

INFORMÁTICA ENTOMOLÓGICA (I)

CONSEJOS Y SUGERENCIAS PARA UNA MAYOR COMPATIBILIDAD ENTRE

BASES DE DATOS ENTOMOLÓGICAS

Diego García Gerverno

RESUMEN: Con el presente trabajo, dividido en varias partes que irán apareciendo sucesivamente, no se pretende crear un "perfecto manual" para informáticos, sino dar una idea aproximada de lo que se puede conseguir con los ordenadores personales actuales, al tiempo que se orienta a los recién iniciados hacia el uso del software más adecuado para cada caso. Asimismo se sugiere una estructura de base de datos general pensada para la gestión de datos entomológicos.

La aparición de los ordenadores personales tal y como los conocemos hoy, es decir, con una combinación espectacular de potencia, poco volumen, y precio bastante asequible, ha supuesto un intento por parte de los usuarios finales de adaptarlo a casi cualquier situación en la que se requiera una máquina que procese información a gran velocidad, ahorrándonos tiempo y aburrimiento, síntomas estos que solían atacar la moral de cualquier entomólogo cuando se disponía por enésima vez a ordenar y clasificar debidamente su colección. Todos tenemos en mente la tópica imagen del entomólogo "antiguo": señor de grave compostura que, debido a la gran cantidad de tiempo libre de la que disponía, dedicaba todos sus esfuerzos y energías en capturar cuanto bicho se cruzara en su camino, con muy pocos resultados de cara al avance científico. Esta estampa, clara referencia a la opinión que el vulgo venía teniendo de nuestro gremio, bastante errónea por suerte, ya resulta bien poco corriente entre nuestros colegas. Cualquier entomólogo que se precie, casos psiquiátricos aparte, intenta por todos los medios que su trabajo de recolección e identificación dé algún fruto patente. Hasta ahora esto se conseguía tras varios años de titánico esfuerzo: una vez que se terminaba la recolección había que tomar debida nota de todos los descubrimientos de mayor o menor importancia realizados durante la jornada para recopilarlos más adelante, cuando se intentaba preparar algún trabajo.

Si lo que se intentaba era simplemente comparar datos para detectar nuevas citas y ampliar el conocimiento de la fauna de una zona determinada, el proceso era relativamente sencillo. Pero cuando lo que se pretende es crear una serie de mapas **U.T.M.** de distribución, o gráficos estadísticos de la población, o simplemente listar una serie de categorías taxonómicas teniendo en cuenta alguna condición más o menos compleja, todo el panorama se transforma. Todos comenzamos con cierta alegría el trabajo, pero cuando descubrimos lo arduo del mismo, la moral nos queda a la altura de los talones. Resulta casi imposible elaborar un fichero "clásico" en el que todas los datos aparezcan ordenados de varias maneras para poder disponer prontamente de las citas precisas. Aún así, hacer un listado de ciertas dimensiones a partir de datos en fichas podría resultar traumático para la salud mental de cualquier entomólogo. Y todo esto teniendo en cuenta el tiempo desperdiciado en ello.

El ordenador personal es la herramienta adecuada para hacer todo esto con un coste mínimo en tiempo y minutos del psiquiatra.

Muchos siguen reacios al empleo de estas herramientas por considerarlas excesivamente complicadas o peligrosas para el crecimiento del empleo. Yo, particularmente, creo que mucho más peligrosas resultan algunas Administraciones del Estado, y, en cambio, nadie intenta boicotearlas tal y como se hace con la imagen del ordenador. Otros piensan que estas máquinas sólo pretenden sustituir al hombre y que no comenten más que errores. Por suerte son cada día menos los que opinan ésto. Estoy seguro de que todos habremos escuchado hasta la saciedad que no son los ordenadores los que se equivocan, sino los que los programan. Por último podemos encontrar quien piensa que son artefactos maravillosos capaces de hacer de todo y que tienen un enanito dentro de la caja de metal que es el que piensa por nosotros. Estos son los peores. Cuando, ya tarde normalmente, descubren que los ordenadores son simples tontos rápidos, suelen ser peor que los fumadores arrepentidos, que no paran de decirte lo mucho que les molesta el humo aunque no estés fumando. Debemos verlos como lo que son: herramientas de gran potencia para el manejo de información que pueden hacer nuestro trabajo mucho más agradable y rápido.

