

MANUAL

DE PROCEDIMIENTOS AUTOMATIZADOS PARA LA CURACIÓN DE BASES DE DATOS SIGUIENDO EL ESTÁNDAR DARWIN CORE



MANUAL DE PROCEDIMIENTOS AUTOMATIZADOS PARA LA CURACIÓN DE BASES DE DATOS SIGUIENDO EL ESTÁNDAR DARWIN CORE

Edición:

Equipo Proyecto GEF Corredores Biológicos de Montaña (GEFSEC ID 5135)

Encargado a:

M.Sc. Matías Rebolledo González

Con la colaboración de:

MMA, División de Recursos Naturales y Biodiversidad

Fotografía de portada: *Escallonia pulverulenta*, Marianne Katunarić

Fotografía de contraportada: *Eriocyce curvispina*, Marianne Katunarić

Desarrollado y financiado por:

Proyecto GEFSEC ID 5135 “Protegiendo la Biodiversidad y Múltiples Servicios Ecosistémicos en Corredores Biológicos de Montaña, en el Ecosistema Mediterráneo de Chile. Ministerio del Medio Ambiente – ONU Medio Ambiente (2016-2022).

Citar este documento como:

MMA y ONU Medio Ambiente. (2019). Manual de procedimientos automatizados para la curación de bases de datos siguiendo el estándar Darwin core. Encargado a: M.Sc. Matías Rebolledo González. Financiado por: Proyecto GEFSEC ID 5135 Ministerio del Medio Ambiente - ONU Medio Ambiente, Santiago, Chile. 89 pp.

ÍNDICE DE CONTENIDOS

1. INTRODUCCIÓN	6
2. METODOLOGÍA.....	8
2.1. DISEÑO DEL PROCESO	8
3. RESULTADOS Y PRODUCTOS	13
3.1. ORGANIZACIÓN DE RUTINAS DE VALIDACIÓN DE REGISTROS TAXONÓMICOS.....	13
3.1.1. Primer set de algoritmos: Definición de insumos	13
3.1.2. Segundo set de algoritmos: Funciones	14
3.1.3. Tercer set de algoritmos: Secuencias lógicas.....	15
3.2. VISIÓN AL DISEÑO REPRODUCIBLE DE CADA PROCESO	16
3.3. ANÁLISIS DETALLADO DE CADA ETAPA POR MÓDULO	19
3.4. GENERACIÓN DE SUB-PRODUCTOS DURANTE LA CURACIÓN DE BASES TAXONÓMICAS	24
3.4.1. Base taxonómica Darwinion.....	24
3.4.2. Base taxonómica de Rodríguez <i>et al.</i> (2018).....	26
4. DISCUSIÓN Y SEGUIMIENTO.....	28
5. APÉNDICES.....	31
APÉNDICE 1	31
BashVariablesFecha.sh	31
BashVariablesCoordenadas.sh	35
BashVariablesTaxonomia.sh	44
APÉNDICE 2	46
FuncionesFecha.sh	46
FuncionesCoordenadas.sh	56
FuncionesTaxonomia.sh.....	67

APÉNDICE 3	69
ProcesarFecha.sh	69
ProcesarCoordenada.sh	77
ProcesarTaxonomia.sh	86

ÍNDICE DE CUADROS

Cuadro 1. Número de archivos con variables a ser procesadas en el primer set de algoritmos.....	14
Cuadro 2. Número de funciones generadas en el segundo set de algoritmos.....	15
Cuadro 3. Número de procesos generados con el tercer set de algoritmos.	16
Cuadro 4. Número de ejecuciones del módulo de coordenadas según archivo base de datos.....	20
Cuadro 5. Número de ejecuciones del módulo de fechas según archivo base de datos.	21
Cuadro 6. Número de ejecuciones del módulo de taxonomía según archivo base de datos.....	22

ÍNDICE DE FIGURAS

Figura 1. Diagrama de organización de rutinas para validación de registros taxonómicos.	10
Figura 2. Diagrama de proceso para validación de fechas.	17
Figura 3. Diagrama resumido de validación de coordenadas.....	18
Figura 4. Diagrama resumido de validación de taxonomía.	19
Figura 5. Diagrama resumido de módulo de extracción Darwinion.	25

A close-up photograph of several bright yellow Calceolaria flowers. The flowers have a distinctive pouch-like shape and are clustered together on green stems. A large yellow diagonal shape is overlaid on the left side of the image, containing the text '01' and 'INTRODUCCIÓN'.

01

INTRODUCCIÓN

Foto: *Calceolaria* sp., por Marianne Katunarić

1. INTRODUCCIÓN

Los procesos de automatización de limpieza y evaluación de integridad de bases de datos de colecciones biológicas que se presentan en este manual, tienen como objetivo facilitar la curación de este tipo de bases de datos cumpliendo con el estándar Darwin Core (DWC) requerido para su publicación en la plataforma internacional GBIF. Específicamente, el objetivo de las rutinas desarrolladas es poder generar insumos, reportes, efectuar consultas, aplicar patrones de corrección, de manera automatizada, ofreciendo así al analista experto en taxonomía un insumo más depurado, disminuyendo su tiempo en revisión de antecedentes y agilizando su trabajo de validación para publicación de estas bases de datos. Es decir, el curador de las colecciones recibirá trabajo adelantando, a partir del cual sólo deberá revisar detenidamente aquellos registros que presentan inconsistencias o modificaciones en sus nombres, dadas las actualizaciones taxonómicas que ocurren regularmente. Usando este enfoque, se elimina el exceso de trabajo manual que, en el caso de trabajar con información compleja y extensa, requiere mucho tiempo, exactitud, y tiende a generar una gran cantidad de “errores humanos”. Este esfuerzo se enmarca en el ensayo piloto a escala regional del Sistema de Información y Monitoreo de Biodiversidad (SIMBIO) del Ministerio del Medio Ambiente, para el área del Proyecto GEF Corredores Biológicos de Montaña.

02

METODOLOGÍA



Foto: *Oxalis* sp., por Bárbara Von Igel

2. METODOLOGÍA

2.1. DISEÑO DEL PROCESO

Para el proceso de curación de bases de datos automatizado se aplicó una lógica iterativa y adaptativa. Inicialmente se trabajó en diseñar un flujo de limpieza de bases de datos automático, para ser ejecutado por un analista de información, con el propósito de optimizar el protocolo de automatización. Es decir, se hizo un diseño más pausado, mostrando resultados parciales de cada etapa intermedia como medio para un diseño posterior más completo y eficiente. En una segunda etapa, se realizó un rediseño de estos procesos, que tuvo la finalidad de enlazar cada una de las etapas en un flujo completo e integrado, escondiendo resultados intermedios de manera de tener una fuente de información inicial sin correcciones y una fuente de información final con recomendaciones para su curación; la que posteriormente se pone a disposición del analista taxónomo o curador de colecciones. Una tercera etapa de la automatización del proceso de curación de bases de datos para formato Darwin Core, consistió en la generación de varios sub-productos, particularmente en el caso de los campos relacionados con la taxonomía, para los cuales se obtuvo una variedad de nombres científicos desde diferentes fuentes de información. Se trabajó con tres bases de listados de especies, dos internacionales (Species Matching y Taxize) y validados por GBIF¹ y uno específico para la flora del Conosur (Darwinion²), por lo cual la curación taxonómica se hizo en dos etapas:

- a) Se contrastó el nombre científico de la colección de los herbarios con los catálogos mundiales (GBIF). Dado que GBIF ya cuenta con herramientas y API's³ para realizar estas comparaciones fue relativamente simple automatizar esta comparación. En este caso el cruce de información cuenta con un método que nos permite encontrar registros en las bases de datos GBIF, incluso si la información tiene inconsistencias.
- b) Se contrastó el nombre científico verificado en la etapa a) con el catálogo regional (Darwinion).

¹ Global Biodiversity Information Facility (GBIF) disponible en <https://www.gbif.org/es/>

² Catálogo de las Plantas Vasculares del Conosur disponible en <http://www.darwin.edu.ar/>

³ API: Interfaz de Programación de Aplicaciones por sus siglas en inglés.

Además, se trabajó en la extracción de las fichas botánicas del catálogo de Rodríguez *et al.* (2018)⁴ para en un futuro poder contar con dicho listado de especies locales e incluirlo en el proceso de automatización. Dado que no existen API's que apoyen la comparación entre los catálogos locales y una base de datos, habiendo sido necesario extraer la información de estos catálogos desde sus páginas web o documentos, no se realizó contrastación directa con los nombres científicos de la colección de herbarios. El desarrollo de herramientas para esta comparación habría requerido una programación que escape los alcances y tiempos de la presente consultoría, siendo no obstante posible desarrollarlas en el futuro.

Para el desarrollo de las etapas descritas se trabajó con tres bases de datos, dos del Museo Nacional de Historia Natural (MNHN) correspondientes al catálogo de especies de la Región de Valparaíso y de la Región Metropolitana de Santiago, y otra base de datos del Herbario de la Universidad de Concepción. Se usó herramientas Unix tools, el lenguaje interpretado Bash, Python y R, usando un enfoque de programación de flujo muy similar al concepto ETL (Extract Transform and Load). La diferencia con el concepto ETL, es que en ese caso, la mayoría de los métodos son aplicados a información almacenada en bases de datos con estructura relacional definida, mientras que en este caso, se trabajó principalmente con información no estructurada en archivos de texto simple. Particularmente en el caso de los campos taxonómicos, la información no poseía ningún tipo de estructura, justificando el uso de las herramientas descritas que entregan mayor flexibilidad que las herramientas ETL del mercado. La Figura 1 presenta un diagrama resumen de todo el proceso, indicando cuál fue la organización metodológica de las rutinas para separar en cada tipo de archivo una lógica diferente. En [Resultados y productos](#) se presenta en detalle en qué consiste cada objeto identificado y un diagrama resumen del proceso. Además, en [Apéndices](#) se adjuntan los códigos que están representados por nombre_de_archivo.sh.

⁴ Rodríguez R., C. Marticorena, D. Alarcón, C. Baeza, L. Cavieres, V.L. Finot, N. Fuentes, A. Kiessling, M. Mihoc, A. Pauchard, E. Ruiz, P. Sanchez & A. Marticorena. (2018). Catálogo de las plantas vasculares de Chile. *Gayana Bot.* 75(1): 1-430

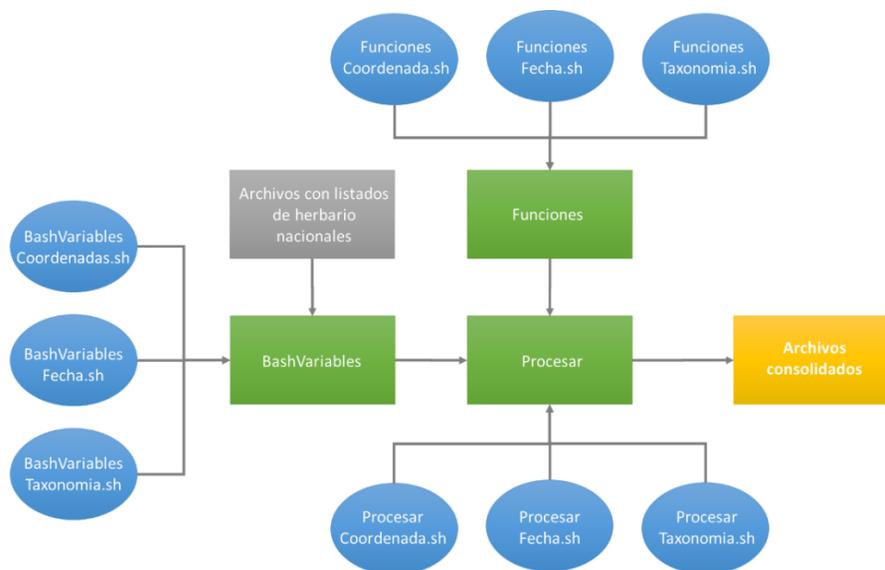


Figura 1. Diagrama de organización de rutinas para validación de registros taxonómicos.

Por otra parte, el denominador común de las herramientas seleccionadas para el desarrollo de este proceso de automatización es que pueden integrarse secuencialmente, en forma de módulos independientes, dentro del entorno Unix. Para otro tipo de plataformas como Windows es posible ejecutar el proceso de validación desarrollado en el entorno Docker, que es una herramienta que permite crear aplicaciones aisladas en contenedores que albergan un sistema operativo con lo mínimo necesario para su funcionamiento de manera autónoma. Todo lenguaje que se escoge para desarrollarlo en Docker, tiene la ventaja de no tener problemas de portabilidad.

En cuanto a la elaboración de este procedimiento hay algunas consideraciones a tener presente:

- Se hizo con un sistema de cómputo debajo de escenas, esto implica que parte de los desarrollos son evaluados en conjunto con el desarrollo del manual, para efectos de analizar resultados en tiempo real y tener una bitácora de avance del desarrollo de cada módulo. Tanto R como Python cuentan con su propio sistema de documentación. Se eligió para este manual el sistema de

documentación de R, pero este no excluye la posibilidad de evaluar sentencias en Python o Unix tools conforme se requiera.

- El sistema operativo en el que se desarrollaron las implementaciones es Linux Debian 9, y el sistema Python que se eligió es la distribución Anaconda. Esto implica que para portarlo al sistema Windows, se debe escoger una imagen Docker que contenga el mismo sistema operativo y versión.
- Se trabajó con el concepto de módulo encadenable y varias herramientas de administración de sistema Linux, que pueden usarse para procesar información alfanumérica. Una potencialidad es el uso de expresiones regulares para diseñar filtros avanzados de validación. Varios de estos aspectos son empleados en el desarrollo de cada módulo.
- La librería Panda está hecha para trabajar con “dataframes”, equivalentes a planillas Excel diseñados para ser manipuladas en un entorno programable sencillo y versátil. El lenguaje R también contiene manipulación de “dataframes”, por lo que para tareas puntuales se utiliza uno u otro lenguaje, según convenga.

03

RESULTADOS Y
PRODUCTOS



Foto: *Viola atropurpurea*, por Bárbara Von Igel

3. RESULTADOS Y PRODUCTOS

3.1. ORGANIZACIÓN DE RUTINAS DE VALIDACIÓN DE REGISTROS TAXONÓMICOS

A continuación se describen brevemente los productos de programación intermedios que se han generado en esta etapa y se finaliza con una representación gráfica del diagrama de flujo de los productos intermedios. Se aplicó una estrategia inteligente de separación de rutinas, con un primer grupo que contiene variables (que corresponden a los insumos con los que se trabaja en cada etapa como archivos, listados de variables, etc.), un segundo grupo de archivos que contiene las funciones (corresponden a los procesos que usan como argumentos, las variables previamente definidas en cada etapa) y ejecuciones (son la secuencia lógica en la cual se aplican las funciones a las variables etapa por etapa). Este flujo se asemeja al concepto ETL (Extract, Transform and Load) como se presentó en [Metodología](#).

3.1.1. Primer set de algoritmos: Definición de insumos

El primer set de algoritmos corresponde a las definiciones de los insumos que son usados en cada una de las etapas de cada módulo. Se tomó como ejemplo el módulo de coordenadas. En este caso, se definió un total de 108 variables, las que principalmente corresponden a archivos iniciales, intermedios y finales. Si se considera que se aplicó a 3 bases de datos (catálogo de especies de la Región de Valparaíso, Región Metropolitana de Santiago y de Concepción), se tiene un total de $(108/3)=36$ variables definidas para cada una de las 3 bases de datos a las que se les aplicó la validación de coordenadas. Es decir, fue requerido definir 1 archivo inicial, 34 archivos intermedios y 1 archivo final para aplicar cada uno de los pasos de validación de coordenadas. El número de 36 variables representa un promedio, puesto que algunas bases de datos no requieren de la misma cantidad de pasos (108 fue el número total definido para las 3 bases de herbarios). Las rutinas corresponden a aquellas que comienzan con prefijo “BashVariables”, así como, “BashVariablesCoordenada.sh” para el módulo de coordenadas en cuestión. En el [Apéndice 1](#) se puede consultar los archivos correspondientes que, a saber, son los siguientes: BashVariablesCoordenadas.sh, BashVariablesFecha.sh y BashVariablesTaxonomia.sh. Se trabajó en forma modular, buscando que cada función pueda ser reutilizada la mayor cantidad de veces en toda etapa que se requiera. Sin embargo, durante el desarrollo de los procedimientos hubo etapas y conversiones muy específicas, por lo cual se crearon nuevas funciones, así como una mayor cantidad de etapas intermedias. Esta tendencia es propia del trabajo de



validación y curación, debiéndose a un gran número de casos atípicos donde es difícil incorporar los pasos intermedios en una función genérica. Estos casos requieren inevitablemente una función específica para aplicar su respectivo patrón de corrección. En el Cuadro 1 se puede encontrar el número de archivos generados en este set de algoritmos para cada uno de los módulos de validación.

Cuadro 1. Número de archivos con variables a ser procesadas en el primer set de algoritmos.

Módulo	Rutina	Número archivos
Coordenada	BashVariablesCoordenada.sh	108
Fecha	BashVariablesFecha.sh	52
Taxonomia	BashVariablesTaxonomia.sh	51
Darwinion	BashVariablesDarwinion.sh	3

El último aspecto importante de mencionar, es que algunos archivos definidos en las rutinas de variables, almacenan la ubicación de listados de archivos a los que se aplicó un proceso en conjunto. Este fue el caso cuando se aplicaron procesos por lotes, lo que se conoce comúnmente como modo “batch” o modo lista. Esto implica que el número de archivos intermedios fue mayor al presentado en el Cuadro 1, puesto que algunos archivos contenían procesos por lotes aplicados a miles de archivos intermedios.

3.1.2. Segundo set de algoritmos: Funciones

El segundo set de algoritmos corresponde a las funciones que se aplican a las variables a ser procesadas y llevan el prefijo “Funciones”. La lógica del Cuadro 2 es similar al anterior. En este caso, se tiene que el módulo de coordenadas requirió de 18 funciones en total para aplicar la validación en este ámbito. En este caso, el número de 18 no puede dividirse en 3 bases de datos, porque son definiciones que aplican para todo el módulo de validación, pero sí es posible que alguna base no requiera la aplicación de las 18 funciones, o que una misma función sea utilizada más de una vez. Estos archivos pueden encontrarse en el [Apéndice 2](#) con los nombres: FuncionesCoordenadas.sh, FuncionesFecha.sh, FuncionesTaxonomia.sh. En el Cuadro 2 se puede encontrar el número de funciones generadas en este set de algoritmos.

Cuadro 2. Número de funciones generadas en el segundo set de algoritmos.

Módulo	Rutina	Número funciones
Coordenada	FuncionesCoordenada.sh	18
Fecha	FuncionesFecha.sh	24
Taxonomia	FuncionesTaxonomia.sh	16
Darwinion	FuncionesDarwinion.sh	4
Varias	BashFuncionesVarias.sh	31

En el Cuadro 2 se puede apreciar una rutina que no tiene equivalente en el Cuadro 1. Se trata de “BashFuncionesVarias.sh”. Este archivo contiene funciones que son usadas libremente en cualquiera de los tres módulos (coordenadas, fechas y taxonomía) incluyendo el sub módulo (Darwinion), es decir contiene 31 funciones adicionales que son de utilidad para todos los módulos desarrollados.

