

00323

44



UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO

FACULTAD DE CIENCIAS

UNA BASE DE DATOS DE CALIBRADORES PARA
MEXART

TESIS CON
FALLA DE ORIGEN

T E S I S
QUE PARA OBTENER EL TÍTULO DE:
F I S I C O
P R E S E N T A:
MAX ULISES DE MENDIZÁBAL CARRILLO

DIRECTOR DE TESIS:
DR. STANLEY EUGENE KURTZ SMITH





Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.



INSTITUTO NACIONAL
DE ESTADÍSTICAS
Y CENSO

DRA. MARÍA DE LOURDES ESTEVA PERALTA
Jefa de la División de Estudios Profesionales de la
Facultad de Ciencias
Presente

Comunicamos a Usted que hemos revisado el trabajo escrito:
Una base de datos de calibradores para MEXART

realizado por De Mendizábal Carrillo Max Ulises con número de cuenta 85526994
quién cubrió los créditos de la carrera de Física.
Dicho trabajo cuenta con nuestro voto aprobatorio.

Atentamente

Director de Tesis
Propietario

Propietario Dr. Stanley Eugene Kurtz Smith

Stanley Eugene Kurtz Smith

Propietario Dr. Américo González Esparza

[Firma]

Propietario Dr. Darío Moreno Osorio

[Firma]

Suplente Dr. Pedro Bosh Giral

[Firma]

Suplente M. en C. Patricia Raquel Avilés Mandujano

[Firma]

Consejo Departamental de Física

[Firma]

DRA. PATRICIA GOLDSTEIN MENACHE
Coordinadora de Licenciatura

Índice general

1. La creación de la base de datos	11
1.1. Identificación de las fuentes de información	11
1.2. Definición del formato de la base de datos	12
1.2.1. El gestor de la base de datos	13
1.2.2. Implementación de la base de datos	14
1.3. La conversión de los datos originales	14
1.4. La conversión B1950.0 a J2000.0	17
2. Las herramientas de consulta	21
2.1. Uso de la información	21
2.2. Los datos de entrada	22
2.3. Consideraciones de programación	23
2.4. La consulta a la base de datos	25
2.5. Los resultados de la consulta	27
2.6. La precesión desde el año 2000 hasta el día de hoy	27
3. Conclusiones	29
A. Cálculo de precesión por el método matricial	31

B. Formato del archivo astroflux-db.sql	33
C. genera-db.pl	35
D. muestra-fuente.pl	45
E. busca-fuentes.pl	47

Agradecimientos

Este trabajo no hubiera sido posible sin la comprensión, paciencia, apoyo y amor de Concha. También debo agradecer a mis hijas Conchita y Carmen por la multitud de tardes, fines de semana y vacaciones que no pude pasar con ellas para dedicárselas a la escuela o al trabajo.

Debo mucha gratitud a mi tía Gloria que me apoyó en los momentos más difíciles de mi vida, a Conchita, mi suegra, a Pepi y a Conchi, mi familia adoptiva de Murcia, que me ayudaron a revivir la inquietud de retomar una carrera que, de otro modo, hubiera quedado trunca.

Sin mis amigos, compañeros y profesores de la Facultad de Ciencias, hubiera sido imposible no sólo este trabajo, sino la misma supervivencia material. Afortunadamente son tantos que hacen muy difícil agradecer a cada uno de ellos de forma individual su generosidad, afecto y apoyo desinteresado.



303 217
LIVRO 3 1913

Introducción

MEXART[7], o *Mexican Array Radio Telescope*, por sus siglas en inglés, es un radio telescopio que se encuentra ubicado en Coeneo, en el estado de Michoacán. Consta de un arreglo plano de 64×64 (4,096) dipolos de onda completa, con una frecuencia de operación de 140 MHz y ocupa una superficie de aproximadamente 10,000 m². Está diseñado para hacer observaciones de centelleo interplanetario, y proporcionará información para el estudio de las perturbaciones solares de gran escala que se propagan en el medio interplanetario.

