ni tildes y devuelve una geometría válida en WGS84. Es la forma recomendada de inspeccionar un límite antes de una búsqueda o de resolver ambigüedades administrativas.

Usage

```
obtener_poligono_distrito(distrito, departamento = NULL, provincia = NULL)
```

Arguments

distrito	Cadena no vacía con el nombre oficial o usual del distrito. La coincidencia ignora tildes y mayúsculas; no se aceptan códigos UBIGEO.
departamento	NULL o cadena con uno de los 25 departamentos del Perú. Recomendado para nombres de distrito repetidos y para evitar descargar la capa nacional completa.
provincia	NULL o cadena con la provincia que contiene el distrito. Se combina con departamento para desambiguar. Si persisten varias coincidencias, la función muestra alternativas y se detiene.

Value

Un objeto sf de una fila, EPSG:4326, con columnas departamento, provincia, distrito, capital (cuando esté disponible) y geometría.

Examples

```
## Not run:
miraflores <- obtener_poligono_distrito(
  distrito = "Miraflores", departamento = "Lima", provincia = "Lima"
)

## End(Not run)
```

`obtener_poligono_provincia`*Obtiene el límite oficial de una provincia peruana*

Description

Recupera los distritos de la provincia desde geoperu y disuelve sus geometrías en una sola entidad válida. Para descargar ocurrencias provinciales use `buscar_especies_provincia()`, que internamente conserva los distritos separados para hacer consultas más resilientes.

Usage

```
obtener_poligono_provincia(provincia, departamento = NULL)
```

Arguments

<code>provincia</code>	Cadena no vacía con el nombre de la provincia. La búsqueda no distingue tildes ni mayúsculas.
<code>departamento</code>	NULL o cadena con el departamento que contiene la provincia. Es obligatorio cuando el nombre existe en más de un departamento.

Value

Un objeto sf de una fila en EPSG:4326, con departamento, provincia, distrito (NA) y la geometría disuelta.

Examples

```
## Not run:  
urubamba <- obtener_poligono_provincia("Urubamba", departamento = "Cusco")  
  
## End(Not run)
```

`obtener_poligono_unidad`*Obtiene un límite administrativo mediante una interfaz única*

Description

Despacha a `obtener_poligono_distrito()` o `obtener_poligono_provincia()` según nivel. Facilita crear funciones genéricas cuando el nivel de consulta se elige en tiempo de ejecución.

Usage

```
obtener_poligono_unidad(
  nombre,
  nivel = c("distrito", "provincia"),
  departamento = NULL,
  provincia = NULL
)
```

Arguments

nombre	Cadena no vacía. Es el nombre del distrito cuando nivel = "distrito" o el de la provincia cuando nivel = "provincia".
nivel	Uno de "distrito" o "provincia". Si se suministra más de un valor, se usa el primero mediante <code>match.arg()</code> .
departamento	NULL o nombre del departamento para limitar la búsqueda y resolver homónimos.
provincia	NULL o nombre de provincia; solo se usa con nivel = "distrito".

Value

Un objeto `sf` en EPSG:4326. Para provincias la geometría está disuelta; para distritos contiene una fila de la capa oficial.

Examples

```
## Not run:
limite <- obtener_poligono_unidad("Tarapoto", nivel = "distrito",
                                  departamento = "San Martin")

## End(Not run)
```

peruocc_data_dir

Configura el directorio de trabajo de peruocc

Description

Define el directorio raíz donde el paquete guarda resultados exportados y, cuando se configura explícitamente, los límites y checkpoints de consultas. La configuración se conserva durante la sesión de R mediante la opción `peruocc.data_dir`; no modifica archivos de configuración permanentes ni crea directorios por defecto en el espacio de trabajo del usuario.

Usage

```
peruocc_data_dir(path = NULL)

peruspecies_data_dir(path = NULL)
```

Arguments

path Cadena de longitud uno con una ruta existente o por crear. Debe apuntar a una ubicación con permisos de escritura. Si es NULL, devuelve la ruta configurada actualmente (o NULL si no se ha configurado ninguna).

Details

Cuando está configurado, los resultados se escriben en `processed/` y los checkpoints en `cache/` dentro de este directorio.

Value

Invisiblemente, la ruta absoluta normalizada activa, o NULL si no se ha definido un directorio.

Examples

```
dir_temporal <- file.path(tempdir(), "peruocc-ejemplo")
peruocc_data_dir(dir_temporal)
# consultar la ruta activa:
peruocc_data_dir()
```

```
preparar_poligono_usuario
```

Valida y normaliza un polígono aportado por el usuario

Description

Acepta una geometría u archivo espacial, lo transforma a WGS84, corrige topología cuando es posible y unifica múltiples elementos en un único límite. Es la preparación previa que usa [buscar_especies_poligono\(\)](#).

Usage

```
preparar_poligono_usuario(poligono, nombre = NULL)
```

Arguments

poligono Un objeto sf, sfc o Spatial, o una ruta de longitud uno a .shp, .geojson, .gpkg o .kml. Debe contener geometrías poligonales. Si no tiene CRS se asume EPSG:4326 y se emite una advertencia.

nombre NULL o una etiqueta de texto no vacía para resultados y exportaciones. Si poligono es una ruta y nombre es NULL, se usa el nombre del archivo sin extensión; para objetos espaciales se usa "Poligono_Personalizado".

Value

Un objeto sf válido de una fila en EPSG:4326, con las columnas unidad, distrito, provincia y departamento. Las tres últimas se rellenan con NA porque el límite no procede de una unidad administrativa.

Examples

```
coords <- matrix(c(-77.05, -12.10, -77.01, -12.10, -77.01, -12.05,  
                  -77.05, -12.05, -77.05, -12.10), ncol = 2, byrow = TRUE)  
zona <- sf::st_as_sf(sf::st_sfc(sf::st_polygon(list(coords))), crs = 4326)  
preparar_poligono_usuario(zona, nombre = "Zona de prueba")
```

```
verificar_y_configurar_entorno
```

Verifica las dependencias de peruocc

Description

Comprueba la disponibilidad de los paquetes requeridos para límites administrativos, operaciones espaciales, consultas a GBIF/iNaturalist, visualización y exportación. Úsela al preparar una instalación nueva o para diagnosticar un error de carga.

Usage

```
verificar_y_configurar_entorno()
```

Value

Invisiblemente TRUE si todas las dependencias están disponibles.

Examples

```
verificar_y_configurar_entorno()
```

Index

[.peruocc_tbl (as_peruocc_tbl), 2

as_peruocc_tbl, 2

buscar_especies_distrito, 3

buscar_especies_distrito(), 10

buscar_especies_peru, 4

buscar_especies_peru(), 3, 4, 8, 10

buscar_especies_poligono, 6

buscar_especies_poligono(), 10, 14

buscar_especies_provincia, 8

buscar_especies_provincia(), 10, 12

exportar_resultados, 9

exportar_resultados(), 5

graficar_ocurrencias, 10

obtener_poligono_distrito, 11

obtener_poligono_distrito(), 12

obtener_poligono_provincia, 12

obtener_poligono_provincia(), 12

obtener_poligono_unidad, 12

peruocc_data_dir, 13

peruocc_data_dir(), 7, 9, 10

peruspecies_data_dir

(peruocc_data_dir), 13

preparar_poligono_usuario, 14

preparar_poligono_usuario(), 6

print.peruocc_tbl (as_peruocc_tbl), 2

verificar_y_configurar_entorno, 15