A partir de este punto voy a comenzar a hacer referencia a una serie de conceptos que deben ser sobradamente conocidos para cualquier aficionado al campo de la informática. Si no se conociera alguno de ellos, recomiendo consultar algún libro especializado de los muchos que inundan el mercado a precios verdaderamente exorbitados (y luego dicen que no se lee lo suficiente en este país). Aclarar cualquiera de estos conocimientos se sale del propósito de esta serie de artículos y continuar adelante sin entender el significado de lo que es un **sistema operativo**, **hardware**, **base de datos**, etc. puede resultar peligroso para la meninge de los que lo intenten. De todas formas, cualquiera que haya llegado leyendo hasta aquí tiene que ser porque sienta un desprecio absoluto por sus neuronas o porque se encuentre sentado en el cuarto de baño y se le haya agotado el papel higiénico; así que, tú mismo.

Un grave problema que tiene la informática actualmente es que el hardware va muy por

delante del software. Visto con un ejemplo práctico, esto quiere decir que, mientras ya existen micros **486** o incluso **Pentium** capaces de trabajar en multitarea, o memorias de varios megas de capacidad a bajo costo, el sistema operativo **MS/DOS** no puede direccionar más allá de un mega por sí solo y se comporta de cara al micro tratándolo como si de un **8086** se tratara. Es como si nos empeñáramos en usar un **arado romano** arrastrado por un **John Deere** de último modelo para labrar un campo. Esta situación se ha solventado en parte en los últimos tiempos gracias a la aparición de algún que otro programa, como por ejemplo el **Windows**, que crea un entorno de trabajo muy particular, que nos permite aprovechar al máximo las características de cada ordenador, o de sistemas operativos mucho más potentes de cara al hardware que el popular (e insulso) **MS/DOS**. Entrando ya definitivamente en materia, todo esto nos viene a demostrar que ciertos modos de trabajar con bases de datos resultan por completo obsoletos. Vayamos por partes. Si hace unos pocos años (dos o tres a lo sumo), un entomólogo se decidía a crear una base de datos para gestionar su colección y me preguntaba que programa gestor de bases de datos utilizar, yo aconsejaba sin dudar el conocido **dBase**. Las necesidades de un entomólogo son extraordinariamente reducidas, desde el punto de vista de programación de bases de datos, por supuesto. El número de registros que puede alcanzar una base de este tipo no es muy elevado, sólo de algunos millones, y la relaciones entre distintas bases son muy sencillas, así como sus índices. Resulta, pues, desde cualquier punto de vista, un tremendo despilfarro utilizar gestores capaces de indexar varios millones de registros en un segundo, o de interaccionar con el disyuntor de furcios megaloplasmático de la cadena del water. Viene a ser como colocar una pantalla de cine de seis por ocho metros en el comedor de tu casa para ver la televisión. El **dBase** se ha extendido masivamente desde sus inicios sobre todo por una razón muy concreta, que será el pie del resto de trabajos sobre el tema que iré preparando: porque creó un estandar universalmente aceptado al desarrollar los archivos de extensión **.DBF** y porque el lenguaje que los gestiona resulta ser tremendamente flexible, de muy fácil manejo y rápido de aprender.

De hecho, a partir de la versión tres del **dBase** comenzaron a aparecer a su sombra una serie de lenguajes muy parecidos que manejaban ficheros **.DBF** y que, en sus inicios, resultaban casi completamente compatibles con el propio **dBase**. A todos estos lenguajes se les denomina con el apelativo común de lenguajes **XBase** y se procura mantener una serie de estándares que no los haga alejarse mucho de su predecesor. Entre ellos destaca sin duda alguna, el compilador **Clipper**.

Cuando alguien encargaba una aplicación escrita en **dBase**, se veía obligado a comprar el propio **dBase**, ya que, al tratarse de un lenguaje interpretado (por lo menos en sus primeras versiones) era necesario arrancarlo para ejecutar la aplicación. En este estado de cosas llegó **Clipper**. Se trata de un lenguaje completamente compatible

con el estandar **XBase**, que compila y luego enlaza las aplicaciones que programemos. Esto quiere decir que el resultado final será un archivo de extensión **.EXE** totalmente independiente del **dBase**, y, lo más importante, que manejará archivos de base de datos con extensión **.DBF**^{nota 1}. En el único punto donde se aparta un poco del **dBase** es en sus índices, que tienen una extensión diferente (**.NDX**) y que son mucho más rápidos.