El estudio de estas perturbaciones en el "viento solar" proporciona información muy valiosa para entender la dinámica del medio interplanetario y la interacción de dichas perturbaciones con el campo magnético de la Tierra. La predicción del clima espacial se ha convertido en asunto importante ya que las tormentas geomagnéticas, provocadas por las eyecciones de masa coronal, pueden causar problemas en las telecomunicaciones, dañar a los sistemas de transmisión de potencia, e incluso dañar irreversiblemente los componentes electrónicos de los satélites artificiales. Estas tormentas pueden provocar daños a seres humanos, como por ejemplo a los astronautas en la Estación Espacial, y en varias especies de animales terrestres que usan el campo magnético de la Tierra para orientarse. En particular, los efectos de las tormentas sobre los satélites de telecomunicación,

pueden tener un gran impacto económico. La habilidad de pronosticar el tiempo espacial es sumamente importante para tomar las medidas adecuadas que permitan minimizar el impacto de las tormentas espaciales y por lo tanto su efecto económico negativo.

El MEXART formará parte de una red de observatorios de centelleo interplanetario, junto con observatorios en la India, Japón, y Noruega, que contribuyen con datos esenciales para el pronóstico del tiempo espacial. El MEXART será el único observatorio de este tipo en el Continente Americano y, por ende, prestará una cobertura crítica en longitud para complementar los datos de los otros observatorios.

La técnica de centelleo interplanetario consiste en detectar la variación aleatoria de la intensidad de la emisión de un astro. Este fenómeno es esencialmente lo mismo que el muy conocido destello de las estrellas. En ambos casos se requiere de un astro pequeño (puntual) y algún medio que distorsione el frente de onda que nos llega. En el caso de las estrellas, las ondas son de luz visible y son las capas de la atmósfera terrestre las que producen las fluctuaciones. En el de centelleo interplanetario, las fuentes de radio son cuásares cuyos frentes de onda están distorsionados por cambios en la densidad del viento solar en el medio interplanetario. El esquema del proceso se indica en la figura 1 en la página 9.

Como se ve en la figura, un solo astro (cuáasar) nos presta información sobre la estructura del medio interplanetario sobre una zona visual. Para inferir cómo se encuentra el viento solar, y si existen perturbaciones de gran escala propagándose a través de éste, se requiere de muchas fuentes de radio bien distribuidas sobre la bóveda celeste, para así poder construir un mapa de las variaciones en densidad que se encuentran dentro del viento solar.

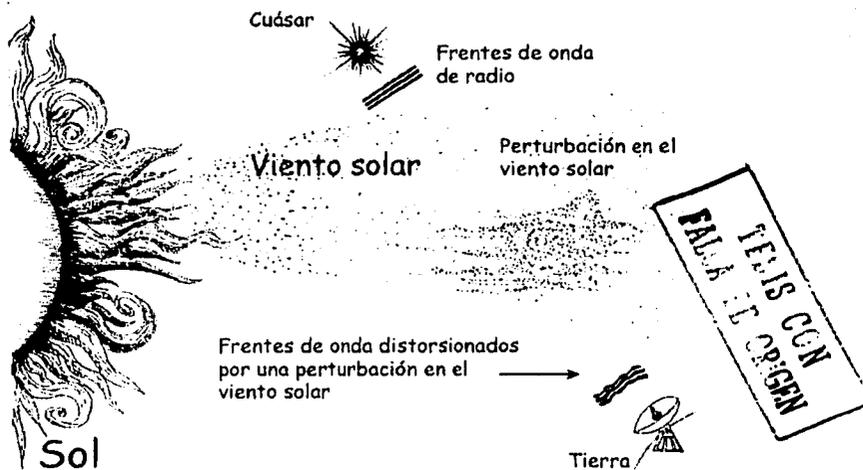


Figura 1: El centelleo interplanetario

El MEXART tiene como propósito la observación de cientos de cuásares y la medición de su centelleo para construir mapas detallados de la estructura del plasma interplanetario. Para efectuar estas observaciones, se necesita un catálogo de los cuásares puntuales visibles desde el sitio del MEXART. Se consideran puntuales a los que tienen tamaños angulares menores a un minuto de arco, pero que son suficientemente fuertes en su emisión, de tal forma que se puedan medir las variaciones en su intensidad con una buena relación señal-a-ruido en poco tiempo de observación. Adicionalmente, para propósitos de verificación y calibración del

instrumento, es indispensable contar con una lista de fuentes fuertes, de posición celestial conocida, para identificar las señales captadas por el telescopio.

