Junta de Andalucía

Consejería de Economía, Hacienda y Fondos Europeos
Instituto de Estadística y Cartografía de Andalucía

aLink. Documentación técnica

Cofinanciado por la Unión Europea

MINISTERIO DE HACIENDA Y FUNCIÓN PÚBLICA

Fondos Europeos

ÍNDICE

1 .Introducción

El proposito de este documento es ofrecer al usuario la documentacion técnica de la aplicacion *aLink: Herramienta de Fusión de Ficheros* partiendo del código fuente de la aplicación de software libre desarrollada por la Universidad Nacional de Australia, Febrl. Dicho código se ha ido modificando en función de las necesidades que el Instituto de Estadística y Cartografía de Andalucía ha ido teniendo. El objetivo del documento es mostrar un análisis que facilite el posterior mantenimiento y la transferencia del conocimiento sobre la aplicación.

2 .Definición General de la Aplicación

2.1 .Definicion de la Aplicación

La aplicacion dispone de dos funcionalidades principales: la *Herramienta de Normalizacion* y la *Herramienta de Enlace*, cada una con sus ventanas individuales que desarrollaremos y describiremos en los siguientes apartados. A parte de las ventanas principales de las herramientas, está la ventana de inicio, que es la primera que se abre al iniciar la aplicacion.

2.2 .Inicio

Es la primera ventana que se abre cuando ejecutamos la aplicación. Desde ella podemos acceder a las diferentes funcionalidades, asi como salir de la aplicación.

2.3 .Herramienta de Normalización

La mayoría de la información con la que se trabaja contiene errores, está incompleta o incorrectamente formateada, se codifica de manera distinta de una fuente a otra, etc. Por consiguiente, el llamado *proceso de normalización* de datos constituye un modo de dar solución a esta situación y posibilita al usuario lograr una mejor *calidad y fiabilidad* en los posteriores análisis de dichos datos.

En el proceso de normalización llevado a cabo con esta herramienta, se distinguen **dos fases principales**:

- **Limpieza y estandarización**. Su objetivo es transformar los datos originales en otros, de modo que:
    - Tengan un formato consistente y bien definido
    - Permitan resolver las inconsistencias sobre la forma en que se presenta y codifica la información

  Mientras que en el proceso de limpieza no importa el contenido semántico del campo a normalizar, pues aquí se realizan tareas de codificación del mismo, eliminación de abreviaturas y signos de puntuación. En el proceso de estandarización, sí que se analiza el contenido semántico del campo, modificando algunos de sus valores por valores normalizados y etiquetando o clasificando el contenido según el valor de sus componentes.

- **Segmentación**. El objetivo de esta fase es separar las entidades presentes en el campo a normalizar, para facilitar las comparaciones. Así, por ejemplo, un campo que contiene una dirección postal puede ser separado su contenido en tres nuevos campos como pueden ser el tipo de vía, el nombre de la vía y el número de la vía.

Por consiguiente, la funcionalidad de esta herramienta es, dado un *fichero csv* separado por *';'*, con una matriz de datos, normalizar ciertos datos de acuerdo a una *lista de corrección* y conjunto de *tablas de búsqueda*, que limpian y estandarizan los elementos que se quieren normalizar y además, etiquetan a cada elemento que componen los valores a normalizar en función de la tabla de búsqueda en la que se encuentre dicho elemento.

Las ***listas de corrección*** son archivos con extensión *'.lst'* que permiten limpiar el fichero de datos a normalizar de caracteres o cadenas que el usuario considere oportuno eliminar o sustituir. De este modo, caracteres del tipo *'|', '$',* etc., pueden ser sustituidos por espacios en blanco o también se pueden sustituir vocales con tildes por vocales sin tildes. Estas listas se proporciona inicialmente con la aplicación *aLink* (en la ubicación: *listas_tablas**listas_de_correccion*) y se puede editar a través del menú *Herramientas>editor de listas de corrección*, añadiendo, eliminando o editando elementos de la misma. Se dispone de listas de corrección para nombres de personas, direcciones postales e identificadores de personas físicas y/o jurídicas. De entre ellas se seleccionará la correspondiente al tipo de normalización que se va a realizar.

Las ***tablas de búsqueda***,que son otro conjunto de ficheros, pero éstos con extensión *'.tbl'* que:

- sustituyen cada elemento del campo a normalizar por su valor estandarizado, y además,
- le asignan una etiqueta.

Aunque se disponen de tablas de búsqueda para la normalización de nombres de personas, direcciones postales e identificadores de personas físicas y/o jurídicas. Para cada uno de estos propósitos puede existir más de una tabla de búsqueda. No obstante, cada una de estas tablas contienen una serie de cadenas de caracteres que hacen referencia a un mismo elemento común, como por ejemplo, tipos de vías, municipios, provincias, nombres masculinos, femeninos, neutros, etc. Y todos los elementos de una misma tabla de búsqueda tienen asignada la misma etiqueta común. Así, las tablas de búsqueda no sólo van a sustituir cada elemento o valor del campo a normalizar por su valor estandarizado, al igual que las listas de corrección, sino que además, le va a asignar una etiqueta.

Estos ficheros también se proporcionan inicialmente con la aplicación *aLink* (en la ubicación: *listas_tablas*\ *tablas_de_busqueda*) y se puede editar a través del menú *Herramientas>editor de tablas de busqueda*, añadiendo, eliminando o editando elementos de la misma. Se dispone de tablas de búsqueda para nombres de personas, direcciones postales e identificadores de personas físicas y/o jurídicas. De entre ellas se seleccionará el tipo de tablas de búsqueda correspondiente al tipo de normalización que se va a realizar.

Además, dado que no siempre es evidente cómo aislar los elementos que describen una dirección o un nombre de forma clara, para la normalización de un fichero de datos se necesita disponer de otros ficheros que nos permiten segmentar la información a normalizar en una serie de campos de salida. Estos ficheros se denominan ***Modelos ocultos de Markov***, los cuales recogen el conocimiento adquirido en una fase anterior, en la que partiendo de una muestra de registros del fichero de datos que contiene el campo a normalizar, se analizó los valores de dicho campo con la idea de detectar el patrón o estructura que siguen tales datos.

De este modo, estos ficheros permiten identificar el patrón o estructura que con más probabilidad siguen los datos del campo a normalizar, segmentando a estos en los distintos elementos que lo componen.

Estos ficheros también se proporcionan inicialmente con la aplicación *aLink* (en la carpeta: *muestras_modelos*). En concreto, existen modelos disponibles para la segmentación de nombres de personas, direcciones postales e identificadores de personas físicas y/o jurídicas. No obstante, si los modelos proporcionados no se adecúan al fichero de datos a normalizar, el usuario tiene la posibilidad de construir unos nuevos. El usario dispone del *Manual de usuario de aLink* donde se puede consultar cómo construir dichos modelos en el caso en el que sea necesario.

A continuación, se muestra el esquema general de la metodología bajo la que se desarrolla un proceso de normalización de un fichero de datos:

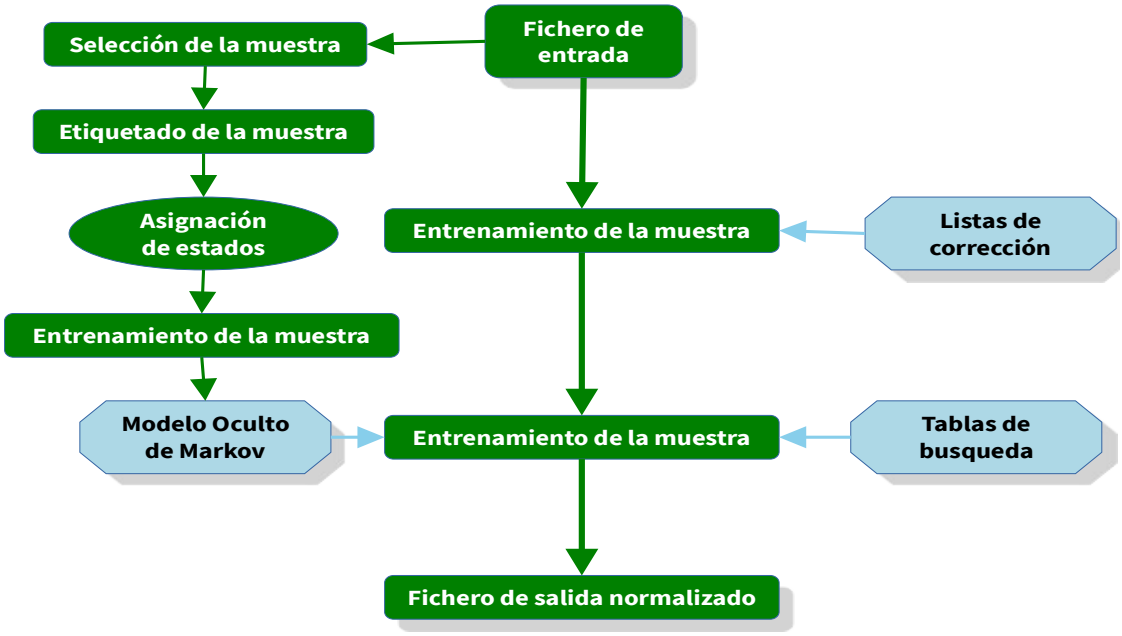


Figura 1: Proceso general de normalización usando un modelo HMM generado a partir de los datos.