3.1.3. Tercer set de algoritmos: Secuencias lógicas

El tercer set de algoritmos corresponde a secuencias lógicas en las cuales se deben aplicar las funciones a las variables declaradas, y llevan el prefijo “Procesar”. Siguiendo con el ejemplo de los herbarios, el módulo de coordenadas consistió de $(212/3)=70$ etapas requeridas para validar las coordenadas en cada una de las tres bases de datos de los herbarios. Estos algoritmos se encuentran en el [Apéndice 3](#) bajo los nombres: ProcesarCoordenadas.sh, ProcesarFechas.sh y ProcesarTaxonomia.sh. Con estos códigos se obtiene el producto final, que corresponde al listado de fechas, coordenadas y registros con sus nombres científicos verificados o con propuestas nuevas según haya sido el resultado con la comparación con los catálogos internacionales y nacionales. En el Cuadro 3 se puede encontrar el número de procesos generados en este set de algoritmos.

Cuadro 3. Número de procesos generados con el tercer set de algoritmos.

Módulo	Rutina	Número ejecuciones
Coordenada	ProcesarCoordenada.sh	212
Fecha	ProcesarFecha.sh	139
Taxonomia	ProcesarTaxonomia.sh	96
Darwinion	ProcesarDarwinion.sh	5

3.2. VISIÓN AL DISEÑO REPRODUCIBLE DE CADA PROCESO

Una de las razones de rediseñar el trabajo de los procesos hechos con anterioridad fue el separar en tres lógicas distintas los códigos para generar un diagrama de proceso, lo que forma parte de la documentación de todo el proceso. A pesar de haber diseñado varios diagramas de proceso tentativos para cada módulo, se llegó a la conclusión que lo requerido es una forma automática de mostrar gráficamente las secuencias que se aplicaron considerando el gran número de etapas y archivos intermedios.

La Figura 2 presenta el módulo de validación de fechas con las fórmulas aplicadas en cada paso, los archivos de entrada y salida en cada etapa, mostrando un gran número de archivos intermedios requeridos para aplicar operaciones simples, pero encadenadas con un orden de prioridad secuencial.

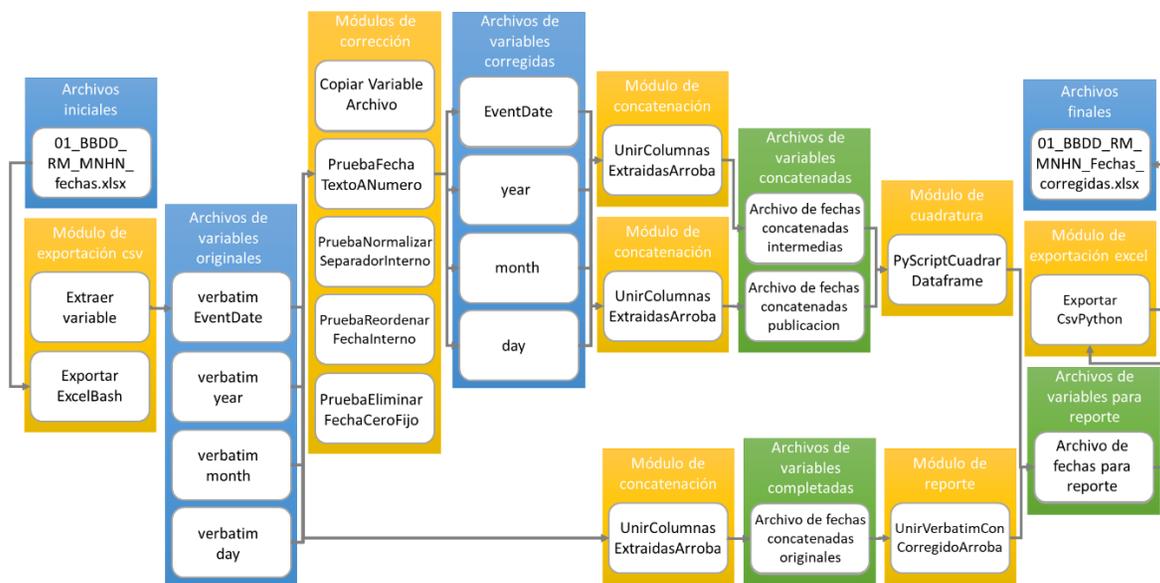


Figura 2. Diagrama de proceso para validación de fechas.

En la Figura 3 se presenta el proceso de validación resumido de coordenadas. Este proceso comparte una característica en común con el de fechas, que es la necesidad de cuadrar la cantidad de registros que fueron originalmente ingresados en diferentes formatos, lo que hizo crecer el proceso en tamaño y complejidad. Como en todos los módulos desarrollados, no se trató únicamente de identificar inconsistencias, sino de operaciones de transformación de estructuras de datos, en donde en cada módulo eso fue acompañado de operaciones específicas. En este caso, las operaciones específicas consistieron en usar una librería matemática de cálculo de reprojección de coordenadas, llamada proj-bin. Con esta se hizo conversión entre los formatos decimal, de grados, minutos y segundos y UTM, respectivamente.

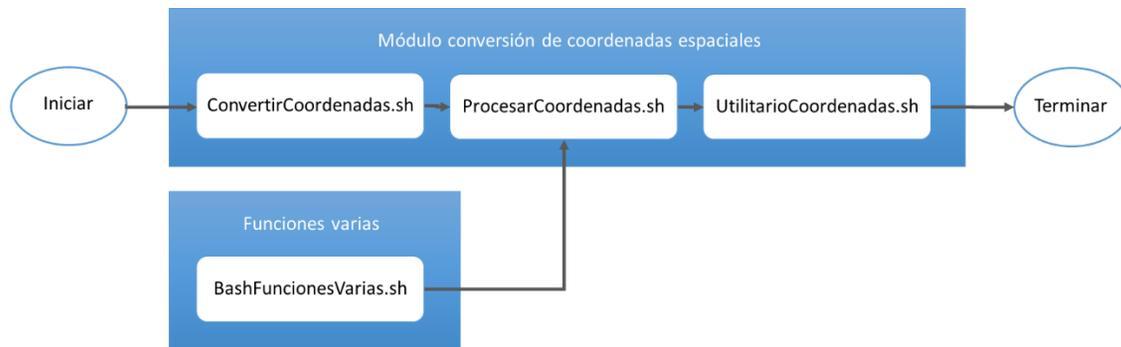


Figura 3. Diagrama resumido de validación de coordenadas.

La Figura 4 presenta el diagrama de proceso de validación de taxonomía. El módulo de taxonomía fue el único que tuvo un énfasis diferente en relación con los demás. Mientras que los demás tuvieron como fin la identificación y corrección de inconsistencias, este tuvo la finalidad de colocar tanto la información original como la información consultada y extraída de servidores de consulta internacional, de manera comparable y sintética, minimizando la presentación original y ampliando la información consultada para que el analista botánico tenga un panorama preciso y sintético sobre las posibles correcciones. Las razones son varias, una de ellas es debido a que la información proviene de muchas fuentes de información y en muchos casos es información que está sujeta a cambios que van más allá de las políticas de administración de la información presente en esos servicios de consulta. Lo que se presenta en la Figura 4, son los pasos correspondientes a la generación del nombre de consulta en modo batch y su posterior consulta usando la API del GBIF y el programa R. En la versión más reciente, se reemplazó el programa R, por la versión pura de la API usando solamente Bash y herramientas Unix, disminuyendo notablemente el consumo de memoria RAM en cada consulta. Como se realizó el mismo paso de consulta usando una segunda herramienta también en R, llamada Taxize, esta no se reemplazó por su versión Bash, puesto que consiste en una batería de APIs de consulta a varios portales, lo que implica convertir varias extracciones en una, pero el consumo de RAM y CPU en esta extracción es más elevado que en la consulta a la API de GBIF.

En este módulo se crearon dos productos de salida, de los cuales se eligió uno finalmente. El primer resultado consistió en 3 bases de datos a nivel de especie con toda la información extraída por los servicios GBIF, Taxize y Darwinion y con comparaciones entre todas esas fuentes para generación de mensajes de la situación del

registro originalmente consultado. Finalmente, de dicho producto se seleccionaron un sub conjunto de columnas de información originalmente consultadas y extraídas desde las fuentes mencionadas, se cambió su estructura a nivel de registro de ocurrencia y se acopló al formato original de los herbarios. Este fue el producto finalmente seleccionado, el cual fue hecho con parte de la información creada para el primer producto.

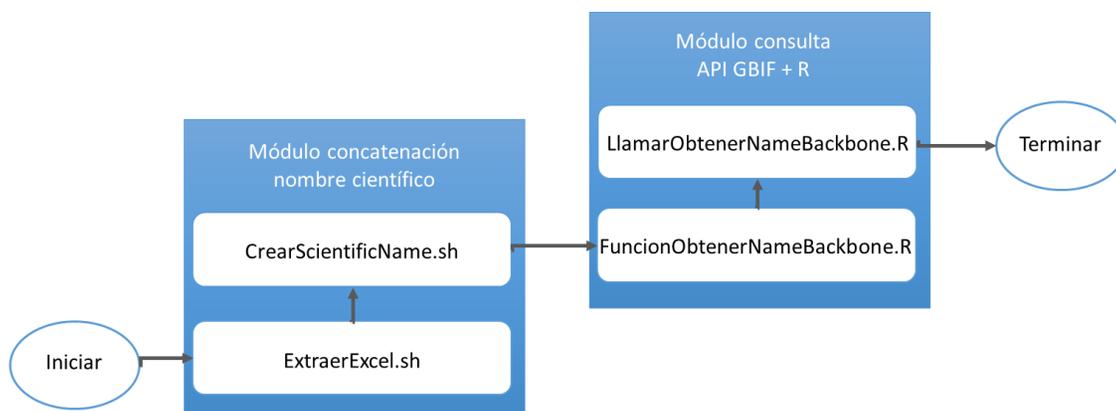


Figura 4. Diagrama resumido de validación de taxonomía.

3.3. ANÁLISIS DETALLADO DE CADA ETAPA POR MÓDULO

A continuación se desglosa un recuento de cada función empleada en cada módulo de validación así como el número de ejecuciones de cada función. En el Cuadro 4 tenemos tanto el desglose de funciones creadas para efectuar esta validación, como un recuento total al final del cuadro, que muestra el número total de ejecuciones, las funciones presentes y el porcentaje de reutilización de cada función. El porcentaje de reutilización se calculó como: $(\text{Ejecuciones totales} - \text{Funciones presentes}) / (\text{Ejecuciones totales}) \times 100$. Su finalidad es mostrar el nivel de funciones que pueden usarse en varias etapas del proceso. En el caso de la validación de coordenadas, este porcentaje se ve elevado considerando que es la función que remueve archivos intermedios la que se ejecuta la mayor cantidad de veces. La idea es mostrar que, dada la existencia de casos atípicos, no es siempre posible usar una misma función en cualquier punto del proceso, siendo necesario crear funciones casi específicas por etapa.

Cuadro 4. Número de ejecuciones del módulo de coordenadas según archivo base de datos.

BD01: Base de Datos del Museo Nacional de Historia Natural Región Metropolitana; BD02: Base de datos del MNHN Valparaíso; BD03: Base de datos de Herbario de Concepción.

Función	Coordenadas		
	BD01	BD02	BD03
ConcatenarDeADosArroba	7	7	5
ConcatenarDeASeisArroba	2	2	
ConvertirCoordenadasGeometria	1	1	1
CopiarVariableArchivo	9	9	2
CuadrarDMSToDDArroba	1	1	
DDToDDUTMArroba	1	1	1
DDToDMSArroba	1	1	1
DMSToDDArroba	1	1	
EliminarRegistroMezclado	8	8	2
ExportarCoordenadasGeometriaArroba	1	1	1
ExportarCsvPython	1	1	1
ExportarShapefilePgsql	1	1	1
ExtraerVariable	8	8	2
ImportarShapefilePgsql	1	1	1
MergeCoordenadaExportada	1	1	1
RemoveArchivoIntermedio	41	41	17
TransferirRegistroMezcladoArroba	2	2	
VerificarCoordenadasArroba	1	1	
Total de ejecuciones	88	88	36
Total de funciones presentes	18	18	13
Porcentaje de reutilización	79,5	79,5	63,9

En el Cuadro 5 se muestra el número de ejecuciones por base de dato para el módulo de fechas. De forma similar, los porcentajes de reutilización son altos, similar al caso de validación de coordenadas.

Cuadro 5. Número de ejecuciones del módulo de fechas según archivo base de datos.

Función	Fechas		
	BD01	BD02	BD03
ConcatenarDeADosArroba	1	1	
CopiarVariableArchivo	5	5	2
ExportarCsvPython	1	1	1
ExtraerVariable	5	5	2
RemoverArchivoIntermedio	14	14	8
ConcatenarDeATresArroba	1	1	2
FormatearDayArroba	1	1	
FormatearMonthArroba	1	1	1
FormatearYearArroba	1	1	1
PyscriptCuadrarDataframe	1	1	
ConcatenarParaCuadraturaArroba	2	2	
PruebaEliminarFechaCeroFijo	5	5	3
PruebaFechaTextoANumero	5	5	2
PruebaNormalizarSeparadorInterno	5	5	2
PruebaReordenarFechaInterno	5	5	2
ConcatenarDeADosUnder			1
PruebaReemplazarCasosErroneosArroba			1
PruebaEliminarCasosErroneosArroba			5
Total de ejecuciones	53	53	33
Total de funciones presentes	15	15	14
Porcentaje de reutilización	71,7	71,7	57,6

Por último, en el Cuadro 6 se muestra el número de ejecuciones totales para el módulo de taxonomía. En este caso la proporción de reutilización fue más bajo, dado que no se removieron archivos intermedios, lo que disminuyó la cantidad de ejecuciones y la brecha entre funciones presentes y total de ejecuciones.

Cuadro 6. Número de ejecuciones del módulo de taxonomía según archivo base de datos.

Función	Taxonomía		
	BD01	BD02	BD03
RemoverArchivoIntermedio	2	2	2
AislarVariableOriginalArroba	1	1	1
ConsolidarResultadoTaxize	1	1	1
ConsultarNuevoIndiceDarwinion	1		
ConsultarTaxizeParalelo	1	1	1
CrearColumnaTituloIndicePagina02	1		
CrearReporteAccion	1	1	1
ExtraccionGbifJsonMultiple	1	1	1
ExtraccionGbifJsonSimpleAutor	1	1	1
ExtraerTaxonCatalogNumber	1	1	1
GenerarColumnaSustraccionAutorGbif	1	1	1
ListarConsultarTaxizeParalelo	1	1	1
ListarPrepararTaxizeArroba	1	1	1
ListarScientificArroba	1	1	1
ModificarConsultaGbifAutor	1	1	1
PrepararNuevoIndiceDarwinion	1		
PrepararTaxizeParalelo	1	1	1
QuitarResultadoExtraGbifTaxize	1	1	1
AislarVariableSpeciesArroba	2	2	2
ConsultarGbifParalelo	2	2	2
GenerarColumnaEstadoEspeciesDarwinion	2	2	2
CopiarArchivoIntermedio	3	1	1

ExportarCsvR	4	4	4
MergeAnyIndicator	6	2	2
Total de ejecuciones	38	29	29
Total de funciones presentes	24	21	21
Porcentaje de reutilización	36,8	27,6	27,6

Los cuadros se presentaron con el fin de apoyar a los diagramas de flujo, que en su mayoría fueron expuestos en forma resumida, mientras que los cuadros dan una estimación del nivel de detalle con que fueron implementadas cada una de las validaciones y el nivel de especificidad de las mismas.

Como observaciones adicionales, en relación las fórmulas de validación se tienen tres tipos, que se aplicaron en cada una de las etapas:

- Un tipo de funciones corrigen inconsistencias, usan en gran medida abreviaciones de texto denominadas expresiones regulares y contienen tanto variedad como número de pasos necesarios para generar insumos para una función de validación. Estas funciones se combinan de manera adecuada para crear un nodo de validación específico.
- El segundo tipo de funciones que se usaron mayormente fueron aquellas que conllevan un cambio en la estructura o forma de disposición de los datos. El cambio puede ser por un lado, en el número de columnas, mientras que por otro puede ser en el número de filas. Como ejemplo de cambio en la disposición y número de columnas se tiene aquellas de los módulos de fechas y coordenadas. En el de coordenadas, el insumo para entrar a un módulo puede contener dos columnas y la salida puede contener seis, siendo esta salida a su vez entrada para un módulo siguiente, el que debe tener en cuenta este cambio de estructura. Similar forma ocurre en el caso de fechas. Por su parte, en el caso de taxonomía tenemos que hay algunos pasos que implican el cambio en el número de registros. Como ejemplo, se tiene que en las funciones de consulta y extracción de información a servidores taxonómicos, en vez de consultar por ejemplo, más de 7.000 registros, solamente se consulta los valores únicos de esas 7.000 filas, por ejemplo 1.500, para posteriormente volver a la estructura original de 7.000. Esto tiene la finalidad de ahorrar consumo conexión a internet y de generar operaciones por lote para ser ejecutadas en paralelo.

- Un tercer tipo de funciones, son aquellas específicas de cada módulo, y que cumplen la función por la cual se concibió un diseño separado en tres ámbitos.

Finalmente, la última característica común en todo el proceso es que el número de funciones tiende a aproximarse al número de pasos totales requeridos en un flujo determinado. Es decir, la cantidad de funciones que pueden reutilizarse dentro del flujo total resulta ser baja. Esto porque cada etapa tiende a tener una necesidad específica y no comparte mucho en común con otros pasos, lo que conlleva prácticamente a desarrollar una función por etapa.

3.4. GENERACIÓN DE SUB-PRODUCTOS DURANTE LA CURACIÓN DE BASES TAXONÓMICAS

3.4.1. Base taxonómica Darwinion

En la Figura 5 se muestra el diagrama del proceso diseñado para generar la base de datos Darwinion, la que se usó como elemento de contraste para generar el producto taxonómico. Esto se diseñó en función de la estructura en la que se encontraba la información en el sitio web en cuestión⁵, con lo cual se definieron cuatro etapas posibles de las que se ejecutaron solamente tres. El número de especies total extraído en este proceso fue de 20.100 especies, las que fueron obtenidas entre las etapas 1 y 2 del mismo proceso, y un total de 6.555 sinónimos obtenidos de la etapa 3. La única diferencia a señalar, es que la extracción de especies fue total, mientras que la de sinónimos fue arbitraria, siguiendo el número de especies presentes en nuestros listados de herbarios, lo que disminuyó el número posible de nombres de sinónimos a extraer, pero este número puede ser extendido a la totalidad si se desea.