El objetivo de este trabajo es crear una base de datos que contenga toda la información existente en los catálogos seleccionados por contener fuentes accesibles al MEXART, y proveer de un mecanismo de búsqueda rápida, eficiente y correcta, a través de una interfase web para que la base de datos pueda ser consultada desde cualquier punto en el Internet. Esta base de datos y su mecanismo correspondiente servirán en las etapas iniciales para hacer pruebas y calibración del instrumento. En etapas más avanzadas, servirán para construir la muestra de astros que se monitorearán todos los días para mapear la estructura del medio interplanetario, prestando información valiosa para estudios teóricos y avisándonos, en su caso, de alguna tormenta geomagnética inminente.

Capítulo 1

La creación de la base de datos

Para crear la base de datos se debió pasar por varias etapas, la primera fue identificar las fuentes de información adecuadas para el proyecto, la segunda fue leer y entender el formato de cada una de ellas y, finalmente, integrar esa información en un formato único que fuera útil para su uso como base de datos de calibración. Una vez terminadas estas etapas, se implementó un programa que leyó las fuentes de información y las transformó en la base de datos para la calibración de MEXART.

1.1. Identificación de las fuentes de información

Para crear la base de datos, lo primero que se hizo fue identificar y obtener los catálogos de radio fuentes de otros observatorios que podrían ser útiles para la tarea de la calibración de MEXART. Dichos catálogos fueron escogidos con base en dos criterios clave: el primero es que deben cubrir la misma zona de cielo que puede “ver” MEXART, es decir, los astros cuyas declinaciones están al Norte de

$\delta = -60^\circ$, y el segundo, que los catálogos fueran de bajas frecuencias, cercanas a la frecuencia de operación de MEXART, que es de 140 MHz, de tal forma que exista cierta probabilidad de que las fuentes detectadas por otros observatorios también sean detectadas por MEXART. Con estos criterios, se determinó seleccionar los catálogos que están en el cuadro 1.1.

Una vez identificados los catálogos, se obtuvieron del sitio Internet *CATS Database, Astrophysical CATalogs support System* ubicado en la dirección <http://cats.sao.ru>. Algunos de estos catálogos constan de varias partes, por lo que se tuvieron que recolectar todas ellas para incorporarlas en la base de datos.

1.2. Definición del formato de la base de datos

Puesto que los catálogos estaban en formatos muy diversos, fue necesario leer cada uno de ellos, analizar cuál era la información común a todos y, a partir de allí, definir un formato que contuviera la información conveniente para el proyecto.

La base de datos que se elaboró contiene toda la información encontrada en cada uno de los catálogos seleccionados. Sin embargo, parte de ella no fue clasificada, en particular aquella que no era relevante para el problema que se quería resolver. Tampoco se clasificó a la información que no era común en todos los catálogos, como es el caso de los flujos entero y del haz, el tamaño angular, las fuentes posibles, y otros datos que sólo están disponibles en algunos de los catálogos seleccionados.

La información común en todos los archivos de radio fuentes fue la siguiente: la ascensión recta y la declinación, en coordenadas B1950.0 y el flujo pico en

Janskys¹. Con base en esta información, se construyó la base de datos.

1.2.1. El gestor de la base de datos

Al principio del proyecto, se pensó en tener toda la información en un archivo de texto plano. Sin embargo, éste medía aproximadamente 14 Megabytes y, para cada consulta, era necesario leer el archivo por completo, lo cual toma del orden de cinco segundos. Para un sistema de consulta en línea, ese lapso es inaceptable.

Además, tener la información en un archivo de texto plano, limita el tipo de consultas de información que se pueden hacer, puesto que el formato del archivo determina, en gran medida, el tipo de búsqueda posible. Para hacer otras búsquedas, es necesario crear nuevos archivos auxiliares, lo cual complica innecesariamente el proceso.

Debido a la lentitud en el acceso de los archivos de texto plano y a su inflexibilidad para hacer consultas, se determinó utilizar un gestor de base de datos que resuelve estos problemas y, como beneficio adicional, se utiliza un lenguaje estándar para búsquedas que permite que otros programadores puedan hacer uso de la información.