Como se puede apreciar en este esquema, para el proceso de limpieza se precisa de las listas de corrección, mientras que para el proceso de estandarización y segmentación son necesarias las tablas de búsqueda y los Modelos Ocultos de Markov. Para este último proceso, se podrán utilizar los modelos que se proporcionan junto con la aplicación o habrá que generarlos previamente si se quiere trabajar con otra fuente de datos que siga un patrón diferente, esto es, que partamos de un fichero de datos a normalizar cuya información no se ajusta al patrón definido en los ficheros de los Modelos Ocultos de Markov iniciales que incorpora la aplicación.

Por ejemplo, si queremos normalizar direcciones y la dirección que tenemos es: C/ Luis Montoto 3, entonces el elemento *C/* es normalizado por *calle* y etiquetados por *TV* (que significa Tipo de Via), ya que el elemento *C/* se encuentra dentro de la tabla de búsqueda de tipos de vía (*kvia*), los elementos *luis* y *montoto* formarán parte del nombre de la vía pero como no se encuentran dentro de ninguna de las tablas de búsqueda de direcciones postales se etiquetaran como *UN* (unknow en inglés) y el elemento *3,* que constituye el número de la vía, no va a estar tampoco en ninguna tabla de búsqueda de direcciones pero se etiquetará con la etiqueta *NU* por ser un valor numérico. Estas etiquetas junto con el estado asignado a las mismas, juegan un papel fundamental a la hora de realizar la segmentación de las direcciones postales en los distintos campos

de salida. El usario dispone del *Manual de usuario de aLink* donde puede ver el uso y funcionamiento de la aplicación *aLink* para más información.

Ha de tenerse en cuenta que para el proceso de normalización que se desea llevar a cabo sea eficiente, es preciso realizar los siguientes pasos:

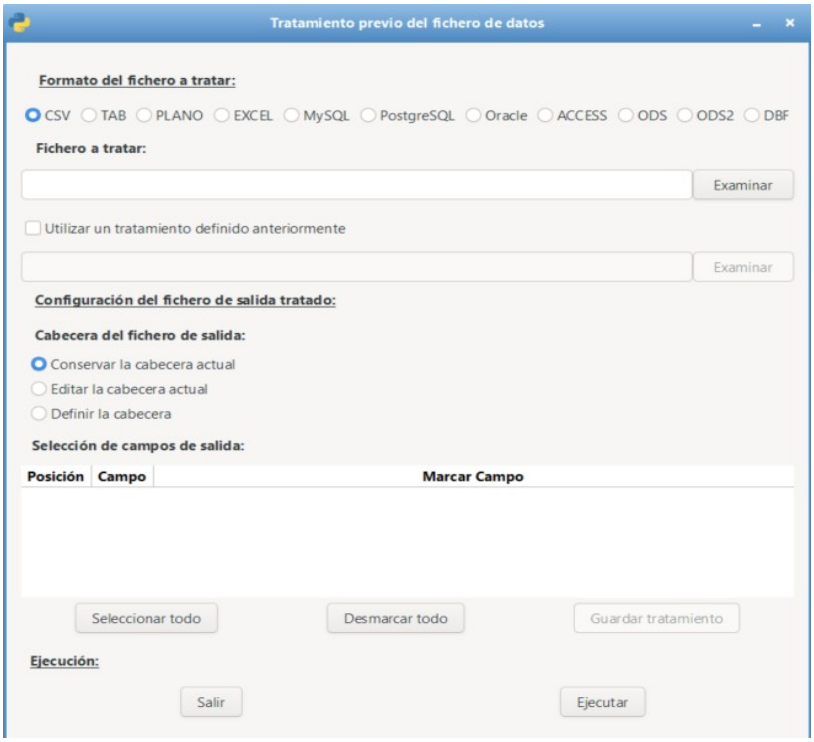
1. *Tratamiento previo del fichero de datos*, que permite transformar el fichero de origen a normalizar o enlazar, que puede presentarse en cualquiera de una serie de formatos (csv, excel, Tab, plano, etc), en un fichero de texto CSV, cuyos elementos están separados por ';', así como recodificar los datos y eliminar algunos símbolos o caracteres que por su codificación pueden provocar fallos en el proceso de normalización o enlace.

2. *Normalización del fichero de datos*, por el que se limpian, estandarizan y segmenta la información del campo a normalizar. Para llevar a cabo este paso, se indicará en la interfaz principal de la Herramienta de Normalización: el fichero a normalizar, el tipo de normalización (nombres propios, direcciones postales o NIF/DNI/NIE), el campo a normalizar, la lista de correccción a usar, así como las tablas de búsquedas y Modelo Oculto de Markov a usar y los campos de salida en los que quedará segmentada la información.

3. *Validación del proceso de normalización*, por el que se comprueba la bondad del proceso de normalización.

Para llevar a cabo el primer paso, nos dirigimos al menú superior de *Herramientas>Tratamiento previo*, y ahí se carga el fichero para tratarlo, pudiendo elegir con qué campos nos quedamos, cuales descartamos, cambiar nombres de las cabeceras, etc.

Al realizar el tratamiento previo los ficheros serán transformados a ficheros codificados a *UTF-8*, también el contenido de los campos del fichero en mayúsculas serán pasados a minúsculas y el carácter '*ñ*' será pasado a '*kk*'. El usario dispone del *Manual de usuario de aLink* donde se describen algunas otras modificaciones que se pueden realizar.



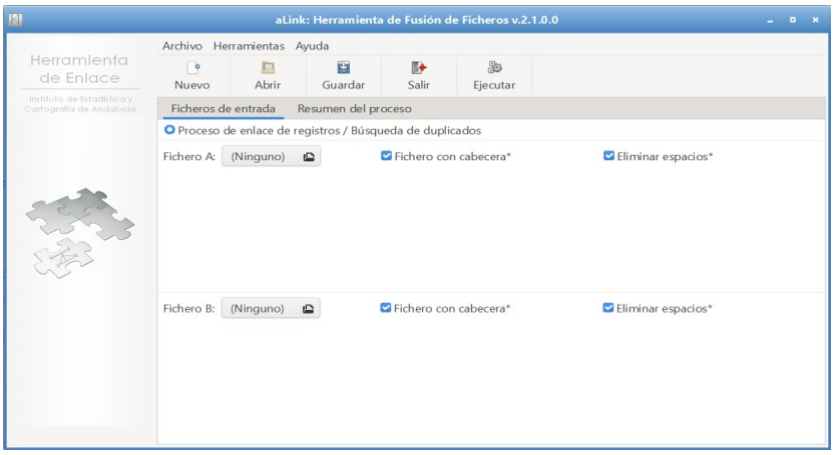
Una vez tratado el fichero de datos original, este fichero tratado es el que se recomienda usar en la herramienta de normalización.

2.4 .Herramienta de Enlace

El objetivo de esta herramienta es, dado dos ficheros con datos en *formato csv*, permite realizar de forma completa un proceso de enlace probabilístico de ambos, mediante el uso de campos con valores comunes, siendo:

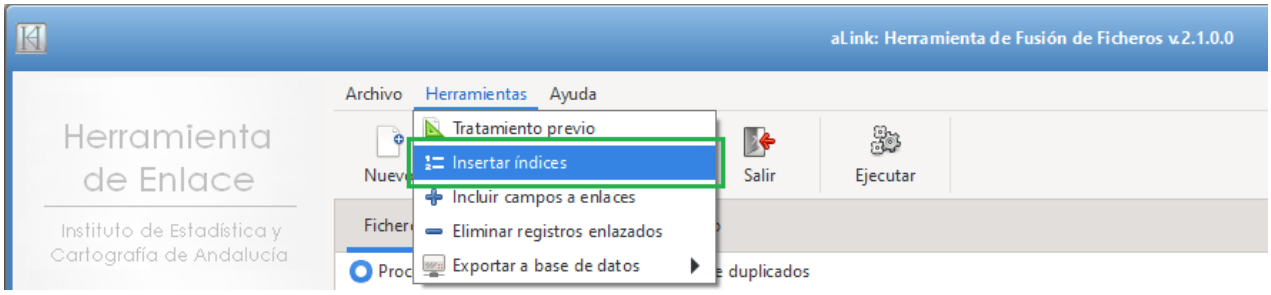
- el *Fichero A*, el fichero “al que se le quieren añadir campos”, y
- el *Fichero B*, el “fichero de referencia para la comparación de donde se quieren traer los campos”.