⁵ <http://www.darwin.edu.ar/>

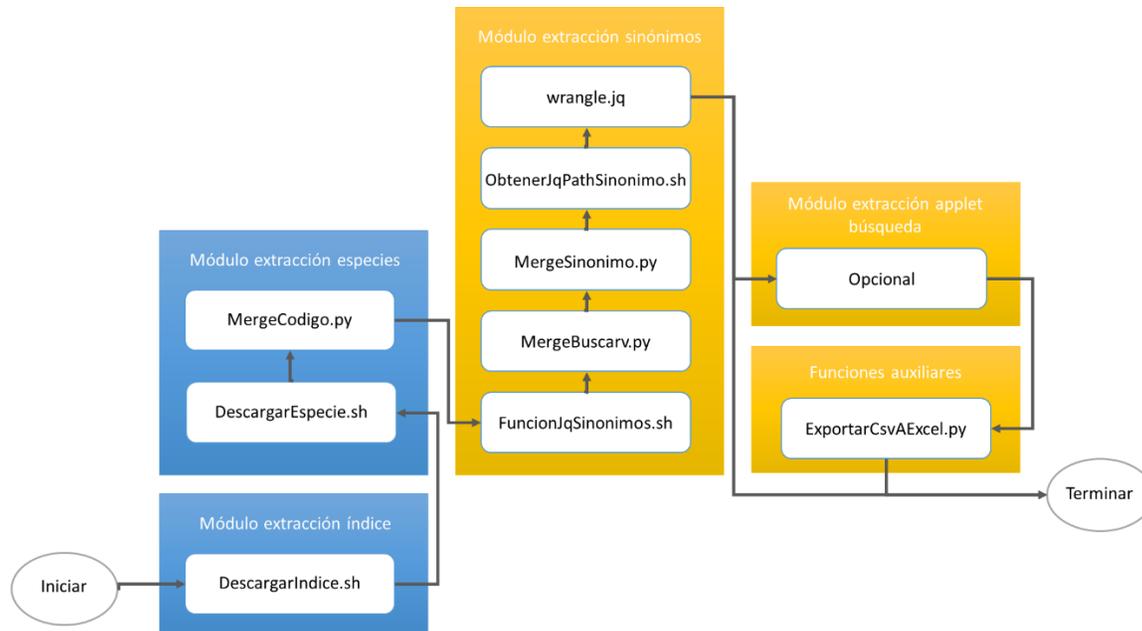


Figura 5. Diagrama resumido de módulo de extracción Darwinion.

Cabe señalar que Darwinion publicó sus datos en formato de ocurrencias al servidor GBIF y que esa base puede ser consultada únicamente a través de la herramienta “API occurrence”. Es decir, no se puede actualmente usar la herramienta “API species” para consultar nombres Darwinion en el GBIF que es algo que se hubiera necesitado durante la curación de los listados nacionales.

3.4.2. Base taxonómica de Rodríguez *et al.* (2018)

Un segundo sub producto es la extracción del catálogo Rodríguez *et al.* (2018) que cuenta con 5.471 especies incluyendo especies de carácter nativo y endémico de Chile. Este catálogo se encontraba originalmente en un archivo pdf de casi 500 páginas con cada especie separada en un párrafo junto a su información parcialmente detallada. El método de extracción fue iterativo, realizando extracciones sucesivas veces, para crear un tabulado final que se corresponda con la información mostrada en el cuadro resumen de dicho catálogo. Se fue agregando una regla de corrección de caso a caso para lograr que la información fuese estructurándose; a medida que se volvía a correr cada extracción se incorporaron correcciones puntuales.

Si bien es cierto, no se pensó en incorporar este sub producto en el proceso de curación de registros taxonómicos, se identificó como de gran valor para posteriores mejoras de herramientas propias de curación de especies de tipo endémico, puesto que los catálogos internacionales muchas veces carecen de esta particularidad, que identifica mayormente a las especies de los listados de herbarios, por lo que se consideró de gran aporte tener esta información de manera estructurada tal y como se publicó originalmente por los autores.

Una vez terminada la etapa de cotejo entre los nombres científicos en las bases de datos de los herbarios y las listas de especies internacionales (GBIF) y regionales (Darwinion), se procedió, únicamente para los registros que cuentan con nombres validados por este cruce de información, a completar los campos relacionados con los niveles superiores de la taxonomía (Reino, Phylum, Clase, Orden, Familia y Género), así como el autor del nombre científico.



04

DISCUSIÓN Y
SEGUIMIENTO

Foto: *Gochnatia foliolosa*, por Marianne Katunarić

4. DISCUSIÓN Y SEGUIMIENTO

El producto que se ha generado responde a un proceso automatizado de búsqueda de inconsistencias en las bases de datos de las colecciones biológicas de los herbarios del Museo Nacional de Historia Natural (MNHN) y de la Universidad de Concepción, de las variables acordadas (fechas, coordenadas y taxonomía). No obstante estos procesos pueden ser aplicados a las mismas variables de colecciones de distintos taxa y del reino animal (siempre y cuando se cuente con listados con nombres científicos validados para el grupo taxonómico, y se tenga acceso a dicho listado). Es importante resaltar el ámbito al cual se acota el resultado de la presente consultoría, el cual es sistematizar y automatizar la búsqueda de inconsistencias dentro de una misma base de datos en relación a fechas y coordenadas y declarar estas inconsistencias. El conjunto de procedimientos busca optimizar y simplificar la tarea del experto en taxonomía, aplicando en forma automatizada tareas repetitivas, con el objetivo de “pre-limpiar” las bases de datos y hacer visibles sus inconsistencias. Es de suma importancia comprender que esta herramienta de automatización no hace la curación final de las colecciones, la cual es responsabilidad del experto en taxonomía.

Se realizaron labores en las cuales el primer enfoque fue crear pasos de análisis con la creación de reportes que entregaban información intermedia en forma de análisis progresivo. Posteriormente se procedió a unificar todo en un proceso único, procurando separar las lógicas del código para mejorar su comprensión y poder ser ejecutado como un todo. Tanto el material del primer diseño como del segundo podrán ser utilizados para crear un aplicativo final, con la creación de reportes preliminares que indiquen al usuario la condición inicial de su información, como también poder seleccionar tanto tareas individuales como tareas genéricas para ser ejecutadas en conjunto.

Se buscó poner a disposición del analista taxonómico información de nombres científicos, puesto que esta información es actualizada regularmente, lo que impele al usuario a realizar un análisis de estos resultados. La herramienta nuevamente, realiza una comparación de los nombres científicos de las colecciones con los catálogos de especies validados internacional o regionalmente, y declara inconsistencias que pudieran surgir de estas comparaciones. De este modo el procedimiento separa dos objetivos diferentes, puesto que los servidores de información taxonómica se enfocan en poner a disposición de los analistas la información, en cuyo caso el estándar Darwin Core (DWC) es un requisito mínimo a cumplir, pero no es su propósito que la información posea una calidad superior que otra información existente en los mismos servicios web, ni que la

misma no contenga información que es detallada y específicamente exacta. Con este propósito es que se diseñaron estos procesos en forma complementaria a la posibilidad de poder usar estos servicios de consulta y acoplarlos dentro del mismo proceso de validación de registros de taxonomía.

05

APÉNDICES



Foto: *Loasa acerifolia*, por Marianne Katunarić

5. APÉNDICES

APÉNDICE 1

BashVariablesFecha.sh

```
#### No borrar con rm -rf
declare -gx
str00zip0001='../basesdb/xlsx2csv/fechas/Reporte_Fechas_Corregidas
.7z'
declare -gx str01xls0001='../basesdb/01_BBDD_RM_MNHN.xlsx'
declare -gx str02xls0002='../basesdb/02_BBDD_V_MNHN.xlsx'
declare -gx str03xls0003='../basesdb/03_BBDD_RM_V_CONCE.xlsx'
declare -gx
str01inp0001='../basesdb/xlsx2csv/01_BBDD_RM_MNHN_xlsx2csv.csv'
declare -gx
str02inp0002='../basesdb/xlsx2csv/02_BBDD_V_MNHN_xlsx2csv.csv'
declare -gx
str03inp0003='../basesdb/xlsx2csv/03_BBDD_RM_V_CONCE_xlsx2csv.csv'

#### No borrar
declare -gx
str01rep0001='../basesdb/xlsx2csv/fechas/01_BBDD_RM_MNHN_fechas_re
porte.csv'
declare -gx
str01exo0001='../basesdb/xlsx2csv/fechas/01_BBDD_RM_MNHN_fechas_re
porte.xlsx'
declare -gx str01dfc0001='dicmod/dicc_01_BBDD_RM_MNHN_fecha.py'
#### Si borrar
declare -gx
str01vin0001='../basesdb/xlsx2csv/fechas/01_BBDD_RM_MNHN_xlsx2csv_
verbatimeventdate.csv'
declare -gx
str01vou0001='../basesdb/xlsx2csv/fechas/01_BBDD_RM_MNHN_xlsx2csv_
eventdate.csv'
```

```
declare -gx
str01vin0002='../basesdb/xlsx2csv/fechas/01_BBDD_RM_MNHN_xlsx2csv_
verbatimyear.csv'
declare -gx
str01vou0002='../basesdb/xlsx2csv/fechas/01_BBDD_RM_MNHN_xlsx2csv_
year.csv'
declare -gx
str01vin0003='../basesdb/xlsx2csv/fechas/01_BBDD_RM_MNHN_xlsx2csv_
verbatimmonth.csv'
declare -gx
str01vou0003='../basesdb/xlsx2csv/fechas/01_BBDD_RM_MNHN_xlsx2csv_
month.csv'
declare -gx
str01vin0004='../basesdb/xlsx2csv/fechas/01_BBDD_RM_MNHN_xlsx2csv_
verbatimday.csv'
declare -gx
str01vou0004='../basesdb/xlsx2csv/fechas/01_BBDD_RM_MNHN_xlsx2csv_
day.csv'
declare -gx
str01vin0005='../basesdb/xlsx2csv/fechas/01_BBDD_RM_MNHN_xlsx2csv_
verbatimdateidentified.csv'
declare -gx
str01vou0005='../basesdb/xlsx2csv/fechas/01_BBDD_RM_MNHN_xlsx2csv_
dateidentified.csv'
declare -gx
str01rcu0001='../basesdb/xlsx2csv/fechas/01_BBDD_RM_MNHN_fechas_in
termedias.csv'
declare -gx
str01rcu0002='../basesdb/xlsx2csv/fechas/01_BBDD_RM_MNHN_fechas_cu
adradas.csv'
declare -gx
str01raw0001='../basesdb/xlsx2csv/fechas/01_BBDD_RM_MNHN_fechas_or
iginales.csv'
declare -gx
str01lexi0001='../basesdb/xlsx2csv/fechas/01_BBDD_RM_MNHN_fechas_co
rregidas.csv'
```

```

#### No borrar
declare -gx
str02rep0001='../basesdb/xlsx2csv/fechas/02_BBDD_V_MNHN_fechas_rep
orte.csv'
declare -gx
str02exo0001='../basesdb/xlsx2csv/fechas/02_BBDD_V_MNHN_fechas_rep
orte.xlsx'
declare -gx str02dfc0001='dicmod/dicc_02_BBDD_V_MNHN_fecha.py'
#### Si borrar
declare -gx
str02vin0001='../basesdb/xlsx2csv/fechas/02_BBDD_V_MNHN_xlsx2csv_v
erbatimeventdate.csv'
declare -gx
str02vou0001='../basesdb/xlsx2csv/fechas/02_BBDD_V_MNHN_xlsx2csv_e
ventdate.csv'
declare -gx
str02vin0002='../basesdb/xlsx2csv/fechas/02_BBDD_V_MNHN_xlsx2csv_v
erbatimyear.csv'
declare -gx
str02vou0002='../basesdb/xlsx2csv/fechas/02_BBDD_V_MNHN_xlsx2csv_y
ear.csv'
declare -gx
str02vin0003='../basesdb/xlsx2csv/fechas/02_BBDD_V_MNHN_xlsx2csv_v
erbatimmonth.csv'
declare -gx
str02vou0003='../basesdb/xlsx2csv/fechas/02_BBDD_V_MNHN_xlsx2csv_m
onth.csv'
declare -gx
str02vin0004='../basesdb/xlsx2csv/fechas/02_BBDD_V_MNHN_xlsx2csv_v
erbatimday.csv'
declare -gx
str02vou0004='../basesdb/xlsx2csv/fechas/02_BBDD_V_MNHN_xlsx2csv_d
ay.csv'
declare -gx
str02vin0005='../basesdb/xlsx2csv/fechas/02_BBDD_V_MNHN_xlsx2csv_v
erbatimdateidentified.csv'

```

```
declare -gx
str02vou0005='../basesdb/xlsx2csv/fechas/02_BBDD_V_MNHN_xlsx2csv_d
ateidentified.csv'
declare -gx
str02rcu0001='../basesdb/xlsx2csv/fechas/02_BBDD_V_MNHN_fechas_int
ermedias.csv'
declare -gx
str02rcu0002='../basesdb/xlsx2csv/fechas/02_BBDD_V_MNHN_fechas_cua
dradas.csv'
declare -gx
str02raw0001='../basesdb/xlsx2csv/fechas/02_BBDD_V_MNHN_fechas_ori
ginales.csv'
declare -gx
str02exi0001='../basesdb/xlsx2csv/fechas/02_BBDD_V_MNHN_fechas_cor
regidas.csv'

#### No borrar
declare -gx
str03rep0001='../basesdb/xlsx2csv/fechas/03_BBDD_RM_V_CONCE_fechas
_reporte.csv'
declare -gx
str03exo0001='../basesdb/xlsx2csv/fechas/03_BBDD_RM_V_CONCE_fechas
_reporte.xlsx'
declare -gx
str03dfc0001='dicmod/dicc_03_BBDD_RM_V_CONCE_fecha.py'
#### Si borrar
declare -gx
str03vin0001='../basesdb/xlsx2csv/fechas/03_BBDD_RM_V_CONCE_xlsx2c
sv_verbatimeventdate.csv'
declare -gx
str03vou0001='../basesdb/xlsx2csv/fechas/03_BBDD_RM_V_CONCE_xlsx2c
sv_eventdate.csv'
declare -gx
str03vin0002='../basesdb/xlsx2csv/fechas/03_BBDD_RM_V_CONCE_xlsx2c
sv_verbatimyear.csv'
```

```

declare -gx
str03vou0002='../basesdb/xlsx2csv/fechas/03_BBDD_RM_V_CONCE_xlsx2c
sv_year.csv'
declare -gx
str03vin0003='../basesdb/xlsx2csv/fechas/03_BBDD_RM_V_CONCE_xlsx2c
sv_verbatimmonth.csv'
declare -gx
str03vou0003='../basesdb/xlsx2csv/fechas/03_BBDD_RM_V_CONCE_xlsx2c
sv_month.csv'
declare -gx
str03rcu0001='../basesdb/xlsx2csv/fechas/03_BBDD_RM_V_CONCE_fechas
_intermedias.csv'
declare -gx
str03exi0001='../basesdb/xlsx2csv/fechas/03_BBDD_RM_V_CONCE_fechas
_corregidas.csv'

```

BashVariablesCoordenadas.sh

```

#### No borrar con rm -rf
declare -gx
str00zip0001='../basesdb/xlsx2csv/coordenadas/Reporte_coordenadas_
Corregidas.7z'
declare -gx str01xls0001='../basesdb/01_BBDD_RM_MNHN.xlsx'
declare -gx str02xls0002='../basesdb/02_BBDD_V_MNHN.xlsx'
declare -gx str03xls0003='../basesdb/03_BBDD_RM_V_CONCE.xlsx'
declare -gx
str01inp0001='../basesdb/xlsx2csv/01_BBDD_RM_MNHN_xlsx2csv.csv'
declare -gx
str02inp0002='../basesdb/xlsx2csv/02_BBDD_V_MNHN_xlsx2csv.csv'
declare -gx
str03inp0003='../basesdb/xlsx2csv/03_BBDD_RM_V_CONCE_xlsx2csv.csv'

#### No borrar

```

```
declare -gx
str01rep0001='../basesdb/xlsx2csv/coordenadas/01_BBDD_RM_MNHN_coordenadas_reporte.csv'
declare -gx
str01exo0001='../basesdb/xlsx2csv/coordenadas/01_BBDD_RM_MNHN_coordenadas_reporte.xlsx'
declare -gx
str01dfc0001='dicmod/dicc_01_BBDD_RM_MNHN_coordenada.py'
#### Si borrar
declare -gx
str01llan0001='../basesdb/xlsx2csv/coordenadas/01_BBDD_RM_MNHN_xlsx2csv_verbatimdecimallatitude.csv'
declare -gx
str01llau0001='../basesdb/xlsx2csv/coordenadas/01_BBDD_RM_MNHN_xlsx2csv_decimallatitude.csv'
declare -gx
str01llon0001='../basesdb/xlsx2csv/coordenadas/01_BBDD_RM_MNHN_xlsx2csv_verbatimdecimallongitude.csv'
declare -gx
str01llou0001='../basesdb/xlsx2csv/coordenadas/01_BBDD_RM_MNHN_xlsx2csv_decimallongitude.csv'
declare -gx
str01llan0002='../basesdb/xlsx2csv/coordenadas/01_BBDD_RM_MNHN_xlsx2csv_verbatimlat_gr.csv'
declare -gx
str01llau0002='../basesdb/xlsx2csv/coordenadas/01_BBDD_RM_MNHN_xlsx2csv_lat_gr.csv'
declare -gx
str01llan0003='../basesdb/xlsx2csv/coordenadas/01_BBDD_RM_MNHN_xlsx2csv_verbatimlat_min.csv'
declare -gx
str01llau0003='../basesdb/xlsx2csv/coordenadas/01_BBDD_RM_MNHN_xlsx2csv_lat_min.csv'
declare -gx
str01llan0004='../basesdb/xlsx2csv/coordenadas/01_BBDD_RM_MNHN_xlsx2csv_verbatimlat_seg.csv'
```

```
declare -gx
str01lau0004='../basesdb/xlsx2csv/coordenadas/01_BBDD_RM_MNHN_xlsx
2csv_lat_seg.csv'
declare -gx
str01lon0002='../basesdb/xlsx2csv/coordenadas/01_BBDD_RM_MNHN_xlsx
2csv_verbatimlong_gr.csv'
declare -gx
str01lou0002='../basesdb/xlsx2csv/coordenadas/01_BBDD_RM_MNHN_xlsx
2csv_long_gr.csv'
declare -gx
str01lon0003='../basesdb/xlsx2csv/coordenadas/01_BBDD_RM_MNHN_xlsx
2csv_verbatimlong_min.csv'
declare -gx
str01lou0003='../basesdb/xlsx2csv/coordenadas/01_BBDD_RM_MNHN_xlsx
2csv_long_min.csv'
declare -gx
str01lon0004='../basesdb/xlsx2csv/coordenadas/01_BBDD_RM_MNHN_xlsx
2csv_verbatimlong_seg.csv'
declare -gx
str01lou0004='../basesdb/xlsx2csv/coordenadas/01_BBDD_RM_MNHN_xlsx
2csv_long_seg.csv'
declare -gx
str01seo0001='../basesdb/xlsx2csv/coordenadas/01_BBDD_RM_MNHN_coor
denadas_originales_sextupla.csv'
declare -gx
str01sec0001='../basesdb/xlsx2csv/coordenadas/01_BBDD_RM_MNHN_coor
denadas_corregidas_sextupla.csv'
declare -gx
str01duo0001='../basesdb/xlsx2csv/coordenadas/01_BBDD_RM_MNHN_coor
denadas_originales_dupla.csv'
declare -gx
str01duc0001='../basesdb/xlsx2csv/coordenadas/01_BBDD_RM_MNHN_coor
denadas_corregidas_dupla.csv'
declare -gx
str01cud0001='../basesdb/xlsx2csv/coordenadas/01_BBDD_RM_MNHN_coor
denadas_corregidas_dupla_cuadradas.csv'
```

```
declare -gx
str01cas0001='../basesdb/xlsx2csv/coordenadas/01_BBDD_RM_MNHN_coordenadas_originales_concatenadas_sextupla.csv'
declare -gx
str01cad0001='../basesdb/xlsx2csv/coordenadas/01_BBDD_RM_MNHN_coordenadas_originales_concatenadas_dupla.csv'
declare -gx
str01goo0001='../basesdb/xlsx2csv/coordenadas/01_BBDD_RM_MNHN_coordenadas_originales_concatenadas.csv'
declare -gx
str01goc0001='../basesdb/xlsx2csv/coordenadas/01_BBDD_RM_MNHN_coordenadas_corregidas_concatenadas.csv'
declare -gx
str01utm0001='../basesdb/xlsx2csv/coordenadas/01_BBDD_RM_MNHN_coordenadas_corregidas_dupla_utm.csv'
declare -gx
str01ver0001='../basesdb/xlsx2csv/coordenadas/01_BBDD_RM_MNHN_coordenadas_verificacion.csv'
declare -gx
str01t010001='../basesdb/xlsx2csv/coordenadas/01_BBDD_RM_MNHN_coordenadas_concatenadas_reporte01.csv'
declare -gx
str01t020001='../basesdb/xlsx2csv/coordenadas/01_BBDD_RM_MNHN_coordenadas_concatenadas_reporte02.csv'
declare -gx
str01t030001='../basesdb/xlsx2csv/coordenadas/01_BBDD_RM_MNHN_coordenadas_concatenadas_reporte03.csv'
declare -gx
str01t040001='../basesdb/xlsx2csv/coordenadas/01_BBDD_RM_MNHN_coordenadas_concatenadas_reporte04.csv'
declare -gx
str01cli0001='../basesdb/xlsx2csv/coordenadas/01_BBDD_RM_MNHN_insumo_interseccion.csv'
declare -gx
str01clo0001='../basesdb/xlsx2csv/coordenadas/01_BBDD_RM_MNHN_salida_interseccion.csv'
```

```

declare -gx
str01sli0001='../basesdb/xlsx2csv/shapefiles/01_BBDD_RM_MNHN_insum
o_interseccion.shp'
declare -gx
str01slo0001='../basesdb/xlsx2csv/shapefiles/01_BBDD_RM_MNHN_salid
a_interseccion.shp'
declare -gx
str01klo0001='../basesdb/xlsx2csv/coordenadas/01_BBDD_RM_MNHN_merg
e_interseccion.csv'

#### No borrar
declare -gx
str02rep0001='../basesdb/xlsx2csv/coordenadas/02_BBDD_V_MNHN_coord
enadas_reporte.csv'
declare -gx
str02exo0001='../basesdb/xlsx2csv/coordenadas/02_BBDD_V_MNHN_coord
enadas_reporte.xlsx'
declare -gx
str02dfc0001='dicmod/dicc_02_BBDD_V_MNHN_coordenada.py'
#### Si borrar
declare -gx
str02lan0001='../basesdb/xlsx2csv/coordenadas/02_BBDD_V_MNHN_xlsx2
csv_verbatimdecimallatitude.csv'
declare -gx
str02lau0001='../basesdb/xlsx2csv/coordenadas/02_BBDD_V_MNHN_xlsx2
csv_decimallatitude.csv'
declare -gx
str02lon0001='../basesdb/xlsx2csv/coordenadas/02_BBDD_V_MNHN_xlsx2
csv_verbatimdecimallongitude.csv'
declare -gx
str02lou0001='../basesdb/xlsx2csv/coordenadas/02_BBDD_V_MNHN_xlsx2
csv_decimallongitude.csv'
declare -gx
str02lan0002='../basesdb/xlsx2csv/coordenadas/02_BBDD_V_MNHN_xlsx2
csv_verbatimlat_gr.csv'