Apliquemos todo lo que hemos dicho a un caso práctico: mi amigo y colega Daniel Grustán (ampliamente conocido por haber llegado a un estado de casi perfecto equilibrio entre sus actividades entomológicas y maritales) me ha encargado una aplicación que le permita realizar el mapeado de sus capturas automáticamente (ilusó él). El hecho de que haya que utilizar posibilidades más o menos avanzadas (como pueda ser presentar en pantalla un mapa en modo gráfico), hace que el **dBase** sencillamente no nos sirva. Y aunque **Clipper** tampoco tenga posibilidades gráficas, resulta que se le pueden añadir librerías con funciones que se las darían. **Clipper** está escrito en **C**; por lo tanto, se puede desarrollar una función en **C**, como pueda ser inicializar el modo gráfico, que se pueda llamar desde **Clipper** y que nos permita alcanzar nuestros objetivos. Una vez elegido el lenguaje de programación, debemos desarrollar la estructura de la base de datos que soporte la información a manipular. Y aquí sí que resulta casi imprescindible el uso de **dBase**, ya que nos facilitará la tarea de crear el archivo **.DBF**, mucho más complicada si sólo utilizáramos **Clipper**.

La estructura de cualquier base de datos, está compuesta por una serie de registros divididos en campos. Estos campos los agruparemos en varios **bloques**^{nota 2} según la información que contengan; estos bloques no se reflejarán en ninguna parte de la base de datos, pero nos darán una idea de lo que queremos realmente. Los campos a utilizar los elegiremos luego. En primer lugar, como no, la identificación de la especie. Si usáramos varios campos para hacer esto, nuestro trabajo sería extremadamente largo, árduo, pesado, y hasta peligroso para nuestra médula espinal. Y si no, haced la prueba: teclead en vuestra base de datos unas trescientas veinticuatro veces el nombre *Palaechrysophanus hippothoe*. Cuando vayais por la ciento cuarenta y nueve, ya me contareis. Lo que habría que hacer es codificar cada especie con su correspondiente **familia**, **superfamilia** y **orden**. En nuestro caso podría quedar como se ilustra en la figura 1. Con ello conseguiremos:

- 1^a **Teclear** mucho menos
- 2^a **Localizar rápidamente** los ejemplares de la misma familia, tribu, orden, etc
- 3^a Ser **elegantes**
- 4^a **Minimizar neuronas**

Todo esto implica de por sí la existencia de al menos dos relaciones distintas: la primera con la nombres de (por ejemplo) las especies clasificadas hasta la fecha en Aragón (nuestro caso), y la segunda con los nombres en clave de cada una de las citas. Esta forma de trabajar sugiere el empleo de una aplicación específica, ya que con un

manejador de bases de datos simple, como el dBase, sería peor el remedio que la enfermedad; ¡a ver quien es el listo que se aprende los nombres científicos de cada especie junto con su clave de codificación correcta!

En un segundo bloque de datos atacaríamos de lleno la localización de cada cita. Nuevamente nos dará más potencia una relación que nos indique mediante claves cada una de los lugares visitados. Por ejemplo, yo utilizo un sistema muy simple: asignar a la primera excursión del año el número uno, a la segunda la dos, etc. junto con las dos últimas cifras del año. El quinto lugar visitado del año 89 sería "05/89", que en mi libro de registro me indica que se trata de Gratal, el 1 de Mayo de 1989, en el término municipal de Arguís, a 1400 mts. de altura, de coordenadas U.T.M. 30TYM0985, que figura en el mapa 284 del Servicio Geográfico del Ejército, y que yo llegue muy cansado después de una larga ascensión. Toda esta información en ¡5 caracteres!... (excepto lo de mi cansancio, que todavía lo guardo en un par de conexiones neuronales que me sobran...)