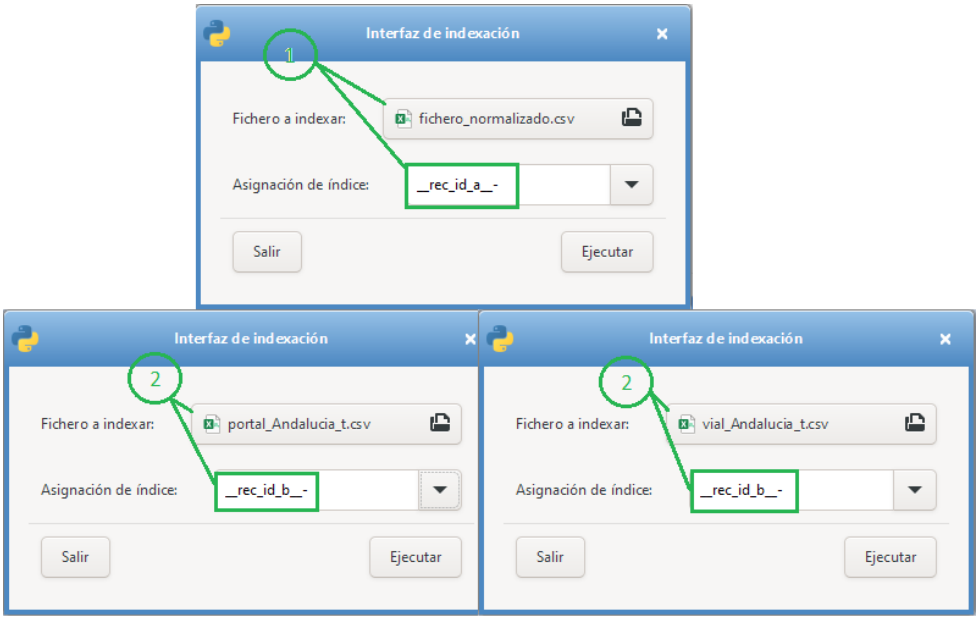
Este mismo proceso puede ser usado para la búsqueda de duplicados, introduciendo el mismo fichero como *Fichero A* y como *Fichero B*.



Para ello, es imprescindible que los ficheros que se introduzcan estén *tratados e indexados* con las herramientas que *aLink* proporciona al respecto y que tengan formato CSV separados por punto y coma.

Para indexar los ficheros nos dirigiremos al menú *Herramientas>Insertar Índices*.





En la interfaz de indexación se deben especificar el fichero a indexar y el índice. Al fichero que se quiere geocodificar se le asignará el índice *__rec_id_a_-* (correspondiente al 1), y a los ficheros de referencia el índice *__rec_id_b_-* (correspondiente al 2). Normalmente se recomienda que el fichero más pequeño sea el fichero **A** y el fichero mayor sea el fichero **B**

El área de enlace consta en un principio de dos pestañas:

1. 'Ficheros de entrada', que es la que está activada por defecto y '
2. 'Resumen del proceso'.

El resto de pestañas:

3. Pestaña de análisis exploratorio
4. Pestaña agrupación
5. Pestaña comparación
6. Pestaña clasificación
7. Pestaña salida
8. Pestaña evaluación
9. Pestaña resultados

aparecerán conforme el usuario vaya introduciendo la información de cada una de las fases del proceso de enlace y pulse el botón *Ejecutar*. Conviene remarcar que para avanzar en el proceso de enlace y aparezcan el resto de pestañas es necesario pulsar siempre el botón *Ejecutar*.

Una vez insertados los indices, se cargan los ficheros y se le da a *ejecutar*, con lo cual nos mostrará las dos siguientes pestañas:

- *Pestaña de análisis exploratorio*, que nos brindará un primer análisis exploratorio, y permite al usuario hacerse una primera idea acerca de la idoneidad de las variables candidatas a formar parte de la siguiente etapa de agrupación.

- *Pestaña agrupación*, a través de la cual se selecciona las variables por las que se realizará el proceso de enlace, con el objetivo de reducir el número de comparaciones a realizar. Pues si cada valor de la variable de agrupación da lugar a un grupo, sólo se procederá a comparar aquellos registros que coincidan en ambos ficheros en los valores de la variable de agrupación

En la definición de la variable de agrupación elegida se ofrece la posibilidad de aplicar diferentes operaciones como:

- Truncamiento de los valores de un campo, indicando la longitud máxima de caracteres que se desea considerar del campo en cuestión.

- Ordenación alfabética de las palabras que forman el campo (ordenar)

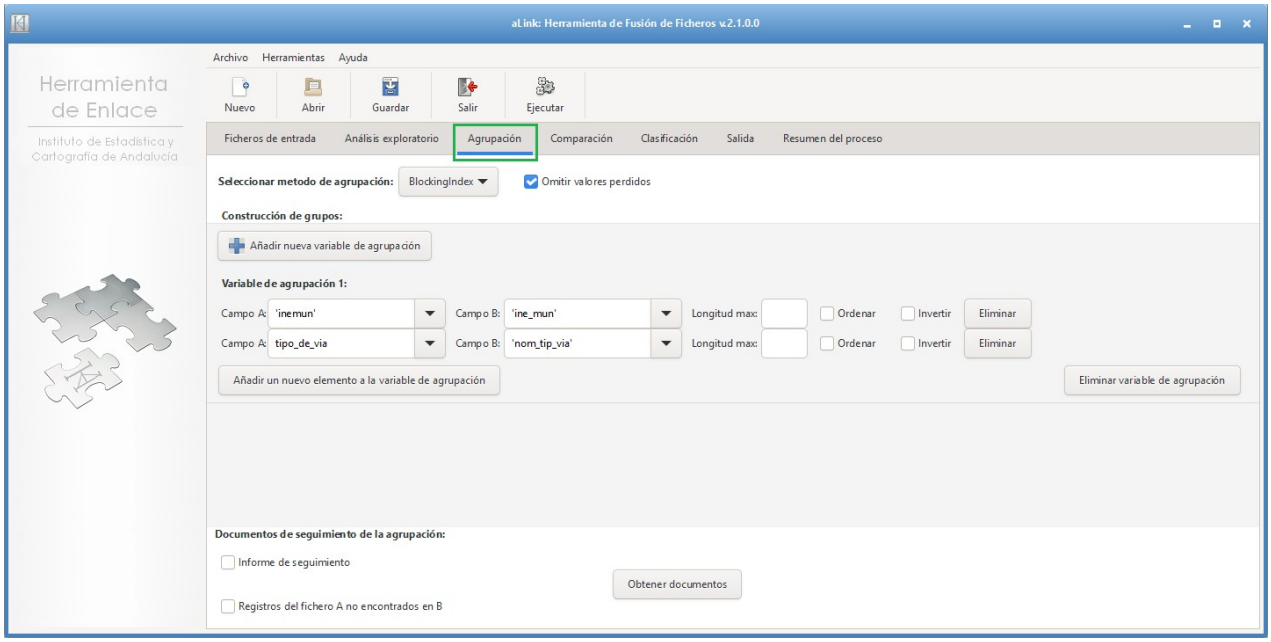
- Inversión alfabética de los valores del vampo (invertir)

Así mismo, ofrece la posibilidad de definir:

- *Una variable de agrupación como concatenación de más de un campo* de los ficheros a comparar. En este caso, se comparán en la siguiente etapa aquellos registros que tengan los mismos valores en cada campo empleado en la definición de esta variable de agrupación.

- *Dos variables de agrupación*. De este modo, en la siguiente etapa, en primer lugar se comparán todos los registros que coincidan en el primer campo. Y seguidamente, para todos aquellos registros que no coincidan en el primer campo, se compararán en función del segundo campo.

La principal utilidad de esta fase de agrupación, es generar grupos de registros en ambos ficheros de datos en función de una o varias variables de agrupación y comparar únicamente los registros que coincidan en los valores de las variables de agrupación construidas, de esta forma se consigue reducir el número de comparaciones a realizar entre los dos registros de los dos ficheros que se quieren enlazar, de manera que computacionalmente el proceso de enlace es más eficiente. El usario dispone del *Manual de usuario de aLink* donde se dan más detalles sobre esta fase.