Hay muchos gestores de bases de datos y por ello se tuvo que elegir uno de ellos, para elegirlo, los criterios fueron que, por un lado, se cumpliera con el estándar para la consulta de información SQL (*Standard Query Language* o lenguaje estándar para búsquedas) que está inspirado en los trabajos sobre álgebra relacional desarrollados por el matemático Edward F. Codd[2] y, por otro lado, a las limitaciones económicas del proyecto, por lo cual fueron descartados los gestores de base de datos comerciales. Con estos criterios, quedaron sólo dos op-

¹La unidad de densidad de flujo utilizada es el Jansky (Jy). 1 Jy = 10^{-26} W m⁻² Hz⁻¹

1115 CON
FALLA EF ORIGEN

ciones, PostgreSQL[1] y MySQL[13]. Se decidió al segundo debido a que es más simple de utilizar y funciona tanto en Unix como en Windows.

1.2.2. Implementación de la base de datos

La estructura de la base de datos consiste en una tabla única que contiene los campos enlistados a continuación

id	Un número único consecutivo	entero
nf	El nombre de la fuente	20 caracteres
ar	La ascensión recta en coordenadas J2000.0	flotante
dc	La declinación en coordenadas J2000.0	flotante
flujo	El flujo pico en Janskys [Jy]	flotante
archivo	El nombre del archivo en donde está el catálogo original	20 caracteres
línea	El número de línea en donde está la información original	entero
original	La línea original que se encontró en el catálogo	254 caracteres

1.3. La conversión de los datos originales

A partir del análisis de los datos contenidos en los catálogos, se buscó un método de conversión al que se le pudiera dar mantenimiento y en el que fuera fácil incorporar un nuevo catálogo. Cuatro de los siete catálogos necesitaban un trato especial. En el cuadro 1.1, que está en la página 15 se muestra el trato especial que se debió hacer.

En todos los casos se debieron convertir las coordenadas de ascensión recta y declinación del sistema Besseliano B1950.0 al sistema Juliano J2000.0.

Para leer y convertir los datos, se eligió el lenguaje Perl[12] debido a que

Catálogo	Proceso especial
3C	Ninguno
4C	Ninguno
6C	Se convirtió la información a partir de la ascensión recta y la declinación en radianes para las partes 4, 5 y 6 del catálogo
7C	Se generó el nombre de la fuente a partir de las coordenadas
8C	Ninguno
Miyun	Se generó el nombre de la fuente a partir de las coordenadas
Wenss	Se generó el nombre de la fuente a partir de las coordenadas

Cuadro 1.1: Catálogos que integran la base de datos

maneja con facilidad expresiones regulares², arreglos asociativos y una función especializada en extracción de datos llamada *unpack*, que lee información a partir de una cadena de texto y la convierte en un conjunto de variables o un arreglo.

En el caso del catálogo 7C1A, por ejemplo, se tiene la siguiente línea de información:

```
05 45 41.03 .450 06 51.56 .434 36.45 86 54 E . . 2 .779 .664 .153 Pr P Us
```

Para extraer la información de esa fuente, se utiliza la instrucción *unpack* de la siguiente forma:

```
warreglo=unpack("x A2 x A2 x A5 x2 A3 x A2 x A5 x3 A6 x3 A*", línea_del_catálogo);
```

²Las expresiones regulares son "cadenas de símbolos de un alfabeto que consiste en la unión de un alfabeto objetivo y un conjunto de símbolos de operación y construcción de grupos. Una expresión regular bien formada define un lenguaje regular desde la cerradura del alfabeto objetivo"[9]

La función *unpack* tiene dos argumentos, el formato en el que está la información y el material que se va a procesar que es, normalmente, un texto.

El formato se lee de la siguiente forma, la letra *q*quis significa ignorar, la letra *A* mayúscula significa que los datos son alfanuméricos. El dígito que se pone junto a la letra es el número de caracteres que se procesarán. Al final del formato hay un asterisco después de la letra *A*, lo que significa que se procesará el resto del material como un sólo elemento alfanumérico.

Esta instrucción procesa todas las líneas de ese catálogo. Con instrucciones similares, se procesaron todos los demás catálogos para su posterior conversión al formato de la base de datos.

Para que el programa se pudiera ampliar fácilmente, se utilizó un arreglo asociativo “%cat”, en donde se encuentran los formatos de cada uno de los catálogos y banderas que indican el o los procesos especiales que se requieren para integrarlo a la base de datos. %cat contiene los elementos que se muestran en el cuadro 1.2 que está en la página 19.