```

```
declare -gx
str021au0002='../basesdb/xlsx2csv/coordenadas/02_BBDD_V_MNHN_xlsx2
csv_lat_gr.csv'
declare -gx
str021an0003='../basesdb/xlsx2csv/coordenadas/02_BBDD_V_MNHN_xlsx2
csv_verbatimlat_min.csv'
declare -gx
str021au0003='../basesdb/xlsx2csv/coordenadas/02_BBDD_V_MNHN_xlsx2
csv_lat_min.csv'
declare -gx
str021an0004='../basesdb/xlsx2csv/coordenadas/02_BBDD_V_MNHN_xlsx2
csv_verbatimlat_seg.csv'
declare -gx
str021au0004='../basesdb/xlsx2csv/coordenadas/02_BBDD_V_MNHN_xlsx2
csv_lat_seg.csv'
declare -gx
str021on0002='../basesdb/xlsx2csv/coordenadas/02_BBDD_V_MNHN_xlsx2
csv_verbatimlong_gr.csv'
declare -gx
str021ou0002='../basesdb/xlsx2csv/coordenadas/02_BBDD_V_MNHN_xlsx2
csv_long_gr.csv'
declare -gx
str021on0003='../basesdb/xlsx2csv/coordenadas/02_BBDD_V_MNHN_xlsx2
csv_verbatimlong_min.csv'
declare -gx
str021ou0003='../basesdb/xlsx2csv/coordenadas/02_BBDD_V_MNHN_xlsx2
csv_long_min.csv'
declare -gx
str021on0004='../basesdb/xlsx2csv/coordenadas/02_BBDD_V_MNHN_xlsx2
csv_verbatimlong_seg.csv'
declare -gx
str021ou0004='../basesdb/xlsx2csv/coordenadas/02_BBDD_V_MNHN_xlsx2
csv_long_seg.csv'
declare -gx
str02seo0001='../basesdb/xlsx2csv/coordenadas/02_BBDD_V_MNHN_coord
enadas_originales_sextupla.csv'
```

```
declare -gx
str02sec0001='../basesdb/xlsx2csv/coordenadas/02_BBDD_V_MNHN_coord
enadas_corregidas_sextupla.csv'
declare -gx
str02duo0001='../basesdb/xlsx2csv/coordenadas/02_BBDD_V_MNHN_coord
enadas_originales_dupla.csv'
declare -gx
str02duc0001='../basesdb/xlsx2csv/coordenadas/02_BBDD_V_MNHN_coord
enadas_corregidas_dupla.csv'
declare -gx
str02cud0001='../basesdb/xlsx2csv/coordenadas/02_BBDD_V_MNHN_coord
enadas_corregidas_dupla_cuadradas.csv'
declare -gx
str02cas0001='../basesdb/xlsx2csv/coordenadas/02_BBDD_V_MNHN_coord
enadas_originales_concatenadas_sextupla.csv'
declare -gx
str02cad0001='../basesdb/xlsx2csv/coordenadas/02_BBDD_V_MNHN_coord
enadas_originales_concatenadas_dupla.csv'
declare -gx
str02goo0001='../basesdb/xlsx2csv/coordenadas/02_BBDD_V_MNHN_coord
enadas_originales_concatenadas.csv'
declare -gx
str02goc0001='../basesdb/xlsx2csv/coordenadas/02_BBDD_V_MNHN_coord
enadas_corregidas_concatenadas.csv'
declare -gx
str02utm0001='../basesdb/xlsx2csv/coordenadas/02_BBDD_V_MNHN_coord
enadas_corregidas_dupla_utm.csv'
declare -gx
str02ver0001='../basesdb/xlsx2csv/coordenadas/02_BBDD_V_MNHN_coord
enadas_verificacion.csv'
declare -gx
str02t010001='../basesdb/xlsx2csv/coordenadas/02_BBDD_V_MNHN_coord
enadas_concatenadas_reporte01.csv'
declare -gx
str02t020001='../basesdb/xlsx2csv/coordenadas/02_BBDD_V_MNHN_coord
enadas_concatenadas_reporte02.csv'
```

```
declare -gx
str02t030001='../basesdb/xlsx2csv/coordenadas/02_BBDD_V_MNHN_coordenadas_concatenadas_reporte03.csv'
declare -gx
str02t040001='../basesdb/xlsx2csv/coordenadas/02_BBDD_V_MNHN_coordenadas_concatenadas_reporte04.csv'
declare -gx
str02cli0001='../basesdb/xlsx2csv/coordenadas/02_BBDD_V_MNHN_insumo_interseccion.csv'
declare -gx
str02clo0001='../basesdb/xlsx2csv/coordenadas/02_BBDD_V_MNHN_salida_interseccion.csv'
declare -gx
str02sli0001='../basesdb/xlsx2csv/shapefiles/02_BBDD_V_MNHN_insumo_interseccion.shp'
declare -gx
str02slo0001='../basesdb/xlsx2csv/shapefiles/02_BBDD_V_MNHN_salida_interseccion.shp'
declare -gx
str02klo0001='../basesdb/xlsx2csv/coordenadas/02_BBDD_V_MNHN_merge_interseccion.csv'

#### No borrar
declare -gx
str03rep0001='../basesdb/xlsx2csv/coordenadas/03_BBDD_RM_V_CONCE_coordenadas_reporte.csv'
declare -gx
str03exo0001='../basesdb/xlsx2csv/coordenadas/03_BBDD_RM_V_CONCE_coordenadas_reporte.xlsx'
declare -gx
str03dfc0001='dicmod/dicc_03_BBDD_RM_V_CONCE_coordenada.py'
#### Si borrar
declare -gx
str03lan0001='../basesdb/xlsx2csv/coordenadas/03_BBDD_RM_V_CONCE_xlsx2csv_verbatimdecimallatitude.csv'
```

```
declare -gx
str03lau0001='../basesdb/xlsx2csv/coordenadas/03_BBDD_RM_V_CONCE_x
lsx2csv_decimallatitude.csv'
declare -gx
str03lon0001='../basesdb/xlsx2csv/coordenadas/03_BBDD_RM_V_CONCE_x
lsx2csv_verbatimdecimallongitude.csv'
declare -gx
str03lou0001='../basesdb/xlsx2csv/coordenadas/03_BBDD_RM_V_CONCE_x
lsx2csv_decimallongitude.csv'
declare -gx
str03seo0001='../basesdb/xlsx2csv/coordenadas/03_BBDD_RM_V_CONCE_c
oordenadas_intermedias.csv'
declare -gx
str03exi0001='../basesdb/xlsx2csv/coordenadas/03_BBDD_RM_V_CONCE_c
oordenadas_corregidas.csv'
declare -gx
str03duo0001='../basesdb/xlsx2csv/coordenadas/03_BBDD_RM_V_CONCE_c
oordenadas_originales_dupla.csv'
declare -gx
str03duc0001='../basesdb/xlsx2csv/coordenadas/03_BBDD_RM_V_CONCE_c
oordenadas_corregidas_dupla.csv'
declare -gx
str03utm0001='../basesdb/xlsx2csv/coordenadas/03_BBDD_RM_V_CONCE_c
oordenadas_corregidas_dupla_utm.csv'
declare -gx
str03cad0001='../basesdb/xlsx2csv/coordenadas/03_BBDD_RM_V_CONCE_c
oordenadas_originales_concatenadas_dupla.csv'
declare -gx
str03cas0001='../basesdb/xlsx2csv/coordenadas/03_BBDD_RM_V_CONCE_c
oordenadas_originales_concatenadas_sextupla.csv'
declare -gx
str03goo0001='../basesdb/xlsx2csv/coordenadas/03_BBDD_RM_V_CONCE_c
oordenadas_originales_concatenadas.csv'
declare -gx
str03t010001='../basesdb/xlsx2csv/coordenadas/03_BBDD_RM_V_CONCE_c
oordenadas_concatenadas_reporte01.csv'
```

```
declare -gx
str03t020001='../basesdb/xlsx2csv/coordenadas/03_BBDD_RM_V_CONCE_c
oordenadas_concatenadas_reporte02.csv'
declare -gx
str03t030001='../basesdb/xlsx2csv/coordenadas/03_BBDD_RM_V_CONCE_c
oordenadas_concatenadas_reporte03.csv'
declare -gx
str03cli0001='../basesdb/xlsx2csv/coordenadas/03_BBDD_RM_V_CONCE_i
nsumo_interseccion.csv'
declare -gx
str03clo0001='../basesdb/xlsx2csv/coordenadas/03_BBDD_RM_V_CONCE_s
alida_interseccion.csv'
declare -gx
str03sli0001='../basesdb/xlsx2csv/shapefiles/03_BBDD_RM_V_CONCE_in
sumo_interseccion.shp'
declare -gx
str03slo0001='../basesdb/xlsx2csv/shapefiles/03_BBDD_RM_V_CONCE_sa
lida_interseccion.shp'
declare -gx
str03klo0001='../basesdb/xlsx2csv/coordenadas/03_BBDD_RM_V_CONCE_m
erge_interseccion.csv'
```

BashVariablesTaxonomia.sh

```
declare -gx
strurl01='http://api.gbif.org/v1/species/match?name='
declare -gx
strxls01='../basesdb/xlsx2csv/01_BBDD_RM_MNHN_xlsx2csv.csv'
declare -gx
strxls02='../basesdb/xlsx2csv/02_BBDD_V_MNHN_xlsx2csv.csv'
declare -gx
strxls03='../basesdb/xlsx2csv/03_BBDD_RM_V_CONCE_xlsx2csv.csv'
declare -gx strsh001='src/01_BBDD_RM_MNHN_xlsx2csv.sh'
declare -gx strsh002='src/02_BBDD_V_MNHN_xlsx2csv.sh'
declare -gx strsh003='src/03_BBDD_RM_V_CONCE_xlsx2csv.sh'
```

```

declare -gx
strtx001='src/01_BBDD_RM_MNHN_xlsx2csv_preparar_taxize.sh'
declare -gx
strtx002='src/02_BBDD_V_MNHN_xlsx2csv_preparar_taxize.sh'
declare -gx
strtx003='src/03_BBDD_RM_V_CONCE_xlsx2csv_preparar_taxize.sh'
declare -gx
strtz001='src/01_BBDD_RM_MNHN_xlsx2csv_consultar_taxize.sh'
declare -gx
strtz002='src/02_BBDD_V_MNHN_xlsx2csv_consultar_taxize.sh'
declare -gx
strtz003='src/03_BBDD_RM_V_CONCE_xlsx2csv_consultar_taxize.sh'
declare -gx strcsv01='csv/01_BBDD_RM_MNHN_xlsx2csv.csv'
declare -gx strcsv02='csv/02_BBDD_V_MNHN_xlsx2csv.csv'
declare -gx strcsv03='csv/03_BBDD_RM_V_CONCE_xlsx2csv.csv'
declare -gx
strcvv01='../basesdb/xlsx2csv/taxonomia/01_BBDD_RM_MNHN_xlsx2csv_t
axonomia_gbif.csv'
declare -gx
strcvv02='../basesdb/xlsx2csv/taxonomia/02_BBDD_V_MNHN_xlsx2csv_ta
xonomia_gbif.csv'
declare -gx
strcvv03='../basesdb/xlsx2csv/taxonomia/03_BBDD_RM_V_CONCE_xlsx2cs
v_taxonomia_gbif.csv'
declare -gx
strxvv01='../basesdb/xlsx2csv/taxonomia/01_BBDD_RM_MNHN_xlsx2csv_t
axonomia_gbif.xlsx'
declare -gx
strxvv02='../basesdb/xlsx2csv/taxonomia/02_BBDD_V_MNHN_xlsx2csv_ta
xonomia_gbif.xlsx'
declare -gx
strxvv03='../basesdb/xlsx2csv/taxonomia/03_BBDD_RM_V_CONCE_xlsx2cs
v_taxonomia_gbif.xlsx'

```

APÉNDICE 2

FuncionesFecha.sh

```
## buscar patrones de fechas
function ListarPatronesFecha ()
{
    ListarCabecerasArroba ${1} ${2} \
        | gawk -F':' 'BEGIN{IGNORECASE=1} {if($2~/day/ || $2~/month/
|| $2~/year/ || $2~/dÃa/ || $2~/mes/ || $2~/aÃ±o/ || $2~/date/ ||
$2~/fecha/) print $0}' OFS=':' \
        | sed -r 's/(.*):(.*):(.*):(.*):(.*):/gawk -F'\''@'\''
'\''NR>\5 {print FILENAME, "\3", "\2", "\$3}'\'' OFS='\''':'\'' \1
| sort | uniq -c/' \
        | bash \
        | sed -r 's/^ +//g;s/ /:/' \
        | gawk -F':' '{if($5=="") {print $1,$2,$3,$4,"0"} else
{gsub(/[/0-9]/,"0",$5) ; print $1,$2,$3,$4,$5}}' OFS=':' \
        | sort | uniq -c \
        | sed -r 's/^ +//g;s/ /:/' \
        | gawk -F':' '{print $3,$4,$5,$6,$2,$1}' OFS=':' \
        | sort -t':' -k1,4 -n \
        | gawk -F':' '{print $1,$2,$3,$4,$5*$6}' OFS=':' \
        | gawk -F ':' '{a[$1":"$2":"$3":"$4] += $5} END{for (i in a)
print i, a[i]}' OFS=':' \
        | sort -t':' -k1,4 -n
    }
export -f ListarPatronesFecha

function PruebaEliminarFechaCeroFijo ()
{
    sed -r -i 's/^(([0-9]{4})_([0-9]{2})_([0]{2}))$/\1_\2/' ${1}
    sed -r -i 's/^(([0-9]{4})_([0]{2}))$/\1/' ${1}
    sed -r -i 's/^0000$//' ${1}
    sed -r -i 's/^00$//' ${1}
    sed -r -i 's/_$//' ${1}
}
```

```

    sed -r -i 's/^\_([0-9]{2})$//' ${1}
}
export -f PruebaEliminarFechaCeroFijo

function FormatearYearArroba ()
{
    gawk -i inplace -F'@' 'NR==1 {print $'"${2}"'} NR>1
{if($'"${2}"'!="") {printf "%04d\n", $'"${2}"'} else {print ""}}'
OFS='@' ${1}
}
export -f FormatearYearArroba

function FormatearMonthArroba ()
{
    gawk -i inplace -F'@' 'NR==1 {print $'"${2}"'} NR>1
{if($'"${2}"'!="") {printf "%02d\n", $'"${2}"'} else {print ""}}'
OFS='@' ${1}
}
export -f FormatearMonthArroba

function FormatearDayArroba ()
{
    gawk -i inplace -F'@' 'NR==1 {print $'"${2}"'} NR>1
{if($'"${2}"'!="") {printf "%02d\n", $'"${2}"'} else {print ""}}'
OFS='@' ${1}
}
export -f FormatearDayArroba

function RscriptCuadrarDataframe ()
{
    R --slave --args ${1} ${2} ${3} ${4} ${5} ${6} ${7} <
FuncionCuadratura.R
}
export -f RscriptCuadrarDataframe

function PyscriptCuadrarDataframe ()
{

```

```

    source activate py36
    python FuncionCuadratura.py ${1} ${2} ${3} ${4} ${5} ${6} ${7}
    source deactivate
}
export -f PyscriptCuadrarDataframe

function CuadrarFechaCompuestaASimple ()
{
    gawk -i inplace -F'[_@]' 'NR==1 {print $0}
                                NR>1 {if($4=="" && $1!="") {printf
"%s_%s_%s@%s@%s@%s\n", $1,$2,$3,$1,$2,$3}
                                else {printf
"%s_%s_%s@%s@%s@%s\n", $1,$2,$3,$4,$5,$6}}' OFS='@' "${1}"
    #gawk -i inplace -F'[_@]' 'NR==1 {print $0}
    #                                NR>1 {if($5=="" && $2!="") {printf
"%s_%s_%s@%s@%s@%s\n", $1,$2,$3,$4,$2,$6}
    #                                else {printf
"%s_%s_%s@%s@%s@%s\n", $1,$2,$3,$4,$5,$6}}' OFS='@' "${1}"
    #gawk -i inplace -F'[_@]' 'NR==1 {print $0}
    #                                NR>1 {if($6=="" && $3!="") {printf
"%s_%s_%s@%s@%s@%s\n", $1,$2,$3,$4,$5,$3}
    #                                else {printf
"%s_%s_%s@%s@%s@%s\n", $1,$2,$3,$4,$5,$6}}' OFS='@' "${1}"
}
export -f CuadrarFechaCompuestaASimple

function CuadrarFechaSimpleACompuesta ()
{
    gawk -i inplace -F'[_@]' 'NR==1 {print $0}
                                NR>1 {if($4!="")
                                    {if($1=="") {printf
"%04d_%02d_%02d@%s@%s@%s\n", $4,$5,$6,$4,$5,$6}
                                    else {printf
"%04d_%02d_%02d@%s@%s@%s\n", $1,$2,$3,$4,$5,$6}}
                                else {printf
"%s_%s_%s@%s@%s@%s\n", $1,$2,$3,$4,$5,$6}}' OFS='@' "${1}"
}
}