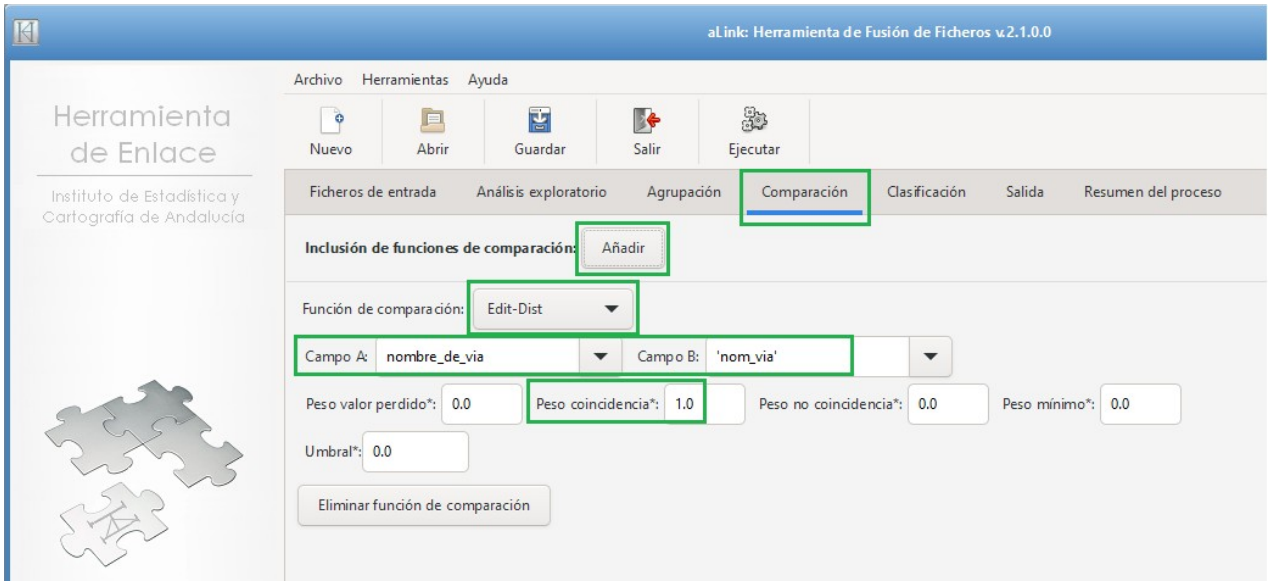
Si se le da otra vez a *Ejecutar*, aparece la pestaña de *Comparación*, la cual nos permite decidir los campos por los que se van a comparar los registros de los ficheros a enlazar. Para ello se tiene que seleccionar una función de comparación que mide la similitud entre dos cadenas de caracteres o valores numéricos. Y ésta puede ser una:

- Función de comparación de cadena exacta (si se busca coincidencia exacta entre las cadenas de caracteres comparadas),
- Función de comparación de cadena contenida (Str-Contains) o truncada (Str-Truncate)
- Función de comparación de cadena aproximada. Para obtener el grado de similitud de dos cadena se ofrece como alternativas varios algoritmos: Jaro, Winkler, Distancia de edición (o de Levenshtein), distancia de Damerau-Levenshtein, distancia Bag, distancia Smith-Waterman, Sep-Match.
- Función de comparación numérica (si compara valores numéricos absolutos o porcentajes)

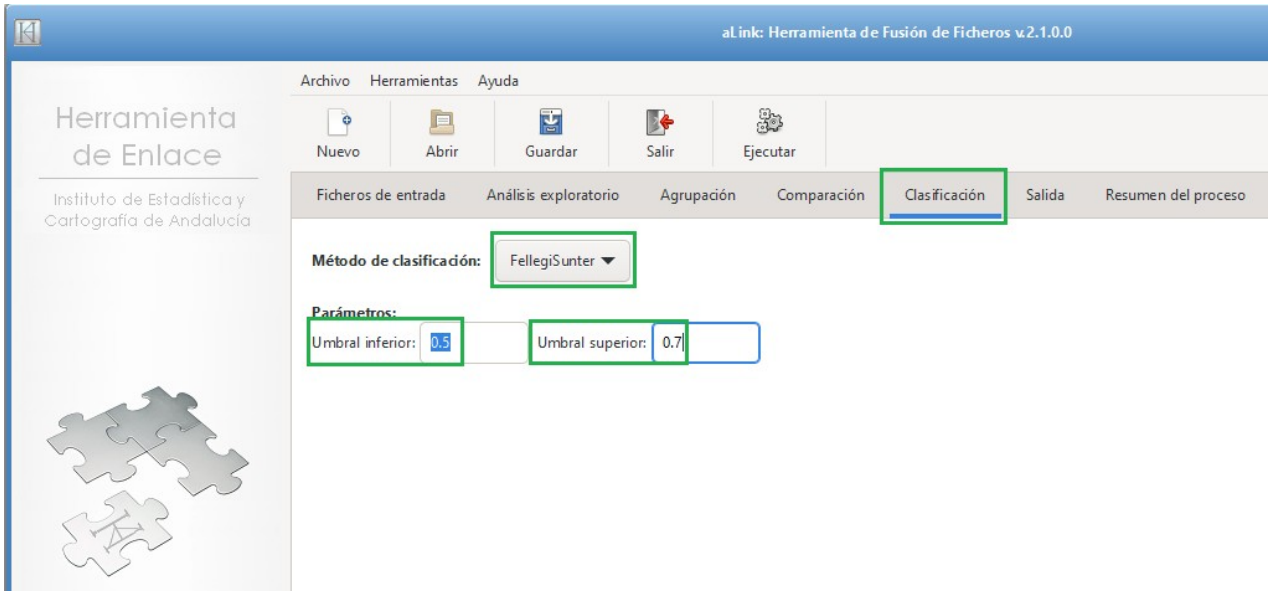
La función elegida devolverá un peso, en base al cual se decidirá que valores podrán ser considerados una coincidencia.

En esta pestaña, el usuario además podrá indicar el peso de coincidencia y el de no coincidencia, de forma que cuando solo se compara un único campo de ambos ficheros se suelen dejar los valores establecidos por defecto de 1 y 0. Si la función de comparación elegida es de cadena aproximada, el peso obtenido al comparar los valores variará entre 0 y 1, de forma que cuanto más cercano a 1 esté el valor, más parecidas

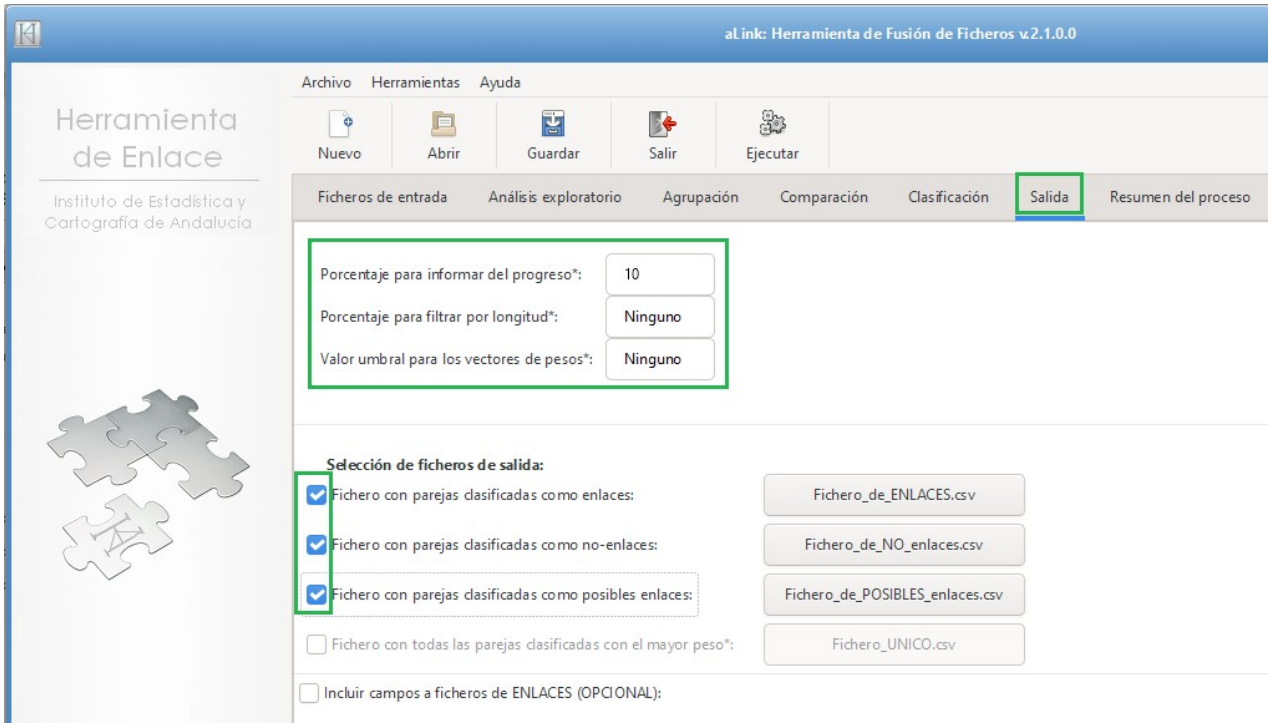
serán las cadenas de caracteres comparadas y cuanto más cercano a 0 esté el peso, más distintas serán las cadenas de caracteres.