Una vez procesados todos los catálogos, el producto es un archivo de texto llamado *astroflux-db.sql* que sirve para incorporar la información al gestor de base de datos. En el apéndice A se muestra una porción de ese archivo.

Las instrucciones para incorporar la información al gestor de base de datos, son las siguientes:

```
mysqladmin create astroflux
mysql astroflux
grant all on astroflux.* to max identified by 'password'
\q
mysql astroflux < astroflux-db.sql
```

La primera línea crea la base de datos `astroflux`, la segunda, llama al conector de la base de datos y se conecta con la base de datos `astroflux`. Luego, se le dan todos los permisos al usuario `max`, con la contraseña `password`. La cuarta línea termina el uso del cliente de la base de datos y, finalmente, se importa toda la información generada por el programa a la base de datos. Una vez concluido este proceso, ya se puede consultar la base de datos a través del programa de consulta del que se tratará el siguiente capítulo.

1.4. La conversión B1950.0 a J2000.0

Todos los catálogos que se utilizaron están en el sistema de coordenadas *besseliano* B1950.0. Para tener una base de datos más simple de consultar, se convirtieron dichas coordenadas al sistema *juliano* J2000.0, de tal forma que, durante la etapa de búsqueda, no sea necesario hacer los cálculos de *precesión*.

La forma rigurosa para convertir coordenadas de una época a otra, está dada por la fórmula matricial para el cálculo de *precesión* que describe *Green*[10] y que se incluye en el apéndice 3.

Este método de conversión se encuentra ya programado en la biblioteca para Perl `Astro::Coords`[11]. La forma en la que se debe invocar la función de conversión entre la época *besseliana* B1950.0 y la *juliana* J2000.0 es

```
($JAR,$JDec) = fk4fk5(str2turn($x_ar,'H'), str2turn($x_dec,'D'));
```

La función `fk4fk5` usa dos parámetros, la *ascensión recta* y la *declinación*, ambas en *vueltas*, por ello se utiliza la función auxiliar `str2turn` para convertir la *ascensión recta* y la *declinación* a ese formato. Dicha función devuelve la *ascensión recta* y la *declinación*, también en *vueltas*, en coordenadas J2000.0, por lo

que luego se debe convertir a radianes, que es el formato que eligió utilizar en la base de datos. El nombre de la función $Fk4$ $Fk5$ se debe a que $Fk4[4]$ y $Fk5[5]$ son dos catálogos muy precisos que contienen 1,535 estrellas fundamentales ajustadas para la época B1950.0 y J2000.0 respectivamente.

Para comprobar el funcionamiento correcto de la biblioteca se hicieron varios cálculos manuales que cubrieron un intervalo amplio en ascensión recta y declinación. En particular se consideraron los casos críticos, que son los astros cercanos a los polos y al cenit. La comparación de esos cálculos manuales con el resultado obtenido por la biblioteca, muestra que el margen de error es del orden de tres centésimas de segundo de arco. Dado que la precisión de los catálogos es de un segundo de arco o mayor, y que la resolución angular del MEXART es de un grado, este margen de error es totalmente aceptable.

archivo	Es el nombre del archivo que contiene los datos del catálogo
título	Es el nombre del catálogo
url	Es el URL de donde se tomó el catálogo, allí suele estar la descripción de los campos
formato	Es una cadena para usar con la función <code>unpack()</code>
radianes	Es una bandera que, si existe, indica que la ascensión recta y la declinación están en radianes
nombre	Es la posición del arreglo autocampos en donde se encuentra el nombre de la fuente. Si este campo no existe, se genera con la función <code>genera_nombre()</code>
ra	Arreglo asociativo en donde se encuentran los componentes de la ascensión recta (h - horas m - minutos s - segundos)
dec	Arreglo asociativo en donde se encuentran los componentes de la declinación (g - grados m - minutos s - segundos)
flux	Es el flujo en Janskys (Jy), está en un arreglo asociativo que puede contener lo siguiente int - El flujo entero peak - El flujo pico beam - El flujo del haz

Cuadro 1.2: el arreglo asociativo %cat

TESIS CON
FALLA DE ORIGEN

Capítulo 2



Las herramientas de consulta

2.1. Uso de la información

Una vez que se tuvo la base de datos en funcionamiento, fue necesario generar un programa que permitiera consultar la información. Este programa se elaboró con base en las necesidades del usuario que dependieron de la etapa en la que se encontraba el proyecto MEXART. En este momento, sólo se necesita identificar las fuentes que están en el cenit. De forma adicional, se requiere que el programa pueda ser utilizado desde varias máquinas distintas y en cualquier lugar, en particular en las distintas sedes del proyecto. Es por este segundo requisito por lo cual se decidió hacer un programa que funcione en Internet, para que pueda ser consultado desde cualquier parte del mundo que tenga conexión a esta red.