```

```

export -f CuadrarFechaSimpleACompuesta

function ImprimirReportePatronesFecha ()
{
    ListarPatronesFecha ${1} ${2} \
        | gawk -F':' '{sub(/\..csv/, ".xlsx", $1) ; sub(/\..\.\/[a-
z]+\\/, "", $1) ; print $1, $3, $2, $4, $5}' OFS=':'
}
export -f ImprimirReportePatronesFecha

function ListarRangoFecha ()
{
    ListarCabecerasArroba ${1} ${2} \
        | gawk -F':' 'BEGIN{IGNORECASE=1} {if($2~/day/ || $2~/month/
|| $2~/year/ || $2~/dÃa/ || $2~/mes/ || $2~/aÃto/ || $2~/date/ ||
$2~/fecha/) print $0}' OFS=':' \
        | sed -r 's/(.*):(.*):(.*):(.*):(.*)/(echo -n
"\1:col:\3:fila:\5:Rango-\2:" ; gawk -F'\''@'\'' '\''NR>\5 {print
\$\3}'\'' \1 | sort -n | uniq | paste -d'\''','\'' -s)/' \
        | bash
}
export -f ListarRangoFecha

function ListarValorFueraRangoFecha ()
{
    ListarCabecerasArroba ${1} ${2} \
        | gawk -F':' 'BEGIN{IGNORECASE=1} {if($2~/day/ || $2~/dÃa/)
print $0}' OFS=':' \
        | sed -r 's/(.*):(.*):(.*):(.*):(.*)/gawk -F'\''@'\''
'\''NR>\5 {if(\$\3<1 || \$\3>31) print FILENAME, "\2", "\3", "fuera-
rango", \$\3}'\'' \1 | sort | uniq -c/' \
        | bash \
        | gawk '{if($6=="") {print $2,$3,$4,$5,"vacios",$1}
else {print $2,$3,$4,$5,$6,$1}}' OFS=':'
    ListarCabecerasArroba ${1} ${2} \
        | gawk -F':' 'BEGIN{IGNORECASE=1} {if($2~/month/ ||
$2~/mes/) print $0}' OFS=':' \

```

```

    | sed -r 's/(.*):(.*):(.*):(.*):(.*)/gawk -F\'\'@\'\'
\'\'NR>\5 {if(\$\3<1 || \$\3>12) print FILENAME,"\2","\3","fuera-
rango",\$\3}\'\' \1 | sort | uniq -c/' \
    | bash \
    | gawk '{if($6=="") {print $2,$3,$4,$5,"vacios",$1}
           else {print $2,$3,$4,$5,$6,$1}}' OFS=':'
ListarCabecerasArroba ${1} ${2} \
    | gawk -F':' 'BEGIN{IGNORECASE=1} {if($2~/year/ ||
$2~/aÃ±o/) print $0}' OFS=':' \
    | sed -r 's/(.*):(.*):(.*):(.*):(.*)/gawk -F\'\'@\'\'
\'\'NR>\5 {if(\$\3<1700 || \$\3>2018) print
FILENAME,"\2","\3","fuera-rango",\$\3}\'\' \1 | sort | uniq -c/' \
    | bash \
    | gawk '{if($6=="") {print $2,$3,$4,$5,"vacios",$1}
           else {print $2,$3,$4,$5,$6,$1}}' OFS=':'
}
export -f ListarValorFueraRangoFecha

function ImprimirReporteFueraRangoFecha ()
{
    ListarValorFueraRangoFecha ${1} ${2} \
    | gawk -F':' '{sub(/\.csv/, ".xlsx", $1) ; sub(/\.\.\/[a-
z]+\.\./, "", $1) ; print $1,$2,$3,$4,$5,$6}' OFS=':'
}
export -f ImprimirReporteFueraRangoFecha

function SepararFechas ()
{
    ListarCabecerasArroba ${1} ${2} \
    | gawk -F':' 'BEGIN{IGNORECASE=1} {if($2~/day/ || $2~/month/
|| $2~/year/ || $2~/dÃ±a/ || $2~/mes/ || $2~/aÃ±o/ || $2~/date/ ||
$2~/fecha/) print $0}' OFS=':' \
    | gawk -F':' '{print $1,$3}' OFS=':' \
    | gawk -F':' '{line="";for (i = 2; i <= NF; i++) line = line
$i " "; table[$1]=table[$1] line;} END {for (key in table) print
key " => " table[key];}' \

```

```

        | sed -r 's/(.*) => (.*)/ \2 \1/;:a;s/ ([0-9]{1,2}) / \$\1,
/g;ta;s/(.*)/,/\1}'\'' OFS='\''@'\'' /;s/^ /gawk -F'\''@'\''
'\''{print /' \
        | bash \
        > ${3}
    }
export -f SepararFechas

function SepararFechasSimples ()
{
    ListarCabecerasArroba ${1} ${2} \
        | gawk -F':' 'BEGIN{IGNORECASE=1} {if($2~/day/ || $2~/month/
|| $2~/year/ || $2~/dÃa/ || $2~/mes/ || $2~/aÃto/) print $0}'
OFS=':' \
        | gawk -F':' '{print $1,$3}' OFS=':' \
        | gawk -F':' '{line="";for (i = 2; i <= NF; i++) line = line
$i " "; table[$1]=table[$1] line;} END {for (key in table) print
key " => " table[key];}' \
        | sed -r 's/(.*) => (.*)/ \2 \1/;:a;s/ ([0-9]{1,2}) / \$\1,
/g;ta;s/(.*)/,/\1}'\'' OFS='\''@'\'' /;s/^ /gawk -F'\''@'\''
'\''{print /' \
        | bash \
        > ${3}
    }
export -f SepararFechasSimples

function SepararFechasCompuestas ()
{
    ListarCabecerasArroba ${1} ${2} \
        | gawk -F':' 'BEGIN{IGNORECASE=1} {if($2~/date/ ||
$2~/fecha/) print $0}' OFS=':' \
        | gawk -F':' '{print $1,$3}' OFS=':' \
        | gawk -F':' '{line="";for (i = 2; i <= NF; i++) line = line
$i " "; table[$1]=table[$1] line;} END {for (key in table) print
key " => " table[key];}' \

```

```
| sed -r 's/(.*) => (.) / \2 \1/;:a;s/ ([0-9]{1,2}) / \$\1,
/g;ta;s/(.*) ,/\1}'\'' OFS='\''@'\'' /;s/^ /gawk -F'\''@'\''
'\''{print '/' \
| bash \
> ${3}
}
export -f SepararFechasCompuestas

function PruebaQuitarColumnaArroba ()
{
cat ${1} | cut -d'@' -f${2} --complement > col_tmp.csv
cp col_tmp.csv ${1} ; rm -rf col_tmp.csv
}
export -f PruebaQuitarColumnaArroba

function PruebaReordenarColumnasFechaArroba ()
{
gawk -i inplace -F'@' '{print $3,$2,$1}' OFS='@' ${1}
}
export -f PruebaReordenarColumnasFechaArroba

function PruebaEliminarCasosErroneosArroba ()
{
##### gawk sed cat cut paste
gawk -F'@' '{print "$${2}"}' OFS='@' ${1} > eler01.csv
gawk -F'@' 'NR>1 {if($1=="${3}") print FILENAME, NR,
""${3}""}' OFS='_-' eler01.csv \
| sed -r 's/(.*)-_(.*)-_(.*)/sed -r -i '\''\2s\/\3\/\1'\''
\1/' \
| bash
cat ${1} | cut -d'@' -f"${2}" --complement > eler02.csv
paste -d'@' eler01.csv eler02.csv > ${1}
rm -rf eler01.csv eler02.csv
}
export -f PruebaEliminarCasosErroneosArroba

function PruebaReemplazarCasosErroneosArroba ()
```

```

{
#### gawk sed cat cut paste
gawk -F'@' '{print "$2"}' OFS='@' $1 > reer01.csv
gawk -F'@' 'NR>1 {if($1~/\"$3\"/) print FILENAME, NR,
\"$3\", \"$4\"}' OFS='-_' reer01.csv \
| sed -r 's/(.*)-_(.*)-_(.*)-_(.*)/sed -r -i
'\2s\3\4\1/' \
| bash
cat $1 | cut -d'@' -f"$2" --complement > reer02.csv
paste -d'@' reer01.csv reer02.csv > $1
rm -rf reer01.csv reer02.csv
}
export -f PruebaReemplazarCasosErroneosArroba

function PruebaEliminarCasoNoAmbiguoArroba ()
{
if [[ $4 == "" ]] ; then
sed -r -i 's/\"$3\"//g' $1
else
sed -r -i 's/\"$3\"/\"$4\"/g' $1
fi
}
export -f PruebaEliminarCasoNoAmbiguoArroba

function PruebaFechaTextoANumero ()
{
gawk -i inplace -F'@' 'BEGIN{IGNORECASE=1} NR==1 {print $0}
NR>1 {if($2!="") {sub( /JANUARY/ , "01" , \"$2\" ) ;
print $0} else {print $0}}' OFS='@' "$1"
gawk -i inplace -F'@' 'BEGIN{IGNORECASE=1} NR==1 {print $0}
NR>1 {if($2!="") {sub( /FEBRUARY/ , "02" , \"$2\" ) ;
print $0} else {print $0}}' OFS='@' "$1"
gawk -i inplace -F'@' 'BEGIN{IGNORECASE=1} NR==1 {print $0}
NR>1 {if($2!="") {sub( /MARCH/ , "03" , \"$2\" ) ;
print $0} else {print $0}}' OFS='@' "$1"
}

```

```

    gawk -i inplace -F'@' 'BEGIN{IGNORECASE=1} NR==1 {print $0}
NR>1 {if("${2}"!="") {sub( /APRIL/      , "04" , "${2}" ) } ;
print $0} else {print $0}}' OFS='@' "$1"
    gawk -i inplace -F'@' 'BEGIN{IGNORECASE=1} NR==1 {print $0}
NR>1 {if("${2}"!="") {sub( /MAY/        , "05" , "${2}" ) } ;
print $0} else {print $0}}' OFS='@' "$1"
    gawk -i inplace -F'@' 'BEGIN{IGNORECASE=1} NR==1 {print $0}
NR>1 {if("${2}"!="") {sub( /JUNE/       , "06" , "${2}" ) } ;
print $0} else {print $0}}' OFS='@' "$1"
    gawk -i inplace -F'@' 'BEGIN{IGNORECASE=1} NR==1 {print $0}
NR>1 {if("${2}"!="") {sub( /JULY/      , "07" , "${2}" ) } ;
print $0} else {print $0}}' OFS='@' "$1"
    gawk -i inplace -F'@' 'BEGIN{IGNORECASE=1} NR==1 {print $0}
NR>1 {if("${2}"!="") {sub( /AUGUST/    , "08" , "${2}" ) } ;
print $0} else {print $0}}' OFS='@' "$1"
    gawk -i inplace -F'@' 'BEGIN{IGNORECASE=1} NR==1 {print $0}
NR>1 {if("${2}"!="") {sub( /SEPTEMBER/ , "09" , "${2}" ) } ;
print $0} else {print $0}}' OFS='@' "$1"
    gawk -i inplace -F'@' 'BEGIN{IGNORECASE=1} NR==1 {print $0}
NR>1 {if("${2}"!="") {sub( /OCTOBER/   , "10" , "${2}" ) } ;
print $0} else {print $0}}' OFS='@' "$1"
    gawk -i inplace -F'@' 'BEGIN{IGNORECASE=1} NR==1 {print $0}
NR>1 {if("${2}"!="") {sub( /NOVEMBER/  , "11" , "${2}" ) } ;
print $0} else {print $0}}' OFS='@' "$1"
    gawk -i inplace -F'@' 'BEGIN{IGNORECASE=1} NR==1 {print $0}
NR>1 {if("${2}"!="") {sub( /DECEMBER/  , "12" , "${2}" ) } ;
print $0} else {print $0}}' OFS='@' "$1"
    gawk -i inplace -F'@' 'BEGIN{IGNORECASE=1} NR==1 {print $0}
NR>1 {if("${2}"!="") {sub( /JAN/       , "01" , "${2}" ) } ;
print $0} else {print $0}}' OFS='@' "$1"
    gawk -i inplace -F'@' 'BEGIN{IGNORECASE=1} NR==1 {print $0}
NR>1 {if("${2}"!="") {sub( /FEB/       , "02" , "${2}" ) } ;
print $0} else {print $0}}' OFS='@' "$1"
    gawk -i inplace -F'@' 'BEGIN{IGNORECASE=1} NR==1 {print $0}
NR>1 {if("${2}"!="") {sub( /MAR/      , "03" , "${2}" ) } ;
print $0} else {print $0}}' OFS='@' "$1"

```

```

    gawk -i inplace -F'@' 'BEGIN{IGNORECASE=1} NR==1 {print $0}
NR>1 {if($"{$2}"!="") {sub( /APR/          , "04" , $"{$2}" ) ;
print $0} else {print $0}}' OFS='@' "${1}"
    gawk -i inplace -F'@' 'BEGIN{IGNORECASE=1} NR==1 {print $0}
NR>1 {if($"{$2}"!="") {sub( /MAY/          , "05" , $"{$2}" ) ;
print $0} else {print $0}}' OFS='@' "${1}"
    gawk -i inplace -F'@' 'BEGIN{IGNORECASE=1} NR==1 {print $0}
NR>1 {if($"{$2}"!="") {sub( /JUN/          , "06" , $"{$2}" ) ;
print $0} else {print $0}}' OFS='@' "${1}"
    gawk -i inplace -F'@' 'BEGIN{IGNORECASE=1} NR==1 {print $0}
NR>1 {if($"{$2}"!="") {sub( /JUL/          , "07" , $"{$2}" ) ;
print $0} else {print $0}}' OFS='@' "${1}"
    gawk -i inplace -F'@' 'BEGIN{IGNORECASE=1} NR==1 {print $0}
NR>1 {if($"{$2}"!="") {sub( /AUG/          , "08" , $"{$2}" ) ;
print $0} else {print $0}}' OFS='@' "${1}"
    gawk -i inplace -F'@' 'BEGIN{IGNORECASE=1} NR==1 {print $0}
NR>1 {if($"{$2}"!="") {sub( /SEP/          , "09" , $"{$2}" ) ;
print $0} else {print $0}}' OFS='@' "${1}"
    gawk -i inplace -F'@' 'BEGIN{IGNORECASE=1} NR==1 {print $0}
NR>1 {if($"{$2}"!="") {sub( /OCT/          , "10" , $"{$2}" ) ;
print $0} else {print $0}}' OFS='@' "${1}"
    gawk -i inplace -F'@' 'BEGIN{IGNORECASE=1} NR==1 {print $0}
NR>1 {if($"{$2}"!="") {sub( /NOV/          , "11" , $"{$2}" ) ;
print $0} else {print $0}}' OFS='@' "${1}"
    gawk -i inplace -F'@' 'BEGIN{IGNORECASE=1} NR==1 {print $0}
NR>1 {if($"{$2}"!="") {sub( /DEC/          , "12" , $"{$2}" ) ;
print $0} else {print $0}}' OFS='@' "${1}"
}
export -f PruebaFechaTextoANumero

function PruebaNormalizarSeparadorInterno ()
{
    gawk -i inplace -F'@' 'NR==1 {print $0} NR>1
{if($"{$2}"!="") {gsub( / / , "_" , $"{$2}" ) ; print $0}
else {print $0}}' OFS='@' "${1}"
}

```

```

    gawk -i inplace -F'@' 'NR==1 {print $0} NR>1
    {if($""$2}""!="") {gsub( /\// , "_" , $""$2}"" ) ; print $0}
    else {print $0}}' OFS='@' "$1}"
    gawk -i inplace -F'@' 'NR==1 {print $0} NR>1
    {if($""$2}""!="") {gsub( /-/ , "_" , $""$2}"" ) ; print $0}
    else {print $0}}' OFS='@' "$1}"
}
export -f PruebaNormalizarSeparadorInterno

function PruebaReordenarFechaInterno ()
{
    gawk -i inplace -F'@' 'NR==1 {print $""$2}""} NR>1 {print
    gensub(/([0-9]{2})_([0-9]{2})_([0-9]{4})/, "\3_\2_\1", "g",
    $""$2}"" )
    }' OFS='@' "$1}"
    # cat $1} | cut -d'@' -f"$2}" --complement > tmp02.csv
    # paste -d'@' tmp01.csv tmp02.csv > $1}
    # rm -rf tmp01.csv tmp02.csv
}
export -f PruebaReordenarFechaInterno

```

FuncionesCoordenadas.sh

```

# Cotejar con conversiones a mano en la herramienta en linea
# http://www.rcn.montana.edu/resources/converter.aspx

# Generar patrones con
# | sed -r 's/[0-9]/0/g;s/\././g' | sed -r ':a;s/00/0/g;ta' |
sort | uniq
# Obtenemos estos patrones
# 0d0'0"W 0d0'0"S 1 1 1 1 1 1
# 0d0'0"W 0d0'S 1 1 1 1 1 0
# 0d0'0"W 0dS 1 1 1 1 0 0
# 0d0'W 0d0'0"S 1 1 0 1 1 1
# 0d0'W 0d0'S 1 1 0 1 1 0
# 0d0'W 0dS 1 1 0 1 0 0
# 0dW 0d0'0"S 1 0 0 1 1 1

```

```

# 0dW 0d0'S          1 0 0 1 1 0
# 0dW 0dS            1 0 0 1 0 0
# Verificar columnas
# | gawk -F',' '{print NF}' \
# | sort | uniq -c

function EliminarRegistroMezclado ()
{
    sed -r -i 's/[0-9]{2}\/[0-9]{2}\/[0-9]{4}\/g' ${1}
}
export -f EliminarRegistroMezclado

function FormatoCoordenadaDecimalArroba ()
{
    gawk -i inplace -F'@' 'NR==1 {print $1} NR>1 {if($1!="")
) {printf "%.6f\n", $1} else {print ""}}' OFS='@' ${1}
    gawk -i inplace -F'@' 'NR==1 {print $1} NR>1 {if($1!="0")
) {printf "%.6f\n", $1} else {print ""}}' OFS='@' ${1}
    gawk -i inplace -F'@' 'NR==1 {print $1} NR>1
{if($1!="0.000000") {printf "%.6f\n", $1} else {print ""}}'
OFS='@' ${1}
}
export -f FormatoCoordenadaDecimalArroba

function DDToDMSArroba ()
{
    dos2unix ${1} &> /dev/null
    gawk -F'@' 'NR>1 {if($1!="") {printf "%.6f %.6f %.6f\n", $2,
$1, "0"}
                    else {print $2, $1, ""}}' ${1} \
    | cs2cs          +proj=latlong +datum=WGS84 \
    | gawk -F' ' '{if($1=="0dE") {print "" }
                    else {print $1, $2}}' \
    | sed -r "s/([0-9\.]*)d([0-9\.]*)'([0-9\.]*)\"W ([0-
9\.]*)d([0-9\.]*)'([0-9\.]*)\"S/\1@2@3@4@5@6/
                    s/([0-9\.]*)d([0-9\.]*)'([0-9\.]*)\"W ([0-
9\.]*)d([0-9\.]*)'S/\1@2@3@4@5@/