Si se le da otra vez a *Ejecutar* aparece la petaña de *Clasificación*, en esta fase el usuario va poder establecer el método mediante el cual se van a clasificar los pares de registros comparados.



Y por ultimo, si se le vuelve a dar a *Ejecutar* aparece la petaña de *Salida*, donde se podrá indicar los ficheros de salida deseados.



Este seria un esquema de los procesos de normalizacion y enlace con *aLink*: Herramienta de fusión de ficheros.


Normalización
Enlace
Paso 0: Tratamiento (Obligatorio)
Herramientas → Tratamiento previo
Se convierten ficheros de formato plano, tab, csv, mdb, Mysql, xlsx, dbf y ods en ficheros csv separados por ";"
Paso 1: Normalización (Obligatorio)
El fichero tratado se normalizará utilizando las listas de corrección, tablas de búsqueda y los modelos HMM disponibles en aLink: Herramienta de Fusión de Ficheros
Paso 2: Validación de la normalización (Obligatorio)
Comprobar cómo de bueno ha sido el proceso de normalización analizando la variable "validacion" del fichero normalizado.
Los registros con valor 1 en la variable "validacion" se extraen a una hoja de cálculo nueva y se guardan como un fichero ".csv" con elementos separados por ";". Además, se eliminan del fichero normalizado y si procede se actualizan las listas de corrección y tablas de búsqueda.
Los Pasos 3 a 6 deberán realizarse únicamente en caso de que el número de registros que quede sin normalizar sea muy elevado.
Paso 3: Selección de la muestra (Opcional)
Herramientas → HMM: Selección de la muestra
Tomar una muestra del fichero con registros mal normalizados utilizando las listas de corrección y tablas de búsqueda correspondientes
Paso 4: Asignación manual de estados (Opcional)
En el fichero obtenido en el Paso 3 se asignan manualmente los estados asociados a la secuencia o patrón de datos
Paso 5: Entrenamiento de la muestra (Opcional)
Herramientas → HMM: Entrenamiento de la muestra
A partir del fichero de registros mal normalizados obtenido en el Paso 2 se entrena la muestra, es decir, se genera el Modelo Oculto de Markov (HMM) que se usará para normalizar dicho fichero
Paso 6: Normalización (Opcional)
Normalización del fichero con registros mal normalizados usando las listas de corrección, las tablas de búsqueda y el modelo HMM generado en el Paso 5. Tras obtener el fichero normalizado ir a Paso 2 y repetir los pasos nuevamente hasta tener todo el fichero normalizado
Paso 0: Enlace. Indexar los ficheros a enlazar (Obligatorio)
Herramientas → Insertar índices
Solo se realiza una vez en el proceso de enlace. Al fichero pequeño se la asigna __rec_id_a__ y al grande __rec_id_b__
Paso 1: Enlace. Ficheros de entrada (Obligatorio)
Fichero A: será el fichero obtenido en el Paso 0 de enlace con índice __rec_id_a__
Fichero B: será el fichero obtenido en el Paso 0 de enlace con índice __rec_id_b__
Paso 2: Enlace. Análisis explotario (Opcional)
Resumen estadístico de los ficheros
Paso3: Enlace. Agrupación (Obligatorio)
Se toman las variables (o combinación de ellas) por las que se agruparán los ficheros de datos
Paso 4: Enlace. Comparación (Obligatorio)
Se eligen las variables a comparar y la función de comparación usada para ello
Paso 5: Enlace. Clasificación (Obligatorio)
Se utilizará el clasificador de Fellegi Sunter
Paso 6: Enlace. Visualización de resultados (Obligatorio)
A través del histograma y del diagrama de sectores se visualizan los resultados y se estudia la conveniencia de repetir el proceso
Paso 7: Revisión manual del fichero de enlaces y posibles enlaces (Obligatorio)
Con el objetivo de comprobar que no haya enlaces duplicados y que los enlaces realmente lo sean.
Paso 8: Enlace. Añadir campos de los ficheros originales al de enlaces (Obligatorio)
Herramientas → Incluir campos a enlaces
Con el fin de que el fichero de enlaces contenga la información que se desee de los ficheros unidos.
Paso 9: Enlace. Eliminar registros enlazados de los ficheros originales (Obligatorio)
Herramientas → Eliminar registros enlazados
Para iniciar un nuevo proceso de enlace

3 .Código

3.1 .Tratamiento Previo

Librerías externas:

Nombre	Descripción
unicodedata	Estandarizador a UNICODE
json	Librería para utilizar JSON (JavaScript Object Notation) es un archivo que se utiliza principalmente para almacenar y transferir datos principalmente entre un servidor y una aplicación web. Se utiliza popularmente para representar datos estructurados.

Liberías internas:

Carpeta	Nombre Liberia	Descripción
Tratamiento.importers	CSVImporter.py, DBFImporter.py, MDBImporter.py, MySQLImporter.py, ODSImporter.py, ODSImporterr.py, OracleImporter.py, PlainImporter.py, PostgreSQLImporter.py, TABImporter.py, XLSImporter.py, base_importer.py	Librerías para facilitar la importación de diferentes ficheros según su tipo
Tratamiento	Tratprevio.py	Toma los datos de un fichero de entrada el cual puede tener varios formatos, y genera un csv con dichos elementos separados por el separador “;”. Este módulo/librería también transforma el fichero de entrada en otro con codificación UTF-8.

3.2 .Herramienta de Normalización

Librerías externas:

Nombre	Descripción
UniversalDetector	Es un paquete de Python que le permite crear modelos de visión artificial y detección de objetos completamente funcionales con solo 5 líneas de código. La inferencia en imágenes fijas y videos, la transferencia de aprendizaje en conjuntos de datos personalizados y la serialización de modelos a archivos son solo algunas de las características de Detecto. Detecto también está construido sobre PyTorch, lo que permite una fácil transferencia de modelos entre las dos bibliotecas

Librerías internas:

Carpeta	Nombre Liberia	Descripción
Raíz	HMM_etiquetado.py	Librería que permite la selección y etiquetado de las componentes nombres de persona o direcciones postales de los registros seleccionados aleatoriamente de un fichero de datos o fichero de entrenamiento. Estos registros, serán posteriormente usados para construir un modelo oculto de Markov (HMM) usando el interfaz de entrenamiento.
Raíz	HMM_entrenamiento.py	Esta librería sirve para crear automaticamente un Modelo Oculto de Markov (HMM) mediante el interfaz de entrenamiento, de tal forma que usara para ello el conjunto de registros etiquetados que previamente hemos seleccionado en la interfaz de etiquetado.
Raíz	Estandarizador.py	Esta librería sirve tanto para la normalización de direcciones postales, como de nombres de personas e identificadores de personas fisicas y/o juridicas (NIF o DNI), para tener bien codificada y estructurada la información de partida. Si el usuario decidiera a posteriori, realizar con esta información normalizada un proceso de enlace.
Raíz	Standardisation.py	Este módulo proporciona clases para la limpieza de registros y estandarizaciones, ya sea basado en reglas o en un enfoque de aprendizaje automático (modelos ocultos de Markov)
Raíz	simplehmm	Rutinas para la funcionalidad que genera el modelo oculto de Markov (HMM) simple.
Raíz	lookup	Este módulo contiene clases para tablas de búsqueda y listas de corrección.