2.2. Los datos de entrada

Debido a la naturaleza de uso del programa, se requiere que se proporcionen la fecha y la hora para las que se quiere saber las fuentes atraviesan por el cenit. Para mejorar la calidad de la información, se agregaron algunas otras entradas que facilitan al usuario el uso del programa. A una de ellas, le llamamos radio, y es el punto alrededor del cual se quiere consultar que fuentes hay. A partir de ese punto se forma un rectángulo con el doble del valor especificado. Esto sirve para que, en caso de no existir ninguna fuente en el punto seleccionado, se puedan consultar los mejores candidatos posibles de lo que se detectó. También hay una entrada llamada límite de flujo, que permite discriminar las fuentes que tengan una potencia menor a la especificada en dicha entrada.

El programa de consulta tiene las siguientes entradas:

Fecha	Es la fecha en que se quiere hacer la observación
Tiempo universal	La hora en formato de tiempo universal
Tiempo local	Es la hora local a la que se quiere hacer la observación
Tiempo local sidéreo	Es la hora local sidérea
Radio	Es la ventana de observación
Límite de Flujo	Es el flujo mínimo que se va a mostrar

Para simplificar el trabajo del usuario, se decidió poner como valores fijos la longitud y latitud del telescopio MEXART que son 19°48' Norte, 101°41' Oeste, respectivamente.

La pantalla de consulta de la información es la mostrada en la figura 2.1 de la página 23.

Búsqueda de fuentes astronómicas - Mozilla

Ele Edt View Go Bookmarks Tools Window Help

Back Forward Reload Stop http://localhost/busca.fuentes.pl

Búsqueda de fuentes astronómicas

Fecha Tiempo universal Radio (ggg mm ss)

(dd/mm/aaaa) Tiempo local Límite de flujo

Tiempo local sidéreo

Mensajes

Buscar

Nombre	Ascensión Recta J2000.0	Declinación J2000.0	Flujo entero	Distancia angular (grados)	Catálogo	Línea
7C1001+2042	10:04:24	20:42:14	3.6	0.980	4c.dat	2148
7C1001+2043	10:04:24	20:42:13	3.07	0.993	7C4_RA09-16.dat	918
7C1003+2006	10:02:57	19:45:39	5.54	1.081	7C4_RA09-16.dat	902
7C1003+2007	10:02:57	19:45:31	5.1	1.062	4c.dat	2147
7C1004+2113	10:07:15	20:45:43	1.37	1.196	7C4_RA09-16.dat	970
7C1009+2112	10:12:08	20:45:27	1.12	1.584	7C4_RA09-16.dat	1047
7C1008+2127	10:11:09	21:12:52	5.29	1.659	7C4_RA09-16.dat	1029
7C1008+2127	10:11:10	21:13:00	5.1	1.662	4c.dat	2213

TESIS CON FALLA DE ORIGEN

Figura 2.1: Pantalla de consulta de la información

2.3. Consideraciones de programación

Aún cuando se eligió Perl por sus ventajas para programar páginas web, se descubrió durante la etapa de pruebas que era muy incómodo seleccionar, de forma manual, el tiempo que se quería usar. Para simplificar la interfase de usuario, se programaron algunas funciones en *JavaScript*[8] que calculan, a partir de una de las casillas de tiempo, las otras dos. Es decir, si se modifica la casilla de **Tiempo Universal**, el programa calculará el tiempo local y el tiempo sidéreo. De forma

similar, el programa actuará cuando se modifique cualquiera de las otras dos casillas de tiempo, ya sea local o sidéreo. También se considera si se está en horario de verano o no. Esto se hace utilizando el reloj de la computadora del usuario, por lo que se deberá cuidar que la máquina de consulta esté correctamente configurada.