```

```

s/([0-9\\.]*d([0-9\\.]*)'([0-9\\.]*)\\"W ([0-
9\\.]*)dS/\\1@\\2@\\3@\\4@@/
s/([0-9\\.]*d([0-9\\.]*)'W ([0-9\\.]*)d([0-
9\\.]*)'([0-9\\.]*)\\"S/\\1@\\2@@\\3@\\4@\\5/
s/([0-9\\.]*d([0-9\\.]*)'W ([0-9\\.]*)d([0-
9\\.]*)'S/\\1@\\2@@\\3@\\4@/
s/([0-9\\.]*d([0-9\\.]*)'W ([0-
9\\.]*)dS/\\1@\\2@@\\3@@/
s/([0-9\\.]*dW ([0-9\\.]*)d([0-9\\.]*)'([0-
9\\.]*)\\"S/\\1@@@\\2@\\3@\\4/
s/([0-9\\.]*dW ([0-9\\.]*)d([0-
9\\.]*)'S/\\1@@@@\\2@\\3@/
s/([0-9\\.]*dW ([0-9\\.]*)dS/\\1@@@@\\2@@/" \\
| sed -r
'1i\\long_gr@long_min@long_seg@lat_gr@lat_min@lat_seg' \\
> ${2}
cp ${2} tmp_ddtodms.csv
gawk -i inplace -F'@' 'NR==1 {printf "%s@s\\n", $1, $2}
NR>1 {if($1!="") {printf
"%0.6f@%0.6f\\n", $1, $2}
else {print "@" } }'
OFS='@' ${1}
gawk -i inplace -F'@' 'NR==1 {print $0} NR>1
{if($1!="0.000000") {print $0} else {print "@"}}' OFS='@' ${1}
paste -d'@' tmp_ddtodms.csv ${1} > ${2}
gawk -i inplace -F'@' 'NR==1 {print $0} NR>1 {if($7!="")
{print $0} else {print "#####"}}' OFS='@' ${2}
gawk -i inplace -F'@' '{print $4,$5,$6,$1,$2,$3,$7,$8}'
OFS='@' ${2}
rm -rf tmp_ddtodms.csv
}
export -f DDTtoDMSArroba

function DDTtoDMSComa ()
{
dos2unix ${1} &> /dev/null

```

```

gawk -F',' 'NR>1 {if($1!="") {printf "%.6f %.6f %.6f\n", $2,
$1, "0"}
                else {print $2, $1, ""}}' ${1} \
| cs2cs +proj=latlong +datum=WGS84 \
| gawk -F' ' '{if($1=="0dE") {print "" }
                else {print $1, $2}}' \
| sed -r "s/([0-9\\.]*)([0-9\\.]*)'([0-9\\.]*)\"W ([0-
9\\.]*)([0-9\\.]*)'([0-9\\.]*)\"S/\\1,\\2,\\3,\\4,\\5,\\6/
                s/([0-9\\.]*)([0-9\\.]*)'([0-9\\.]*)\"W ([0-
9\\.]*)([0-9\\.]*)'S/\\1,\\2,\\3,\\4,\\5,/
                s/([0-9\\.]*)([0-9\\.]*)'([0-9\\.]*)\"W ([0-
9\\.]*)([0-9\\.]*)'dS/\\1,\\2,\\3,\\4,,/
                s/([0-9\\.]*)([0-9\\.]*)'W ([0-9\\.]*)([0-
9\\.]*)'([0-9\\.]*)\"S/\\1,\\2,,\\3,\\4,\\5/
                s/([0-9\\.]*)([0-9\\.]*)'W ([0-9\\.]*)([0-
9\\.]*)'S/\\1,\\2,,\\3,\\4,/
                s/([0-9\\.]*)([0-9\\.]*)'W ([0-
9\\.]*)([0-9\\.]*)'dS/\\1,\\2,,\\3,,/
                s/([0-9\\.]*)([0-9\\.]*)'dW ([0-9\\.]*)([0-9\\.]*)'([0-
9\\.]*)\"S/\\1,,,\\2,\\3,\\4/
                s/([0-9\\.]*)([0-9\\.]*)'dW ([0-9\\.]*)([0-
9\\.]*)'S/\\1,,,\\2,\\3,/
                s/([0-9\\.]*)([0-9\\.]*)'dW ([0-9\\.]*)'dS/\\1,,,\\2,,/" \
| sed -r
'1i\\long_gr,long_min,long_seg,lat_gr,lat_min,lat_seg' \
> ${2}
cp ${2} tmp_ddtodms.csv
gawk -i inplace -F',' 'NR==1 {printf "%s,%s\n", $1, $2}
NR>1 {if($1!="") {printf
"%s,%s\n", $1, $2}
                else {print ", " }}}'
OFS=', ' ${1}
paste -d',' tmp_ddtodms.csv ${1} > ${2}
gawk -i inplace -F',' 'NR==1 {print $0} NR>1 {if($7!="")
{print $0} else {print ",,,,,,,,,"}}' OFS=', ' ${2}
gawk -i inplace -F',' '{print $4,$5,$6,$1,$2,$3,$7,$8}'
OFS=', ' ${2}

```

```

    rm -rf tmp_ddtodms.csv
}
export -f DDToDMSComa

#    0,0,,0,0,
#    0,0,,0,0,0
#    0,0,0,0,0,0

function DMSToDDArroba ()
{
    dos2unix ${1} &> /dev/null
    gawk -F'@' 'NR>1 {print $4, $5, $6, $1, $2, $3}' OFS='@' ${1}
\
    | sed -r "s/^([0-9\.]+)@([0-9\.]+)@([0-9\.]+)@([0-
9\.]+)@([0-9\.]+)@([0-9\.]+)/\1d\2'\3\"W \4d\5'\6\"S/
        s/^([0-9\.]+)@([0-9\.]+)@([0-9\.]+)@([0-
9\.]+)@([0-9\.]+)@(\1d\2'\3\"W \4d\5'S/
        s/^([0-9\.]+)@([0-9\.]+)@([0-9\.]+)@([0-
9\.]+)@@(\1d\2'\3\"W \4dS/
        s/^([0-9\.]+)@([0-9\.]+)@@([0-9\.]+)@([0-
9\.]+)@([0-9\.]+)/\1d\2'W \3d\4'\5\"S/
        s/^([0-9\.]+)@([0-9\.]+)@@([0-9\.]+)@([0-
9\.]+)@(\1d\2'W \3d\4'S/
        s/^([0-9\.]+)@([0-9\.]+)@@([0-9\.]+)@@(\1d\2'W
\3dS/
        s/^([0-9\.]+)@@@([0-9\.]+)@([0-9\.]+)@([0-
9\.]+)/\1dW \2d\3'\4\"S/
        s/^([0-9\.]+)@@@([0-9\.]+)@([0-9\.]+)@(\1dW
\2d\3'S/
        s/^([0-9\.]+)@@@([0-9\.]+)@@(\1dW \2dS/" \
    | cs2cs -f "%.6f" +proj=latlong +datum=WGS84 \
    | gawk '{if($1=="0.000000") {print "@"}
        else {print $1, $2}}' OFS='@' \
    | sed -r 'li\\decimalLongitude@decimalLatitude' \
    > ${2}
    cp ${2} tmp_dmstodd.csv
    paste -d'@' ${1} tmp_dmstodd.csv > ${2}

```

```

    gawk -i inplace -F'@' '{print $1,$2,$3,$4,$5,$6,$8,$7}'
OFS='@' ${2}
    rm -rf tmp_dmstodd.csv
}
export -f DMSToDDArroba

function DMSToDDComa ()
{
    dos2unix ${1} &> /dev/null
    gawk -F',' 'NR>1 {print $4, $5, $6, $1, $2, $3}' OFS=', ' ${1}
\
    | sed -r "s/^([0-9\.]+),([0-9\.]+),([0-9\.]+),([0-
9\.]+),([0-9\.]+),([0-9\.]+)/\1d\2'\3\"W \4d\5'\6\"S/
        s/^([0-9\.]+),([0-9\.]+),([0-9\.]+),([0-
9\.]+),([0-9\.]+)/,\1d\2'\3\"W \4d\5'S/
        s/^([0-9\.]+),([0-9\.]+),([0-9\.]+),([0-
9\.]+),,\1d\2'\3\"W \4dS/
        s/^([0-9\.]+),([0-9\.]+),,([0-9\.]+),([0-
9\.]+),([0-9\.]+)/\1d\2'W \3d\4'\5\"S/
        s/^([0-9\.]+),([0-9\.]+),,([0-9\.]+),([0-
9\.]+),,\1d\2'W \3d\4'S/
        s/^([0-9\.]+),([0-9\.]+),,([0-9\.]+),,\1d\2'W
\3dS/
        s/^([0-9\.]+),,,([0-9\.]+),([0-9\.]+),([0-
9\.]+)/\1dW \2d\3'\4\"S/
        s/^([0-9\.]+),,,([0-9\.]+),([0-9\.]+)/,\1dW
\2d\3'S/
        s/^([0-9\.]+),,,([0-9\.]+),,\1dW \2dS/" \
    | cs2cs -f "%.6f" +proj=latlong +datum=WGS84 \
    | gawk '{if($1=="0.000000") {print ","}
        else {print $1, $2}}' OFS=', ' \
    | sed -r 'li\\decimalLongitude,decimalLatitude' \
    > ${2}
    cp ${2} tmp_dmstodd.csv
    paste -d',' ${1} tmp_dmstodd.csv > ${2}
    gawk -i inplace -F',' '{print $1,$2,$3,$4,$5,$6,$8,$7}'
OFS=', ' ${2}

```

```

    rm -rf tmp_dmstodd.csv
}
export -f DMSToDDComa

function DDToDDUTMArroba ()
{
    dos2unix ${1} &> /dev/null
    gawk -F'@' 'NR>1 {print $2, $1}' OFS=' ' ${1} \
    | cs2cs -f "%.6f" +proj=latlong +datum=WGS84 +to +proj=utm
+zone=19 +south +datum=WGS84 \
    | gawk '{if($1=="11308151.532026") {print "@"}
    else {print $2, $1}}' OFS='@' \
    | sed -r 'li\\decimalLatitudeUtm@decimalLongitudeUtm' \
    > ${2}
##cp ${2} tmp_dmstodd.csv
##paste -d'@' ${1} tmp_dmstodd.csv > ${2}
##rm -rf tmp_dmstodd.csv
}
export -f DDToDDUTMArroba

function DDToDDUTMComa ()
{
    dos2unix ${1} &> /dev/null
    gawk -F',' 'NR>1 {print $2, $1}' OFS=', ' ${1} \
    | cs2cs -f "%.6f" +proj=latlong +datum=WGS84 +to +proj=utm
+zone=19 +south +datum=WGS84 \
    | gawk '{if($1=="11308151.532026") {print ","}
    else {print $2, $1}}' OFS=', ' \
    | sed -r 'li\\decimalLatitudeUtm,decimalLongitudeUtm' \
    > ${2}
cp ${2} tmp_dmstodd.csv
paste -d',' ${1} tmp_dmstodd.csv > ${2}
rm -rf tmp_dmstodd.csv
}
export -f DDToDDUTMComa

function TransferirRegistroMezcladoArroba ()

```

```

{
    paste -d'@' ${1} ${2} > tra_tmp00.csv
    gawk -F'@' 'NR==1 {print $1} NR>1 {if($1~/\./ && $2=="")
{print ""} else {print $1}}' OFS='@' tra_tmp00.csv > tra_tmp01.csv
    gawk -F'@' 'NR==1 {print $2} NR>1 {if($1!~/\./ && $2!="")
{print $2} else {print ""}}' OFS='@' tra_tmp00.csv > tra_tmp02.csv
    cp tra_tmp01.csv ${1}
    cp tra_tmp02.csv ${2}
    rm -rf tra_tmp00.csv tra_tmp01.csv tra_tmp02.csv
}
export -f TransferirRegistroMezcladoArroba

function VerificarCoordenadasArroba ()
{
    DMSToDDArroba ${1} tmp_comp01.csv
    gawk -F'@' '{print $7,$8}' OFS='@' ${1} > tmp_comp02.csv
    gawk -F'@' '{print $8,$7}' OFS='@' tmp_comp01.csv > remove.csv
    cat remove.csv > tmp_comp01.csv
    rm -rf remove.csv
    paste -d'@' tmp_comp01.csv tmp_comp02.csv > tmp_comp03.csv
    rm -rf tmp_comp01.csv
    rm -rf tmp_comp02.csv
    gawk -F'@' 'NR>1 {if($1!="" && $3!="") {if($1==$3) {print
"IDENTICO"} else {print "DISTINTO"}} else {print ""}}' OFS='@'
tmp_comp03.csv > verific_tmp01.csv
    gawk -F'@' 'NR>1 {if($2!="" && $4!="") {if($2==$4) {print
"IDENTICO"} else {print "DISTINTO"}} else {print ""}}' OFS='@'
tmp_comp03.csv > verific_tmp02.csv
    rm -rf tmp_comp03.csv
    paste -d'@' verific_tmp02.csv verific_tmp01.csv > ${2}
    rm -rf      verific_tmp01.csv verific_tmp02.csv
    sed -r -i '1i\verifdeclatr@verifdeclon' ${2}
}
export -f VerificarCoordenadasArroba

function ExportarCoordenadasGeometriaArroba ()
{

```

```

    gawk -F'@' 'NR==1 {print "id",$1,$2,"geometry"} NR>1
    {if($1!="") {print NR-1,$1,$2,"POINT ("$2" "$1)"} else {print NR-
    1,$1,$2,"" }}' OFS=', ' ${1} > ${2}
    }
    export -f ExportarCoordenadasGeometriaArroba

    function ConvertirCoordenadasGeometria ()
    {
        ogr2ogr -s_srs '+proj=latlong +datum=WGS84' -t_srs
        '+proj=latlong +datum=WGS84' -oo X_POSSIBLE_NAMES=decimalLon* -oo
        Y_POSSIBLE_NAMES=decimalLat* -oo GEOM_POSSIBLE_NAMES=geometry -f
        "ESRI Shapefile" ${2} ${1} &> /dev/null
    }
    export -f ConvertirCoordenadasGeometria

    function ImportarShapefilePgsql ()
    {
        PGPASSWORD=dbpostgres psql -d dbdwc -h localhost -U postgres -
        p 5432 -c "DROP TABLE IF EXISTS dwc.${2};"
        shp2pgsql -s ${3} -c -I ${1} dwc.${2} | PGPASSWORD=dbpostgres
        psql -d dbdwc -h localhost -U postgres -p 5432 -q
    }
    export -f ImportarShapefilePgsql

    function ExportarShapefilePgsql ()
    {
        ogr2ogr -lco GEOMETRY=AS_WKT -skipfailures -lco SEPARATOR=TAB
        -f "CSV" -overwrite ${2} ${1}
    }
    export -f ExportarShapefilePgsql

    function MergeCoordenadaExportada ()
    {
        sed -r -i 's/,/@/g'          ${1}
        sed -r -i 's/\t/@/g'        ${2}
        source activate py36
        python MergeAnyIndicator.py ${1} ${2} ${3} ${4}
    }

```

```

source deactivate
### dejaremos la ultima columna solamente
gawk -F'@' '{print $10}' ${3} > temp_merge.csv
gawk -i inplace 'NR==1 {print "zona"}
                NR>1 {if($1=="left_only") {print "Fuera de la
zona de estudio"}
                    else {print "Dentro de
la zona de estudio"}}' temp_merge.csv
cat temp_merge.csv > ${3}
rm -rf temp_merge.csv
}
export -f MergeCoordenadaExportada

function CuadrarDDToDMS ()
{
dos2unix ${1} &> /dev/null
dos2unix ${2} &> /dev/null
paste -d',' ${1} ${2} > cub_ddtodms.csv
gawk -i inplace -F',' '{print $7,$8,$9,$10}' OFS=',',
cub_ddtodms.csv
gawk -i inplace -F',' '{if($3=="0.000000" || $3=="") {print
$1,$2,$1,$2}
                    else {print
$1,$2,$3,$4}}' OFS=',', cub_ddtodms.csv
gawk -F',' 'NR==1 {print "decimalLatitude",
"decimalLongitude"} NR>1 {print $3, $4}' OFS=',', cub_ddtodms.csv >
${3}
rm -rf cub_ddtodms.csv
}
export -f CuadrarDDToDMS

function CuadrarDMSToDDArroba ()
{
dos2unix ${1} &> /dev/null
dos2unix ${2} &> /dev/null
paste -d'@' ${1} ${2} > cub_dmstodd.csv

```

```

    gawk -i inplace -F'@' '{print $7,$8,$9,$10}' OFS='@'
cub_dmstodd.csv
    gawk -i inplace -F'@' '{if($1!=""           && $3=="") {print
$1,$2,$1,$2}
                                else                               {print
$1,$2,$3,$4}}' OFS='@' cub_dmstodd.csv
    gawk -F'@' 'NR==1 {print "decimalLatitude",
"decimalLongitude"} NR>1 {print $3, $4}' OFS='@' cub_dmstodd.csv >
${3}
    rm -rf cub_dmstodd.csv
}
export -f CuadrarDMSToDDArroba

function CuadrarDMSToDDComa ()
{
    dos2unix ${1} &> /dev/null
    dos2unix ${2} &> /dev/null
    paste -d',' ${1} ${2} > cub_dmstodd.csv
    gawk -i inplace -F',' '{print $7,$8,$9,$10}' OFS=', '
cub_dmstodd.csv
    gawk -i inplace -F',' '{if($1!=""           && $3=="") {print
$1,$2,$1,$2}
                                else                               {print
$1,$2,$3,$4}}' OFS=', ' cub_dmstodd.csv
    gawk -F',' 'NR==1 {print "decimalLatitude",
"decimalLongitude"} NR>1 {print $3, $4}' OFS=', ' cub_dmstodd.csv >
${3}
    rm -rf cub_dmstodd.csv
}
export -f CuadrarDMSToDDComa