3.3 .Herramienta de Enlaces

Librerías externas:

Nombre	Descripción
matplotlib	Es una biblioteca para la generación de gráficos en dos dimensiones, a partir de datos contenidos en listas o arrays en el lenguaje de programación Python.
collections	El módulo collections es una parte potente de la biblioteca estándar de Python que le permite trabajar con datos de forma concisa y eficiente.
json	Librería para utilizar JSON (JavaScript Object Notation) es un archivo que se utiliza principalmente para almacenar y transferir datos principalmente entre un servidor y una aplicación web. Se utiliza popularmente para representar datos estructurados.

Librerías internas:

Carpeta	Nombre Liberia	Descripción
Raíz	Enlazador.py	Esta librería se encarga de enlazar los diferentes archivos introducidos.
Raíz	Classification.py	Este módulo proporciona varios clasificadores, tanto basados en supervisados como en métodos no supervisados, que clasifican los vectores de peso en «coincidencias» y 'no coincidentes', y posiblemente 'posibles coincidencias' (algunos clasificadores no lo hacen) clasificar los vectores de peso en esta tercera clase).
Raíz	Comparison.py	Este módulo proporciona clases para comparar registros y seleccionar campos que puedan ser utilizados en el proceso de enlace.
Raíz	Indexing.py	Este módulo proporciona clases para generar bloques o grupos e índices que se utilizan en el proceso de enlace para reducir el número de comparaciones a realizar y llevar a cabo una comparación de pares de registros mas eficiente.
Raíz	output.py	Este módulo proporciona varias funciones que permiten guardar los resultados del proceso de enlace o de búsqueda de duplicados en archivos de varios formatos También contiene varias funciones auxiliares relacionadas con la salida (y la entrada).
Raíz	em.py	Este modulo contiene el código Phyton del algoritmo de Expectation-Maximization para encontrar de manera no supervisada estimadores de máxima verosimilitud para los parametros del modelo de Fellegi y Sunter.
Fusionadores	Fusionador_Enlaces.py	Fusionador por campo común con interfaz gráfica. Permite añadir campos de los ficheros A y B a los ficheros de enlaces, no enlaces y posibles enlaces generados en el proceso de enlace. Esta tarea se realiza

Carpeta	Nombre Liberia	Descripción
		a través de los campos índices (rec_id1 y rec_id2) insertados inicialmente en los ficheros A y B que se quieren enlazar
Fusionadores	Fusionador_NO_Enlaces.py	Fusionador por campo común con interfaz gráfica. Permite eliminar del fichero A aquellos registros que han sido considerados como enlaces en un primer proceso de enlace y reindexa el nuevo fichero A de partida con el resto de registros que se han considerado como no enlaces. Esta tarea se realiza a través del campo índice rec_id1 insertado inicialmente en el fichero A
Fusionadores	Funcion_Fusionador_Enlaces.py	Contiene el código Python que realiza el proceso de incluir campos a los ficheros de enlaces, no enlaces y posibles enlaces y escribirlos en csv.
Exporters	csv_oracle_export.py csv_postgresql_export.py oracle_export.py postgresql_export.py	Permiten exportar los ficheros csv de salida tras los procesos de enlace a una base de datos PostgreSQL u Oracle.

3.4 .Uso en las tres áreas.

Librerías externas:

Nombre	Descripción
Gi Gtk GdkPixbuf	Se encargan de la interfaz gráfica
csv	Para manejar las plantillas guardadas de los importadores de origen
Os subprocess sys	Librerías básicas de sistema
Math.ceil	Función matemática del ceil
logging	Con el módulo logging se dispone de una solución más elegante para analizar errores que, además, ahorra mucho trabajo. El logging de Python presenta cinco niveles de gravedad distintos,

Nombre	Descripción
	que en inglés reciben el nombre de “levels of severity”. Si se desea crear un propio filtro de registro, obviamente se puede hacer, aunque los niveles de gravedad incluidos en el módulo logging de Python son: debug, info, warning, error, critical.
random	La librería random es también proveída por Python. Ofrece generadores de números pseudo-aleatorios para varias distribuciones.
types	Este módulo define funciones de utilidad para ayudar en la creación dinámica de tipos nuevos.
numpy	Es una librería para el lenguaje de programación Python que da soporte para crear vectores y matrices grandes multidimensionales, junto con una gran colección de funciones matemáticas de alto nivel para operar con ellas.
time	El módulo Time contiene funcionalidades que nos permiten entre otras cosas, manipular y dar formato a fechas y horas, obtener fechas actuales, rangos y hacer cálculos con estas.

Librerías internas:

Carpeta	Nombre Liberia	Descripción
Monitor Negro	logger.py	Paquete de registro para Python, en el que muestra los diferentes mensajes informativos
Raíz	utils.py	Serie de utilidades para el buen funcionamiento de la aplicación, como controlar la codificación de los ficheros o transformar todo a minúsculas para garantizar mas homogeneidad
Raíz	Auxiliary.py	Librería con pequeñas funciones para checkear strings y numbers
Raíz	dataset.py	Este módulo proporciona clases para acceder a diferentes tipos de conjuntos de datos, incluyendo archivos de texto (valores separados por coma y columnas), bases de datos (usando la API de base de datos de Python), archivos binarios (usando estantes de Python) y datos basados en memoria (usando diccionarios Python).
Raíz	Mymath.py	Varias rutinas matemáticas.
Raíz	storage.py	Librería de tablas para base de datos.
Raíz	constants.py	Librería de constantes

4 .Montaje, compilación y despliegue de la infraestructura para su mantenimiento

En dicho punto, se explicaría como montar y compilar la aplicación para desarrolladores, los cuales deberán realizar cambios en codigo para su mantenimiento.

4.1 .Descargar del repositorio los recursos necesarios

Tras la descarga de los recursos disponibles para su montaje, que podemos encontrar actualmente en la la opción de ***Windows: sólo código fuente***, disponible en el siguiente enlace:

https://www.juntadeandalucia.es/institutodeestadisticaycartografia/dega/herramienta-de-fusion-de-ficheros-alink

4.2 .Descargar y configuración de Anaconda

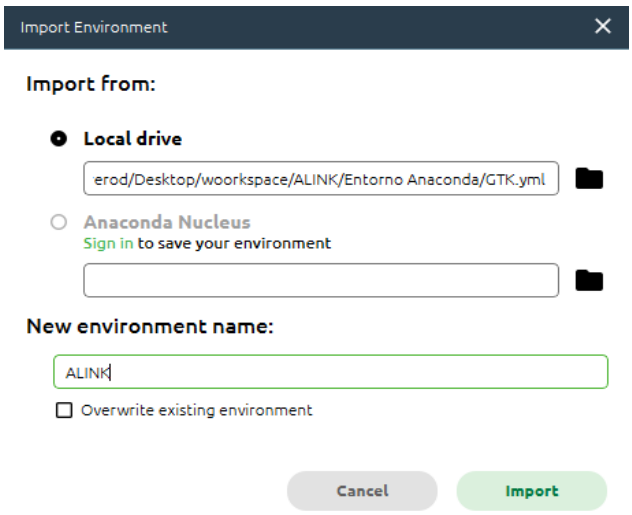
Para ejecutar la aplicación desde el código fuente, debemos descargar e instalar ***Anaconda*** en nuestro ordenador (disponible para todas las plataformas).

Una vez instalado, disponemos de dos vias para crear el entorno de Python que vamos a precisar:

1. Creando un entorno nuevo con todas las dependencias del proyecto, en caso de querer añadir alguna dependencia más, como puede ser pyinstaller para la creación de archivos .exe
2. Importarlo desde el fichero ***GTK.yml*** que se proporciona dentro de la aplicación en la carpeta Entorno Anaconda, ubicada en la carpeta alink-codigo_fuente> alink-develop descargada en el paso anterior.

Para importarlo, nos desplazaremos a la pestaña Environments de la izquierda. Y pulsamos en la opción *Import*.

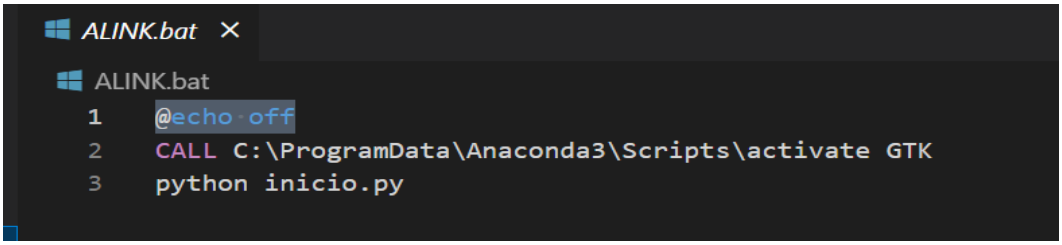
Navegamos hacia el archivo, nombramos al entorno como queramos y esperamos a que se instale.