Y, por último, el inevitable campo de "observaciones". Aquí se guardarían los inefables comentarios acerca de si el bicho "descansaba apaciblemente sobre una linda florecilla silvestre" o el inescrutible y, por desgracia, típico, "teratología extraña parecida a un pennis hiperdimensionado sobre la zona antero-posterior del hemielitro izquierdo, bajo mancha costeroidea del artejo subnumerario veinticinco". Yo, particularmente, suelo más bien tomar notas del tipo "hallada extensa colonia bien representada junto a imponente maciza de origen nórdico bañándose en top-less a escasos 100 metros, y sin síntomas aparentes de malformaciones ginecológicas", que me han costado más de un pescozón por parte de mi novia, especialmente cuando tomo fotografías.

Mucha gente preguntará: ¿y los datos de número de ejemplares, sexo, etc? Bien, pues la solución consiste o bien en crear pequeños campos (y, además, muy pocos campos) para estos y otros datos de corta extensión, o bien incluirlos de algún modo en el campo de identificación. Cualquiera de los dos métodos sirve. Tened en cuenta una cosa fundamental en el mundo de la informática: no es mejor el más sabe de ordenadores ni el que tiene el mejor equipo, sino el que al iniciar un trabajo sabe qué resultados quiere conseguir, escoge la herramientas (programas), adecuados y resulta claro y conciso en sus métodos. Aquí está la clave de un buen programa: concisión y claridad. Las estructuras más simples, (con pocas ramas), y muy generales, (que sirvan para muchas cosas), son las que mejor resultado dan.

Bueno, aquí termino mi exposición. Acompaño una serie de figuras que ilustran la estructura de la base de datos que propongo, y que me servirá de punto de referencia a la serie de trabajos que seguirán a éste. Si alguien ha conseguido llegar hasta aquí... ¡enhorabuena!, prometo ser menos aburrido en sucesivas ocasiones.

NOTAS:

1.- Existen otros lenguajes que no son dBase, pero que aceptan los ficheros formato dBase

2.- En la práctica: relaciones separadas, auxiliares de la principal

AVISO: Todas las marcas mencionadas en el presente trabajo son marcas registradas por sus respectivos propietarios, incluido el Palaeochrysothamus hippothoe y el arado romano. El autor se responsabiliza de las burradas plasmadas en este artículo, pero no quiere saber nada de los efectos psicológicos que pueda causar en aquellos que lo lean. Por cierto, pulsa con el alfiler.

Ejemplo práctico de codificación de una identificación de cara al manejo de una base de datos:

LEP : PP / A . 1 . 1

NOMBRES SUPRAGENÉRICOS:

ORDEN: **LEP -> LEPIDÓPTERA**

FAMILIA: **PP -> PAPILIONIDAE**

NOMBRES ESPECÍFICOS:

GÉNERO: **A -> PAPILIO**

ESPECIE: **1 -> MACHAON**

SUBESPECIE: **1 -> HISPANICUS**

figura nº1

**PROPUESTA DE ESTRUCTURA DE BASE DE DATOS ENTOMOLÓGICA
DE PROPOSITO GENERAL (SIMPLIFICADA)**

BASE DE DATOS:

INSECTA

✓RELACION n° 1

CITAS

		NOMBRE _____	TIPO _____
ESTRUCTURA:	CAMPO 1:	CITA	CHARACTER
	CAMPO 2:	LOCALIZACIÓN	CHARACTER
	CAMPO 3:	OBSERVACIONES	MEMO

✓RELACIÓN n° 2

IDENTIFICACIÓN

		NOMBRE _____	TIPO _____
ESTRUCTURA:	CAMPO 1:	CITA	CHARACTER
	CAMPO 2:	ORDEN	CHARACTER
	CAMPO 3:	FAMILIA	CHARACTER
	CAMPO 4:	GENERO	CHARACTER
	CAMPO 5:	ESPECIE	CHARACTER

✓RELACIÓN n° 3

LOCALIZACIONES

		NOMBRE _____	TIPO _____
ESTRUCTURA:	CAMPO 1:	LOCALIZACIÓN	CHARACTER
	CAMPO 2:	LUGAR	CHARACTER
	CAMPO 3:	FECHA	FECHA
	CAMPO 4:	ALTITUD	NUMÉRICO
	CAMPO 5:	COORDENADAS UTM	CHARACTER
	CAMPO 6:	OBSERVACIONES	MEMO