```

FuncionesTaxonomia.sh

```
declare -gx
strurl01='http://api.gbif.org/v1/species/match?name='
declare -gx
strxls01='../basesdb/xlsx2csv/01_BBDD_RM_MNHN_xlsx2csv.csv'
declare -gx
strxls02='../basesdb/xlsx2csv/02_BBDD_V_MNHN_xlsx2csv.csv'
declare -gx
strxls03='../basesdb/xlsx2csv/03_BBDD_RM_V_CONCE_xlsx2csv.csv'
declare -gx strsh001='src/01_BBDD_RM_MNHN_xlsx2csv.sh'
declare -gx strsh002='src/02_BBDD_V_MNHN_xlsx2csv.sh'
declare -gx strsh003='src/03_BBDD_RM_V_CONCE_xlsx2csv.sh'
declare -gx
strtx001='src/01_BBDD_RM_MNHN_xlsx2csv_preparar_taxize.sh'
declare -gx
strtx002='src/02_BBDD_V_MNHN_xlsx2csv_preparar_taxize.sh'
declare -gx
strtx003='src/03_BBDD_RM_V_CONCE_xlsx2csv_preparar_taxize.sh'
declare -gx
strtz001='src/01_BBDD_RM_MNHN_xlsx2csv_consultar_taxize.sh'
declare -gx
strtz002='src/02_BBDD_V_MNHN_xlsx2csv_consultar_taxize.sh'
declare -gx
strtz003='src/03_BBDD_RM_V_CONCE_xlsx2csv_consultar_taxize.sh'
declare -gx strcsv01='csv/01_BBDD_RM_MNHN_xlsx2csv.csv'
declare -gx strcsv02='csv/02_BBDD_V_MNHN_xlsx2csv.csv'
declare -gx strcsv03='csv/03_BBDD_RM_V_CONCE_xlsx2csv.csv'
declare -gx
strcvv01='../basesdb/xlsx2csv/taxonomia/01_BBDD_RM_MNHN_xlsx2csv_t
axonomia_gbif.csv'
declare -gx
strcvv02='../basesdb/xlsx2csv/taxonomia/02_BBDD_V_MNHN_xlsx2csv_ta
xonomia_gbif.csv'
declare -gx
strcvv03='../basesdb/xlsx2csv/taxonomia/03_BBDD_RM_V_CONCE_xlsx2cs
v_taxonomia_gbif.csv'
```



```
declare -gx
strxvv01='../basesdb/xlsx2csv/taxonomia/01_BBDD_RM_MNHN_xlsx2csv_t
axonomia_gbif.xlsx'
declare -gx
strxvv02='../basesdb/xlsx2csv/taxonomia/02_BBDD_V_MNHN_xlsx2csv_ta
xonomia_gbif.xlsx'
declare -gx
strxvv03='../basesdb/xlsx2csv/taxonomia/03_BBDD_RM_V_CONCE_xlsx2cs
v_taxonomia_gbif.xlsx'
```

APÉNDICE 3

ProcesarFecha.sh

```
#### Cargar funciones y variables {{{
source FuncionesFecha.sh
source BashFuncionesVarias.sh
source BashVariablesFecha.sh
#### }}}

# #### Exportar Excel Bash {{{
# ExportarExcelBash                                ${str01xls0001} 1
${str01inp0001}
# ExportarExcelBash                                ${str02xls0002} 3
${str02inp0002}
# ExportarExcelBash                                ${str03xls0003} 3
${str03inp0003}
# #### }}}

# #### Ejecucion BBDD 01 {{{
# ExtraerVariable                                  ${str01inp0001} 1
verbatimEventDate ${str01vin0001}
# CopiarVariableArchivo                            ${str01vin0001}
${str01vou0001}
# PruebaFechaTextoANumero                          ${str01vou0001} 1
# PruebaNormalizarSeparadorInterno                 ${str01vou0001} 1
# PruebaReordenarFechaInterno                     ${str01vou0001} 1
# PruebaEliminarFechaCeroFijo                     ${str01vou0001}
# ExtraerVariable                                  ${str01inp0001} 1 year
${str01vin0002}
# CopiarVariableArchivo                            ${str01vin0002}
${str01vou0002}
# PruebaFechaTextoANumero                          ${str01vou0002} 1
# PruebaNormalizarSeparadorInterno                 ${str01vou0002} 1
# PruebaReordenarFechaInterno                     ${str01vou0002} 1
# FormatearYearArroba                              ${str01vou0002} 1
```

```

# PruebaEliminarFechaCeroFijo          ${str01vou0002}
# ExtraerVariable                        ${str01inp0001} 1 month
${str01vin0003}
# CopiarVariableArchivo                  ${str01vin0003}
${str01vou0003}
# PruebaFechaTextoANumero                ${str01vou0003} 1
# PruebaNormalizarSeparadorInterno      ${str01vou0003} 1
# PruebaReordenarFechaInterno           ${str01vou0003} 1
# FormatearMonthArroba                   ${str01vou0003} 1
# PruebaEliminarFechaCeroFijo           ${str01vou0003}
# ExtraerVariable                        ${str01inp0001} 1 day
${str01vin0004}
# CopiarVariableArchivo                  ${str01vin0004}
${str01vou0004}
# PruebaFechaTextoANumero                ${str01vou0004} 1
# PruebaNormalizarSeparadorInterno      ${str01vou0004} 1
# PruebaReordenarFechaInterno           ${str01vou0004} 1
# FormatearDayArroba                    ${str01vou0004} 1
# PruebaEliminarFechaCeroFijo           ${str01vou0004}
# ExtraerVariable                        ${str01inp0001} 1
dateidentified    ${str01vin0005}
# CopiarVariableArchivo                  ${str01vin0005}
${str01vou0005}
# PruebaFechaTextoANumero                ${str01vou0005} 1
# PruebaNormalizarSeparadorInterno      ${str01vou0005} 1
# PruebaReordenarFechaInterno           ${str01vou0005} 1
# PruebaEliminarFechaCeroFijo           ${str01vou0005}
# ConcatenarParaCuadraturaArroba        ${str01vin0001}
${str01vin0002} ${str01vin0003} ${str01vin0004} ${str01raw0001}
# ConcatenarParaCuadraturaArroba        ${str01vou0001}
${str01vou0002} ${str01vou0003} ${str01vou0004} ${str01rcu0001}
# PyscriptCuadrarDataframe               ${str01rcu0001}
${str01rcu0002} fechaConv02 verbatimEventDate year month day
# ConcatenarDeADosArroba                 ${str01raw0001}
${str01rcu0002} ${str01lexi0001}
# ConcatenarDeATresArroba                ${str01lexi0001}
${str01vin0005} ${str01vou0005} ${str01rep0001}

```

```

# ExportarCsvPython                                ${str01rep0001}
${str01exo0001} 13 Arroba 1 ${str01dfc0001} drow01
# RemoverArchivoIntermedio                        ${str01exi0001}
# RemoverArchivoIntermedio                        ${str01raw0001}
# RemoverArchivoIntermedio                        ${str01rcu0001}
# RemoverArchivoIntermedio                        ${str01rcu0002}
# RemoverArchivoIntermedio                        ${str01vin0001}
# RemoverArchivoIntermedio                        ${str01vin0002}
# RemoverArchivoIntermedio                        ${str01vin0003}
# RemoverArchivoIntermedio                        ${str01vin0004}
# RemoverArchivoIntermedio                        ${str01vin0005}
# RemoverArchivoIntermedio                        ${str01vou0001}
# RemoverArchivoIntermedio                        ${str01vou0002}
# RemoverArchivoIntermedio                        ${str01vou0003}
# RemoverArchivoIntermedio                        ${str01vou0004}
# RemoverArchivoIntermedio                        ${str01vou0005}
# ##### }}}

# ##### Ejecucion BBDD 02 {{{
# ExtraerVariable                                ${str02inp0002} 1
verbatimEventDate ${str02vin0001}
# CopiarVariableArchivo                          ${str02vin0001}
${str02vou0001}
# PruebaFechaTextoANumero                        ${str02vou0001} 1
# PruebaNormalizarSeparadorInterno               ${str02vou0001} 1
# PruebaReordenarFechaInterno                    ${str02vou0001} 1
# PruebaEliminarFechaCeroFijo                    ${str02vou0001}
# ExtraerVariable                                ${str02inp0002} 1 year
${str02vin0002}
# CopiarVariableArchivo                          ${str02vin0002}
${str02vou0002}
# PruebaFechaTextoANumero                        ${str02vou0002} 1
# PruebaNormalizarSeparadorInterno               ${str02vou0002} 1
# PruebaReordenarFechaInterno                    ${str02vou0002} 1
# FormatearYearArroba                            ${str02vou0002} 1
# PruebaEliminarFechaCeroFijo                    ${str02vou0002}

```

```

# ExtraerVariable                ${str02inp0002} 1 month
${str02vin0003}
# CopiarVariableArchivo          ${str02vin0003}
${str02vou0003}
# PruebaFechaTextoANumero       ${str02vou0003} 1
# PruebaNormalizarSeparadorInterno ${str02vou0003} 1
# PruebaReordenarFechaInterno   ${str02vou0003} 1
# FormatearMonthArroba          ${str02vou0003} 1
# PruebaEliminarFechaCeroFijo   ${str02vou0003}
# ExtraerVariable                ${str02inp0002} 1 day
${str02vin0004}
# CopiarVariableArchivo          ${str02vin0004}
${str02vou0004}
# PruebaFechaTextoANumero       ${str02vou0004} 1
# PruebaNormalizarSeparadorInterno ${str02vou0004} 1
# PruebaReordenarFechaInterno   ${str02vou0004} 1
# FormatearDayArroba            ${str02vou0004} 1
# PruebaEliminarFechaCeroFijo   ${str02vou0004}
# ExtraerVariable                ${str02inp0002} 1
dateidentified    ${str02vin0005}
# CopiarVariableArchivo          ${str02vin0005}
${str02vou0005}
# PruebaFechaTextoANumero       ${str02vou0005} 1
# PruebaNormalizarSeparadorInterno ${str02vou0005} 1
# PruebaReordenarFechaInterno   ${str02vou0005} 1
# PruebaEliminarFechaCeroFijo   ${str02vou0005}
# ConcatenarParaCuadraturaArroba ${str02vin0001}
${str02vin0002} ${str02vin0003} ${str02vin0004} ${str02raw0001}
# ConcatenarParaCuadraturaArroba ${str02vou0001}
${str02vou0002} ${str02vou0003} ${str02vou0004} ${str02rcu0001}
# PyscriptCuadrarDataframe      ${str02rcu0001}
${str02rcu0002} fechaConv02 verbatimEventDate year month day
# ConcatenarDeADosArroba        ${str02raw0001}
${str02rcu0002} ${str02exi0001}
# ConcatenarDeATresArroba       ${str02exi0001}
${str02vin0005} ${str02vou0005} ${str02rep0001}

```

```

# ExportarCsvPython                                ${str02rep0001}
${str02exo0001} 13 Arroba 1 ${str02dfc0001} drow01
# RemoverArchivoIntermedio                        ${str02exi0001}
# RemoverArchivoIntermedio                        ${str02raw0001}
# RemoverArchivoIntermedio                        ${str02rcu0001}
# RemoverArchivoIntermedio                        ${str02rcu0002}
# RemoverArchivoIntermedio                        ${str02vin0001}
# RemoverArchivoIntermedio                        ${str02vin0002}
# RemoverArchivoIntermedio                        ${str02vin0003}
# RemoverArchivoIntermedio                        ${str02vin0004}
# RemoverArchivoIntermedio                        ${str02vin0005}
# RemoverArchivoIntermedio                        ${str02vou0001}
# RemoverArchivoIntermedio                        ${str02vou0002}
# RemoverArchivoIntermedio                        ${str02vou0003}
# RemoverArchivoIntermedio                        ${str02vou0004}
# RemoverArchivoIntermedio                        ${str02vou0005}
# #### }}}

# #### Ejecucion BBDD 03 {{{
# ExtraerVariable                                ${str03inp0003} 1 year
${str03vin0002}
# CopiarVariableArchivo                          ${str03vin0002}
${str03vou0002}
# PruebaFechaTextoANumero                        ${str03vou0002} 1
# PruebaNormalizarSeparadorInterno              ${str03vou0002} 1
# PruebaReordenarFechaInterno                    ${str03vou0002} 1
# PruebaEliminarCasosErroneosArroba             ${str03vou0002} 1 "3250"
# PruebaEliminarCasosErroneosArroba             ${str03vou0002} 1 "CONC"
# PruebaEliminarCasosErroneosArroba             ${str03vou0002} 1 "NA"
# PruebaReemplazarCasosErroneosArroba           ${str03vou0002} 1 O 0
# PruebaEliminarCasosErroneosArroba             ${str03vou0002} 1 "NA"
# FormatearYearArroba                            ${str03vou0002} 1
# PruebaEliminarFechaCeroFijo                    ${str03vou0002}
# ExtraerVariable                                ${str03inp0003} 1 month
${str03vin0003}
# CopiarVariableArchivo                          ${str03vin0003}
${str03vou0003}

```

```

# PruebaFechaTextoANumero          ${str03vou0003} 1
# PruebaNormalizarSeparadorInterno  ${str03vou0003} 1
# PruebaReordenarFechaInterno       ${str03vou0003} 1
# PruebaEliminarCasosErroneosArroba ${str03vou0003} 1 "22"
# FormatearMonthArroba               ${str03vou0003} 1
# PruebaEliminarFechaCeroFijo        ${str03vou0003}
# ConcatenarDeADosUnder              ${str03vou0002}
${str03vou0003} ${str03vou0001}
# PruebaEliminarFechaCeroFijo        ${str03vou0001}
# ConcatenarDeATresArroba            ${str03vou0002}
${str03vou0003} ${str03vou0001} ${str03rcu0001}
# ConcatenarDeATresArroba            ${str03vin0002}
${str03vin0003} ${str03rcu0001} ${str03rep0001}
# ExportarCsvPython                  ${str03rep0001}
${str03exo0001} 5 Arroba 1 ${str03dfc0001} drow01
# RemoverArchivoIntermedio           ${str03vin0001}
# RemoverArchivoIntermedio           ${str03vou0001}
# RemoverArchivoIntermedio           ${str03vin0002}
# RemoverArchivoIntermedio           ${str03vou0002}
# RemoverArchivoIntermedio           ${str03vin0003}
# RemoverArchivoIntermedio           ${str03vou0003}
# RemoverArchivoIntermedio           ${str03rcu0001}
# RemoverArchivoIntermedio           ${str03exi0001}
# #### }}}

# #### Comprimir fechas corregidas {{{
# 7za a -t7z -mx=9 -aoa ${str00zip0001} ${str01exo0001}
${str02exo0001} ${str03exo0001}
# #### }}}

# #### separando fechas en grupo obsoleto {{{
# SepararFechas                       ${strinp0001} 1
${strcol0001}
# SepararFechas                       ${strinp0002} 1
${strcol0002}
# SepararFechas                       ${strinp0003} 1
${strcol0003}

```

```

# PruebaQuitarColumnaArroba          ${strcol0003}  1
# PruebaQuitarColumnaArroba          ${strcol0003}  3
# PruebaQuitarColumnaArroba          ${strcol0003}  3
# #### }}}

# #### genera reporte de patrones de fechas distintas todas fechas
{{{
# echo PRE VALIDACION
# ( ListarPatronesFecha ${strcol0001} 1 ;
#   ListarPatronesFecha ${strcol0002} 1 ;
#   ListarPatronesFecha ${strcol0003} 1 ) \
#   | sed -r 'li\archivo:columna:numero-columna:patron-
fecha:cantidad' \
#   # | sed -r 's/:/\t/g'
# ## genera reporte valores fuera de rango fechas individuales dÃa
mes aÃo
# echo PRE VALIDACION
# ( ListarValorFueraRangoFecha        ${strcol0001} 1 ;
#   ListarValorFueraRangoFecha        ${strcol0002} 1 ;
#   ListarValorFueraRangoFecha        ${strcol0003} 1 ) \
#   | sed -r 'li\archivo:columna:numero-
columna:mensaje:valor:cantidad' \
#   # | sed -r 's/:/\t/g'
# #### }}}

# #### Ejecucion BBDD 03 obsoleto {{{
# PruebaEliminarCasoNoAmbiguoArroba  ${strcol0001} 1 " 00:00:00"
# PruebaEliminarCasoNoAmbiguoArroba  ${strcol0001} 2 " 00:00:00"
# PruebaEliminarCasosErroneosArroba  ${strcol0003} 1 "3250"
# PruebaEliminarCasosErroneosArroba  ${strcol0003} 1 "CONC"
# PruebaEliminarCasosErroneosArroba  ${strcol0003} 1 "NA"
# PruebaReemplazarCasosErroneosArroba ${strcol0003} 1 0 0
# PruebaEliminarCasosErroneosArroba  ${strcol0003} 2 "22"
# PruebaEliminarCasosErroneosArroba  ${strcol0003} 1 "NA"
# PruebaFechaTextoANumero            ${strcol0001} 1
# PruebaFechaTextoANumero            ${strcol0001} 5
# PruebaReordenarFechaInterno        ${strcol0001} 5

```

```

# PruebaNormalizarSeparadorInterno      ${strcol0001} 1
# PruebaNormalizarSeparadorInterno      ${strcol0001} 5
# PruebaReordenarFechaInterno           ${strcol0001} 1
# PruebaFechaTextoANumero                ${strcol0002} 5
# PruebaNormalizarSeparadorInterno      ${strcol0002} 5
# PruebaReordenarFechaInterno           ${strcol0002} 5
# SepararFechasSimples                   ${strcol0001} 1
${strsim0001}
# PruebaReordenarColumnasFechaArroba    ${strsim0001}
# SepararFechasSimples                   ${strcol0002} 1
${strsim0002}
# SepararFechasSimples                   ${strcol0003} 1
${strsim0003}
# SepararFechasCompuestas                ${strcol0001} 1
${strcom0001}
# SepararFechasCompuestas                ${strcol0002} 1
${strcom0002}
# #### }}}

# #### genera reporte de patrones de fechas distintas todas fechas
{{{
# echo POST VALIDACION
# ( ListarPatronesFecha ${strcol0001} 1 ;
#   ListarPatronesFecha ${strcol0002} 1 ;
#   ListarPatronesFecha ${strcol0003} 1 ) \
#     | sed -r 'li\archivo:columna:numero-columna:patron-
fecha:cantidad' \
#   # | sed -r 's/:/\t/g'
# ## genera reporte valores fuera de rango fechas individuales dÃa
mes aÃo
# echo POST VALIDACION
# ( ListarValorFueraRangoFecha      ${strcol0001} 1 ;
#   ListarValorFueraRangoFecha      ${strcol0002} 1 ;
#   ListarValorFueraRangoFecha      ${strcol0003} 1 ) \
#     | sed -r 'li\archivo:columna:numero-
columna:mensaje:valor:cantidad' \
#   # | sed -r 's/:/\t/g'

```

```
# #### }}}
```

ProcesarCoordenada.sh

```
#### Cargar funciones y variables {{{
source FuncionesCoordenada.sh
source BashFuncionesVarias.sh
source BashVariablesCoordenada.sh
#### }}}}

# #### Exportar Excel Bash {{{
# ExportarExcelBash                ${str01xls0001} 1
${str01inp0001}
# ExportarExcelBash                ${str02xls0002} 3
${str02inp0002}
# ExportarExcelBash                ${str03xls0003} 3
${str03inp0003}
# #### }}}}

#### Ejecucion BBDD 01 {{{
ExtraerVariable                    ${str01inp0001} 1
decimallatitude    ${str01lan0001}
CopiarVariableArchivo              ${str01lan0001}
${str01lau0001}
EliminarRegistroMezclado          ${str01lau0001}
ExtraerVariable                    ${str01inp0001} 1
verbatimlat_gr    ${str01lan0002}
CopiarVariableArchivo              ${str01lan0002}
${str01lau0002}
EliminarRegistroMezclado          ${str01lau0002}
ExtraerVariable                    ${str01inp0001} 1
verbatimlat_min    ${str01lan0003}
CopiarVariableArchivo              ${str01lan0003}
${str01lau0003}
EliminarRegistroMezclado          ${str01lau0003}
ExtraerVariable                    ${str01inp0001} 1
verbatimlat_seg    ${str01lan0004}
```