Para actualizar en un futuro los paquetes del entorno seleccionamos el filtro Updatable y seleccionamos los que deseemos.

Para el montaje de la aplicación se han tenido que realizar algunos cambios en el fichero ALINK.bat que se pasa ahora a desglosar con la finalidad de que la persona que lea este documento y los siguientes apartados, sea capaz de desplegar el código en un entorno local sin la complicación extra del desconocimiento de estas vicisitudes. Se han seguido los siguientes pasos:

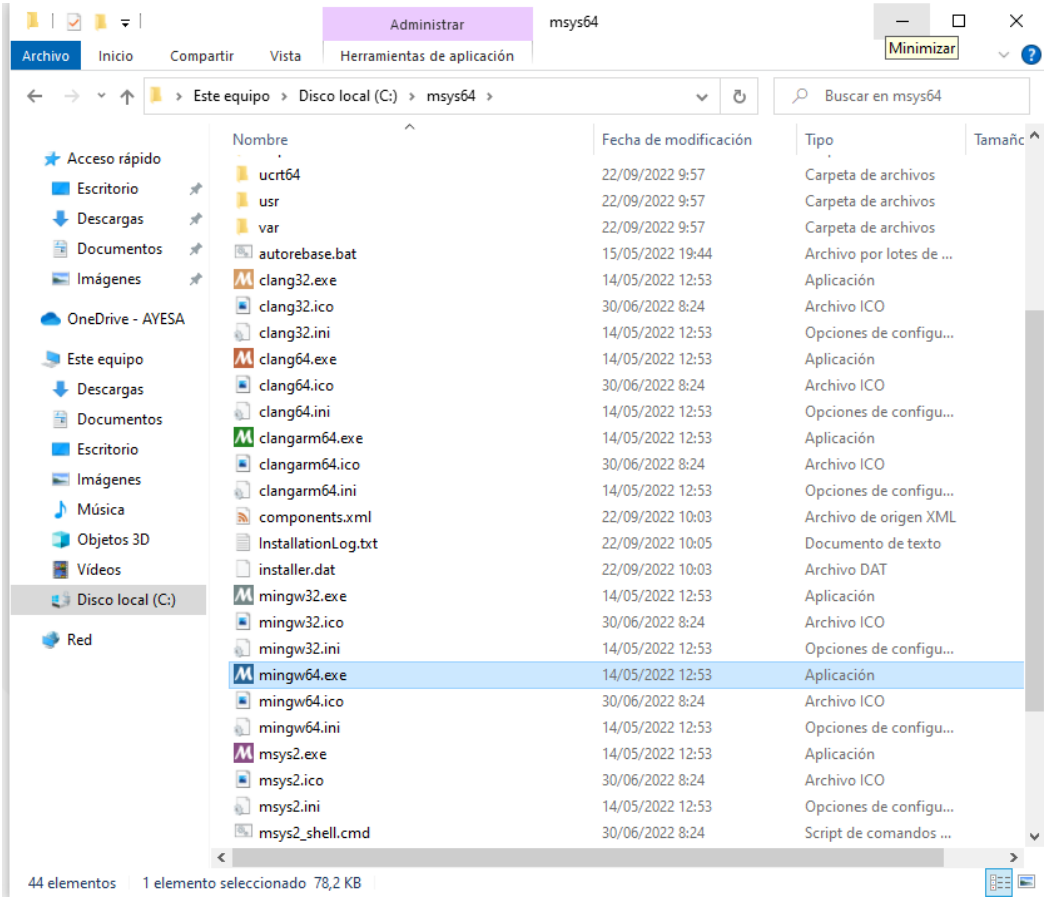
1. Descargar el código fuente de la aplicación. (La carpeta comprimida *alink*-develop del GitLab)
2. Cambiar la llamada del archivo ALINK.bat.
3. Deberá buscar la carpeta Scripts, usualmente creada en la carpeta Anaconda3 de su usuario, dirección semejante a esta: C:\Users\sherrerod\Anaconda3\Scripts.
4. Y ponerlo en la llamada delante de activate GTK, como queda en la siguiente imagen:



4.3 .Descarga de Glade

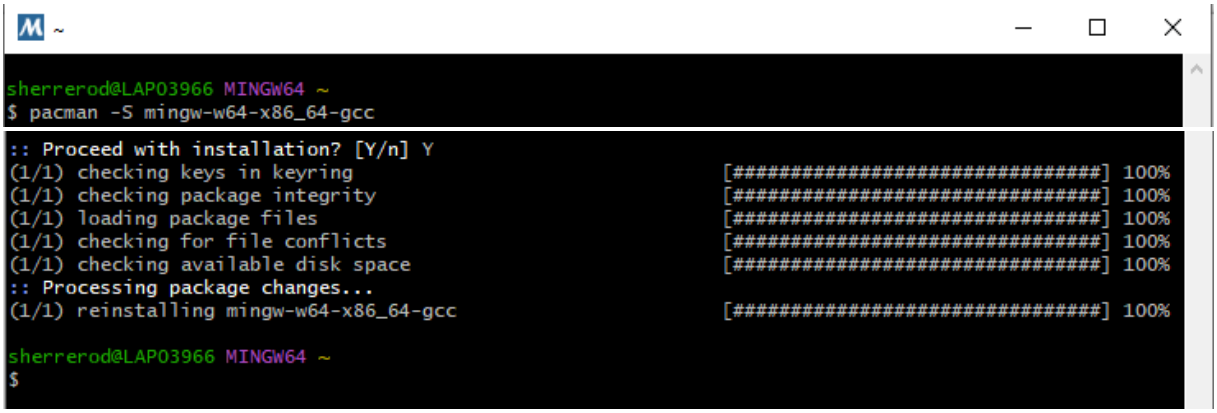
Para los archivos .glade, para la realización de interfaces gráficas debemos descargar Glade a partir de la consola de comandos de MSYS2.

Una vez instalado MYS2 debemos buscar la consola que se encuentra entre los archivos descargados (archivo mingw64.exe).

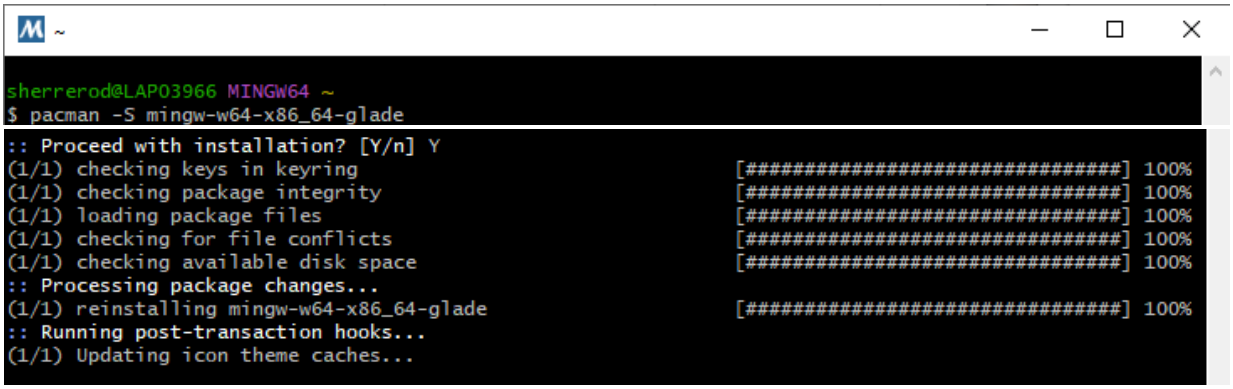


Cuando nos aparezca la consola de *mingw64* deberemos introducir los siguientes comandos

Para compilar: *pacman -S mingw-w64-x86_64-gcc*



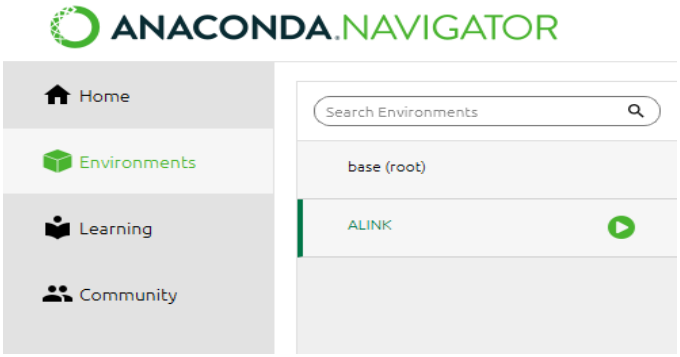
Para instalar Glade: *pacman -S mingw-w64-x86_64-glade*