CopiarVariableArchivo	{str01lan0004}
{str01lau0004}	
EliminarRegistroMezclado	{str01lau0004}
ExtraerVariable	{str01inp0001} 1
decimallongitude {str01lon0001}	
CopiarVariableArchivo	{str01lon0001}
{str01lou0001}	
EliminarRegistroMezclado	{str01lou0001}
ExtraerVariable	{str01inp0001} 1
verbatimlong_gr {str01lon0002}	
CopiarVariableArchivo	{str01lon0002}
{str01lou0002}	
EliminarRegistroMezclado	{str01lou0002}
ExtraerVariable	{str01inp0001} 1
verbatimlong_min {str01lon0003}	
CopiarVariableArchivo	{str01lon0003}
{str01lou0003}	
EliminarRegistroMezclado	{str01lou0003}
ExtraerVariable	{str01inp0001} 1
verbatimlong_seg {str01lon0004}	
CopiarVariableArchivo	{str01lon0004}
{str01lou0004}	
EliminarRegistroMezclado	{str01lou0004}
TransferirRegistroMezcladoArroba	{str01lau0002}
{str01lau0001}	
TransferirRegistroMezcladoArroba	{str01lou0002}
{str01lou0001}	
ConcatenarDeASeisArroba	{str01lau0002}
{str01lau0003} {str01lau0004} {str01lou0002} {str01lou0003}	
{str01lou0004} {str01seo0001}	
ConcatenarDeADosArroba	{str01lau0001}
{str01lou0001} {str01duo0001}	
DMSToDDArroba	{str01seo0001}
{str01sec0001}	
CuadrarDMSToDDArroba	{str01sec0001}
{str01duo0001} {str01cud0001}	

```

    DDToDMSArroba                ${str01cud0001}
${str01duc0001}
    ConcatenarDeASeisArroba      ${str01lan0002}
${str01lan0003} ${str01lan0004} ${str01lon0002} ${str01lon0003}
${str01lon0004} ${str01cas0001}
    ConcatenarDeADosArroba      ${str01lan0001}
${str01lon0001} ${str01cad0001}
    ConcatenarDeADosArroba      ${str01cas0001}
${str01cad0001} ${str01goo0001}
    DDToDDUTMArroba            ${str01cud0001}
${str01utm0001}
    CopiarVariableArchivo        ${str01duc0001}
${str01goc0001}
    ConcatenarDeADosArroba      ${str01goo0001}
${str01goc0001} ${str01t010001}
    VerificarCoordenadasArroba  ${str01duc0001}
${str01ver0001}
    ConcatenarDeADosArroba      ${str01t010001}
${str01ver0001} ${str01t020001}
    ConcatenarDeADosArroba      ${str01t020001}
${str01utm0001} ${str01t030001}
    ExportarCoordenadasGeometriaArroba ${str01cud0001}
${str01cli0001}
    ConvertirCoordenadasGeometria ${str01cli0001}
${str01sli0001}
    ImportarShapefilePgsql      ${str01sli0001}
insint_base01 4326
    #Pendiente incorporar clip
    ExportarShapefilePgsql      ${str01slo0001}
${str01clo0001}
    MergeCoordenadaExportada    ${str01cli0001}
${str01clo0001} ${str01klo0001} id
    ConcatenarDeADosArroba      ${str01t030001}
${str01klo0001} ${str01t040001}
    ExportarCsvPython           ${str01t040001}
${str01exo0001} 21 Arroba 1 ${str01dfc0001} drow01 dtype01
    RemoverArchivoIntermedio    ${str01rep0001}

```



RemoverArchivoIntermedio	\${str01utm0001}
#RemoverArchivoIntermedio	\${str01t040001}
RemoverArchivoIntermedio	\${str01t030001}
RemoverArchivoIntermedio	\${str01t020001}
RemoverArchivoIntermedio	\${str01t010001}
RemoverArchivoIntermedio	\${str01ver0001}
RemoverArchivoIntermedio	\${str01goo0001}
RemoverArchivoIntermedio	\${str01goc0001}
RemoverArchivoIntermedio	\${str01cad0001}
RemoverArchivoIntermedio	\${str01cas0001}
RemoverArchivoIntermedio	\${str01seo0001}
RemoverArchivoIntermedio	\${str01sec0001}
RemoverArchivoIntermedio	\${str01duo0001}
#RemoverArchivoIntermedio	\${str01cud0001}
RemoverArchivoIntermedio	\${str01duc0001}
RemoverArchivoIntermedio	\${str01lan0001}
RemoverArchivoIntermedio	\${str01lan0002}
RemoverArchivoIntermedio	\${str01lan0003}
RemoverArchivoIntermedio	\${str01lan0004}
RemoverArchivoIntermedio	\${str01lan0005}
RemoverArchivoIntermedio	\${str01lau0001}
RemoverArchivoIntermedio	\${str01lau0002}
RemoverArchivoIntermedio	\${str01lau0003}
RemoverArchivoIntermedio	\${str01lau0004}
RemoverArchivoIntermedio	\${str01lau0005}
RemoverArchivoIntermedio	\${str01lon0001}
RemoverArchivoIntermedio	\${str01lon0002}
RemoverArchivoIntermedio	\${str01lon0003}
RemoverArchivoIntermedio	\${str01lon0004}
RemoverArchivoIntermedio	\${str01lon0005}
RemoverArchivoIntermedio	\${str01lou0001}
RemoverArchivoIntermedio	\${str01lou0002}
RemoverArchivoIntermedio	\${str01lou0003}
RemoverArchivoIntermedio	\${str01lou0004}
RemoverArchivoIntermedio	\${str01lou0005}
#RemoverArchivoIntermedio	\${str01cli0001}
#RemoverArchivoIntermedio	\${str01clo0001}

```

#RemoverArchivoIntermedio      ${str01sli0001}
#RemoverArchivoIntermedio      ${str01slo0001}
#RemoverArchivoIntermedio      ${str01kli0001}
#### }}}

#### Ejecucion BBDD 02 {{{
  ExtraerVariable                ${str02inp0002} 1
decimallatitude   ${str02lan0001}
  CopiarVariableArchivo          ${str02lan0001}
${str02lau0001}
  EliminarRegistroMezclado      ${str02lau0001}
  ExtraerVariable                ${str02inp0002} 1
verbatimlat_gr   ${str02lan0002}
  CopiarVariableArchivo          ${str02lan0002}
${str02lau0002}
  EliminarRegistroMezclado      ${str02lau0002}
  ExtraerVariable                ${str02inp0002} 1
verbatimlat_min  ${str02lan0003}
  CopiarVariableArchivo          ${str02lan0003}
${str02lau0003}
  EliminarRegistroMezclado      ${str02lau0003}
  ExtraerVariable                ${str02inp0002} 1
verbatimlat_seg  ${str02lan0004}
  CopiarVariableArchivo          ${str02lan0004}
${str02lau0004}
  EliminarRegistroMezclado      ${str02lau0004}
  ExtraerVariable                ${str02inp0002} 1
decimallongitude ${str02lon0001}
  CopiarVariableArchivo          ${str02lon0001}
${str02lou0001}
  EliminarRegistroMezclado      ${str02lou0001}
  ExtraerVariable                ${str02inp0002} 1
verbatimlong_gr  ${str02lon0002}
  CopiarVariableArchivo          ${str02lon0002}
${str02lou0002}
  EliminarRegistroMezclado      ${str02lou0002}

```

```

ExtraerVariable                                ${str02inp0002} 1
verbatimlong_min  ${str02lon0003}
CopiarVariableArchivo                          ${str02lon0003}
${str02lou0003}
EliminarRegistroMezclado                      ${str02lou0003}
ExtraerVariable                                ${str02inp0002} 1
verbatimlong_seg  ${str02lon0004}
CopiarVariableArchivo                          ${str02lon0004}
${str02lou0004}
EliminarRegistroMezclado                      ${str02lou0004}
TransferirRegistroMezcladoArroba             ${str02lau0002}
${str02lau0001}
TransferirRegistroMezcladoArroba             ${str02lou0002}
${str02lou0001}
ConcatenarDeASeisArroba                      ${str02lau0002}
${str02lau0003} ${str02lau0004} ${str02lou0002} ${str02lou0003}
${str02lou0004} ${str02seo0001}
ConcatenarDeADosArroba                      ${str02lau0001}
${str02lou0001} ${str02duo0001}
DMSToDDArroba                                ${str02seo0001}
${str02sec0001}
CuadrarDMSToDDArroba                        ${str02sec0001}
${str02duo0001} ${str02cud0001}
DDToDMSArroba                                ${str02cud0001}
${str02duc0001}
ConcatenarDeASeisArroba                      ${str02lan0002}
${str02lan0003} ${str02lan0004} ${str02lon0002} ${str02lon0003}
${str02lon0004} ${str02cas0001}
ConcatenarDeADosArroba                      ${str02lan0001}
${str02lon0001} ${str02cad0001}
ConcatenarDeADosArroba                      ${str02cas0001}
${str02cad0001} ${str02goo0001}
DDToDDUTMArroba                              ${str02cud0001}
${str02utm0001}
CopiarVariableArchivo                          ${str02duc0001}
${str02goc0001}

```

```

ConcatenarDeADosArroba                ${str02goo0001}
${str02goc0001} ${str02t010001}
VerificarCoordenadasArroba            ${str02duc0001}
${str02ver0001}
ConcatenarDeADosArroba                ${str02t010001}
${str02ver0001} ${str02t020001}
ConcatenarDeADosArroba                ${str02t020001}
${str02utm0001} ${str02t030001}
ExportarCoordenadasGeometriaArroba    ${str02cud0001}
${str02cli0001}
ConvertirCoordenadasGeometria          ${str02cli0001}
${str02sli0001}
ImportarShapefilePgsql                 ${str02sli0001}
insint_base02 4326
#Pendiente incorporar clip
ExportarShapefilePgsql                 ${str02slo0001}
${str02clo0001}
MergeCoordenadaExportada               ${str02cli0001}
${str02clo0001} ${str02klo0001} id
ConcatenarDeADosArroba                ${str02t030001}
${str02klo0001} ${str02t040001}
ExportarCsvPython                       ${str02t040001}
${str02exo0001} 21 Arroba 1 ${str02dfc0001} drow01 dtype01
RemoverArchivoIntermedio               ${str02rep0001}
RemoverArchivoIntermedio               ${str02utm0001}
#RemoverArchivoIntermedio              ${str02t040001}
RemoverArchivoIntermedio               ${str02t030001}
RemoverArchivoIntermedio               ${str02t020001}
RemoverArchivoIntermedio               ${str02t010001}
RemoverArchivoIntermedio               ${str02ver0001}
RemoverArchivoIntermedio               ${str02goo0001}
RemoverArchivoIntermedio               ${str02goc0001}
RemoverArchivoIntermedio               ${str02cad0001}
RemoverArchivoIntermedio               ${str02cas0001}
RemoverArchivoIntermedio               ${str02seo0001}
RemoverArchivoIntermedio               ${str02sec0001}
RemoverArchivoIntermedio               ${str02duo0001}

```

```

#RemoverArchivoIntermedio      ${str02cud0001}
  RemoverArchivoIntermedio      ${str02duc0001}
  RemoverArchivoIntermedio      ${str02lan0001}
  RemoverArchivoIntermedio      ${str02lan0002}
  RemoverArchivoIntermedio      ${str02lan0003}
  RemoverArchivoIntermedio      ${str02lan0004}
  RemoverArchivoIntermedio      ${str02lan0005}
  RemoverArchivoIntermedio      ${str02lau0001}
  RemoverArchivoIntermedio      ${str02lau0002}
  RemoverArchivoIntermedio      ${str02lau0003}
  RemoverArchivoIntermedio      ${str02lau0004}
  RemoverArchivoIntermedio      ${str02lau0005}
  RemoverArchivoIntermedio      ${str02lon0001}
  RemoverArchivoIntermedio      ${str02lon0002}
  RemoverArchivoIntermedio      ${str02lon0003}
  RemoverArchivoIntermedio      ${str02lon0004}
  RemoverArchivoIntermedio      ${str02lon0005}
  RemoverArchivoIntermedio      ${str02lou0001}
  RemoverArchivoIntermedio      ${str02lou0002}
  RemoverArchivoIntermedio      ${str02lou0003}
  RemoverArchivoIntermedio      ${str02lou0004}
  RemoverArchivoIntermedio      ${str02lou0005}
#RemoverArchivoIntermedio      ${str02cli0001}
#RemoverArchivoIntermedio      ${str02clo0001}
#RemoverArchivoIntermedio      ${str02sli0001}
#RemoverArchivoIntermedio      ${str02slo0001}
#RemoverArchivoIntermedio      ${str02kli0001}
#### }}}

#### Ejecucion BBDD 03 {{{
  ExtraerVariable                ${str03inp0003} 1
decimallatitude   ${str03lan0001}
  CopiarVariableArchivo          ${str03lan0001}
${str03lau0001}
  EliminarRegistroMezclado       ${str03lau0001}
  ExtraerVariable                ${str03inp0003} 1
decimallongitude  ${str03lon0001}

```

CopiarVariableArchivo	`\${str03lon0001}`
`\${str03lou0001}`	
EliminarRegistroMezclado	`\${str03lou0001}`
ConcatenarDeADosArroba	`\${str03lau0001}`
`\${str03lou0001}` `\${str03duo0001}`	
DDToDMSArroba	`\${str03duo0001}`
`\${str03duc0001}`	
ConcatenarDeADosArroba	`\${str03lan0001}`
`\${str03lon0001}` `\${str03cas0001}`	
DDToDDUTMArroba	`\${str03duo0001}`
`\${str03utm0001}`	
ConcatenarDeADosArroba	`\${str03cas0001}`
`\${str03duc0001}` `\${str03t010001}`	
ConcatenarDeADosArroba	`\${str03t010001}`
`\${str03utm0001}` `\${str03t020001}`	
ExportarCoordenadasGeometriaArroba	`\${str03cas0001}`
`\${str03cli0001}`	
ConvertirCoordenadasGeometria	`\${str03cli0001}`
`\${str03sli0001}`	
ImportarShapefilePgsql	`\${str03sli0001}`
insint_base03 4326	
#Pendiente incorporar clip	
ExportarShapefilePgsql	`\${str03slo0001}`
`\${str03clo0001}`	
MergeCoordenadaExportada	`\${str03cli0001}`
`\${str03clo0001}` `\${str03klo0001}` id	
ConcatenarDeADosArroba	`\${str03t020001}`
`\${str03klo0001}` `\${str03t030001}`	
ExportarCsvPython	`\${str03t030001}`
`\${str03exo0001}` 13 Arroba 1 `\${str03dfc0001}` drow01 dtype01	
RemoverArchivoIntermedio	`\${str03utm0001}`
#RemoverArchivoIntermedio	`\${str03t030001}`
RemoverArchivoIntermedio	`\${str03t020001}`
RemoverArchivoIntermedio	`\${str03t010001}`
RemoverArchivoIntermedio	`\${str03goo0001}`
#RemoverArchivoIntermedio	`\${str03cas0001}`
RemoverArchivoIntermedio	`\${str03duo0001}`

```

RemoverArchivoIntermedio      ${str03duc0001}
RemoverArchivoIntermedio      ${str03lan0001}
RemoverArchivoIntermedio      ${str03lau0001}
RemoverArchivoIntermedio      ${str03lon0001}
RemoverArchivoIntermedio      ${str03lou0001}
#RemoverArchivoIntermedio     ${str03cli0001}
#RemoverArchivoIntermedio     ${str03clo0001}
#RemoverArchivoIntermedio     ${str03sli0001}
#RemoverArchivoIntermedio     ${str03slo0001}
#RemoverArchivoIntermedio     ${str03kli0001}
#### }}}

# #### Comprimir coordenadas corregidas {{{
# 7za a -t7z -mx=9 -aoa ${str00zip0001} ${str01exo0001}
${str02exo0001} ${str03exo0001}
# #### }}}

```

ProcesarTaxonomia.sh

```

source FuncionesTaxonomia.sh
source BashVariablesTaxonomia.sh
source ../src/BashFuncionesVarias.sh

#ListarCabecerasArroba ${strxls01} | gawk -F':'
'BEGIN{IGNORECASE=1} $2~/^genus$/ {print $0}'
#ListarCabecerasArroba ${strxls02} | gawk -F':'
'BEGIN{IGNORECASE=1} $2~/^genus$/ {print $0}'
#ListarCabecerasArroba ${strxls03} | gawk -F':'
'BEGIN{IGNORECASE=1} $2~/^genus$/ {print $0}'

# ListarCanonicalArroba ${strxls01} 79 81
# ListarCanonicalArroba ${strxls02} 76 78
# ListarCanonicalArroba ${strxls03} 49

# #### Preparar consultas gbif para ejecutar en paralelo
# ListarScientificArroba ${strxls01} 79 81 82 83 84
# ListarScientificArroba ${strxls02} 76 78 79 80 81

```

```

# ListarScientificArroba ${strxls03} 49 60

# ConsultarGbifParalelo ${strsh001}
# ConsultarGbifParalelo ${strsh002}
# ConsultarGbifParalelo ${strsh003}

# ExtraccionGbifJsonMultiple ${strcsv01}
# ExtraccionGbifJsonMultiple ${strcsv02}
# ExtraccionGbifJsonMultiple ${strcsv03}

# ListarCabecerasArroba ${strcsv01} | grep -i --color 'matchType'
# ListarCabecerasArroba ${strcsv02} | grep -i --color 'matchType'
# ListarCabecerasArroba ${strcsv03} | grep -i --color 'matchType'

# gawk -F'@' 'NR>1 {print $9}' ${strcsv01} | sort | uniq -c
# gawk -F'@' 'NR>1 {print $9}' ${strcsv02} | sort | uniq -c
# gawk -F'@' 'NR>1 {print $9}' ${strcsv03} | sort | uniq -c

# cp ${strcsv01} ${strcvv01}
# cp ${strcsv02} ${strcvv02}
# cp ${strcsv03} ${strcvv03}

# ExportarCsvR $( basename ${strcvv01} .csv ) ${strxvv01}
${strcvv01} $( basename ${strcvv01} .csv )
# ExportarCsvR $( basename ${strcvv02} .csv ) ${strxvv02}
${strcvv02} $( basename ${strcvv02} .csv )
# ExportarCsvR $( basename ${strcvv03} .csv ) ${strxvv03}
${strcvv03} $( basename ${strcvv03} .csv )

# #### Preparar consultas taxize para ejecutar en paralelo
# ListarPrepararTaxizeArroba ${strsh001} ${strtx001}
# ListarPrepararTaxizeArroba ${strsh002} ${strtx002}
# ListarPrepararTaxizeArroba ${strsh003} ${strtx003}

# PrepararTaxizeParalelo ${strtx001}
# PrepararTaxizeParalelo ${strtx002}
# PrepararTaxizeParalelo ${strtx003}

```



```
# ListarConsultarTaxizeParalelo ${strtx001} ${strtz001}
# ListarConsultarTaxizeParalelo ${strtx002} ${strtz002}
# ListarConsultarTaxizeParalelo ${strtx003} ${strtz003}
```

```
ConsultarTaxizeParalelo ${strtz001}
ConsultarTaxizeParalelo ${strtz002}
ConsultarTaxizeParalelo ${strtz003}
```

SIMBIO



SISTEMA DE INFORMACIÓN Y MONITOREO DE LA BIODIVERSIDAD