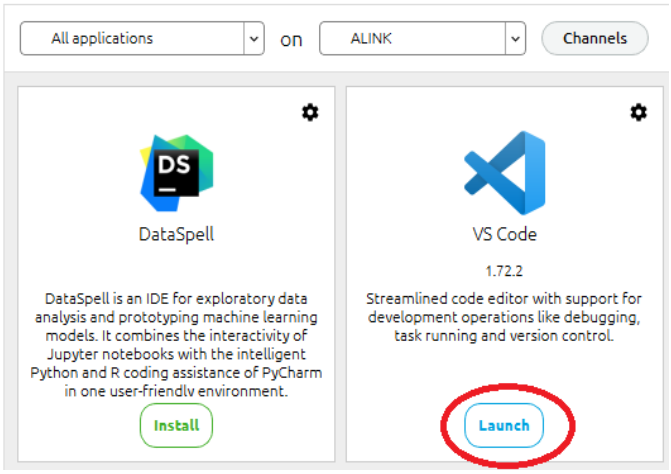
Ahora para abrir Glade solo tendremos que escribir en la consola Glade

4.4 .Montaje

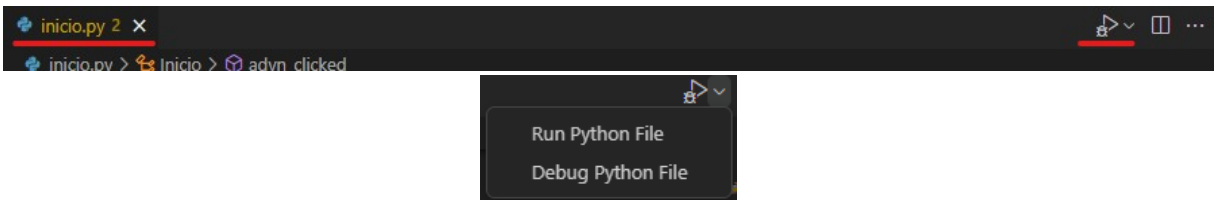
Una vez tenemos todo preparado, debemos *ejecutar* el Environment de Anaconda que hemos creado anteriormente y nos quedara la pantalla así:



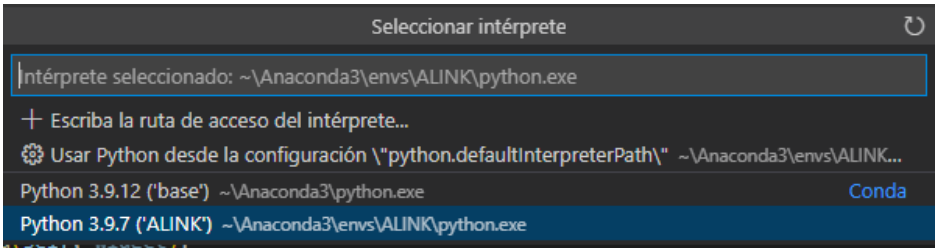
Una vez ejecutado, lanzaremos VS Code desde Anaconda para que se nos enlace con el Enviroment y poder tener el entorno donde realizaremos los cambios de código y ejecutaremos para la realización de las pruebas necesarias de nuevo código. Vs Code debemos instalarlo desde Anaconda.



Una vez estemos en VS Code y hayamos importado el proyecto, deberemos abrir el archivo inicio.py y *ejecutar* desde el botón que se encuentra en la parte superior derecha de la pantalla.



En caso de que nos pida seleccionar el interprete, deberemos seleccionar el creado en el Enviroment de Anaconda.



Si todo ha ido bien, se nos abrirá la pantalla principal de *aLink*:

5 .Montaje para usuarios

En dicho punto, se explicaría como *ejecutar* la aplicación por un usuario que descargue la aplicación por primera vez para su uso y no para su modificación

5.1 .Encontrar el archivo ejecutor y ejecutarlo

Deberemos descargar la carpeta ALINK.zip y descomprimirla, dicho archivo se encontrará en la siguiente URL (Actualmente no se encuentra dicho archivo):

https://www.juntadeandalucia.es/institutodeestadisticaycartografia/dega/herramienta-de-fusion-de-ficheros-alink

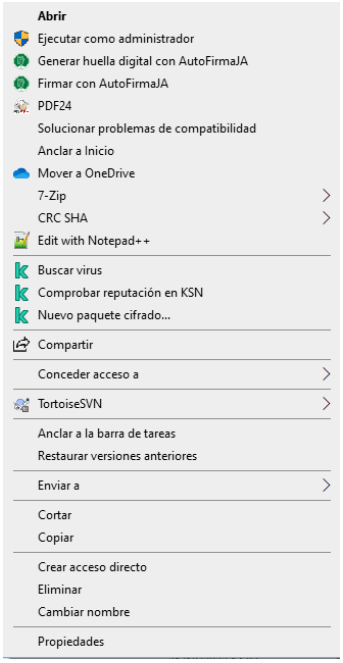
Al abrir el archivo ALINK que encontramos en el ALINK.zip, el cual sera creado por los desarrolladores para la ejecución de la aplicación *aLink* nos deberá aparecer algo así:

Nombre	Fecha de modificación	Tipo	Tamaño
brotli	31/10/2022 14:02	Carpeta de archivos	
cairo	31/10/2022 14:02	Carpeta de archivos	
certifi	31/10/2022 14:03	Carpeta de archivos	
editor	31/10/2022 14:03	Carpeta de archivos	
gi	31/10/2022 14:02	Carpeta de archivos	
gi_typelibs	31/10/2022 14:03	Carpeta de archivos	
greenlet	31/10/2022 14:02	Carpeta de archivos	
gui	31/10/2022 14:03	Carpeta de archivos	
lib	31/10/2022 14:02	Carpeta de archivos	
lxml	31/10/2022 14:03	Carpeta de archivos	
matplotlib	31/10/2022 14:03	Carpeta de archivos	
MySQLdb	31/10/2022 14:02	Carpeta de archivos	
numexpr	31/10/2022 14:02	Carpeta de archivos	
numpy	31/10/2022 14:02	Carpeta de archivos	
pandas	31/10/2022 14:02	Carpeta de archivos	
PIL	31/10/2022 14:02	Carpeta de archivos	
psycopg2	31/10/2022 14:02	Carpeta de archivos	
PyQt5	31/10/2022 14:02	Carpeta de archivos	
pytz	31/10/2022 14:03	Carpeta de archivos	
scipy	31/10/2022 14:02	Carpeta de archivos	
setuptools-60.5.0-py3.9.egg-info	31/10/2022 14:03	Carpeta de archivos	
share	31/10/2022 14:03	Carpeta de archivos	

En esta carpeta estarán todos los archivos necesarios para la ejecución del archivo .exe

El archivo que necesitamos para ejecutar *aLink* tiene que ser .exe y tener el icono de *aLink* (Fondo blanco con la palabra *aLink* en verde) como este:

Para la ejecución del archivo .exe, podemos hacer doble clic sobre él o abrirlo con el botón derecho > Abrir, como aparece en las siguientes imágenes.



Automáticamente, se nos deberán abrir dos pantallas el monitor y el inicio de la aplicación.

El inicio de la aplicación:

Ya estaría la aplicación lista para su uso.

6 .ANEXOS

Los términos utilizados en el presente documento, así como el material de soporte utilizado para su elaboración.

6.1 .Glosario de términos

Término	Descripción
VS Code	Es un editor de código fuente que incluye soporte para la depuración, resaltado de síntesis, canalización inteligente de código, fragmentos y caracterización de código. Es gratuito y de código abierto
Glade	Es una herramienta de desarrollo visual de interfaces gráficas mediante GTK / GNOME. Es independiente del lenguaje de programación y predeterminadamente no genera código fuente sino un archivo XML
Anaconda	Es una distribución libre y abierta de los lenguajes Phyton y R utilizada en ciencia de datos, y aprendizaje automático (*machine learning*).Esto incluye procesamiento de grandes volúmenes de información, análisis predictivo y cómputos científicos. Está orientado a simplificar el despliegue y administración de los paquetes de software
csv	son un tipo de fichero en formato abierto sencillo para representar datos en forma de tabla, en las que las columnas se separan por comas (o punto y coma en donde la coma es el separador decimal y las filas por saltos de línea. El formato CSV no está estandarizado

Tabla 1: Glosario

6.2 .Bibliografía y referencias

Bibliografía y referencias
[aLink]MDJIMP_IECA_Manual_Usuario_aLink.pdf
[aLink]guia_rapida_definitiva.pdf

Tabla 2: Bibligrafías y referencias