Además de hacer estos cálculos, *JavaScript* revisa que la información que se pone en las casillas sea aceptable para el programa, es decir, que la fecha exista, que la hora sea posible y que esté dentro del intervalo permitido, que el flujo sea un valor numérico y que el radio tenga valores aceptables.

El código *JavaScript* está incorporado dentro del programa en Perl que construye la página de búsqueda de fuentes. Puesto que las casillas de tiempo siempre son recalculadas dinámicamente, no es necesario incorporar cálculos de conversión de tiempo dentro del programa en Perl. Por ello se utiliza, como único parámetro para la búsqueda, el tiempo local sidéreo, que puede haber sido calculado a partir de alguno de los otros dos tiempos, o introducido directamente.

El programa de búsqueda se simplifica porque, como el radio telescopio apunta al cenit, simplemente se convierte el tiempo sidéreo local a radianes y se busca en la base de datos a las fuentes que estén cerca. Lo "cerca" está determinado por el parámetro "radio" que permite dar un intervalo hacia adelante y atrás en el tiempo sidéreo local y hacia un lado y hacia el otro en la coordenada de latitud. La longitud y la latitud del radio telescopio son datos fijos dentro del programa y sólo pueden ser modificados si se altera el código. Se decidió hacerlo así para evitar confundir al usuario con un exceso de variables en la página de búsqueda.

La búsqueda de las fuentes se limita por la intensidad mínima de las fuentes que se quieren detectar por medio del parámetro "Límite de Flujo [Jy]". Si se quieren ver todas las fuentes que están en la base de datos, se puede poner en esa

casilla el valor cero, ya que los flujos son siempre valores positivos.

Una vez seleccionados los parámetros, se puede escoger el botón "Buscar" y, entonces se realizará la búsqueda dentro de la base de datos. La forma en que se hace el proceso de búsqueda es la siguiente: se hace una conexión a la base de datos, se leen los parámetros que introdujo el usuario, se convierte a radianes el tiempo sidéreo local, se calcula el "rectángulo" de la porción del cielo en donde se quiere saber que fuentes hay y, finalmente, se consulta en la base de datos.

El "rectángulo" se construye con las siguientes coordenadas de ascensión recta y declinación respectivamente:

$$\alpha_1 = \alpha - \Delta\alpha$$

$$\alpha_2 = \alpha + \Delta\alpha$$

$$\delta_1 = \delta - \Delta\delta$$

$$\delta_2 = \delta + \Delta\delta$$

en donde

$$\Delta\delta = \text{radio}$$

$$\Delta\alpha = \Delta\delta * \cos \delta.$$

Las unidades de medida son, para la ascensión recta (α), horas, minutos y segundos y, para la declinación (δ), grados, minutos y segundos.

2.4. La consulta a la base de datos

A partir de las coordenadas de ascensión recta y declinación y del rectángulo, se construye la siguiente solicitud de búsqueda para el manejador de la base de datos en lenguaje SQL,

```

SELECT id,nf,ar,de,flujo,
       round(degrees(sqrt(pow((ar-$r_ar) *
       cos((de+$r_dec)/2),2) +
       pow(($r_dec-de),2))),3)
AS da,archivo,linea
FROM astrodb
WHERE ar > $ar_inf AND ar < $ar_sup
      AND de > $dec_inf AND de < $dec_sup
      AND flujo >= $limiteflujo
ORDER BY da,ar,de

```

Esta instrucción significa que seleccione los campos identificador (id), nombre de la fuente (nf), ascensión recta (ar), declinación (de), flujo, distancia angular (da), el nombre del archivo (archivo), y la línea en donde estaba la información (línea).

La variable de distancia angular da se genera de acuerdo con la siguiente fórmula,

$$da = \sqrt{\left((\alpha_n - \alpha) * \frac{\cos(\delta_n + \delta)}{2} \right)^2 + (\delta - \delta_n)^2}$$

Esos campos los extrae de la tabla llamada astrodb a condición de que la ascensión recta esté acotada por α_1 y α_2 y que la declinación esté acotada por δ_1 y δ_2 y que el flujo sea mayor o igual que el parámetro Límite de Flujo. Finalmente se le indica que organice la lista por distancia angular.

2.5. Los resultados de la consulta

Una vez hecha la consulta, el programa despliega en una tabla, todas las fuentes que encontró. Además, se hace un enlace con el identificador de cada fuente porque si el usuario quiere como estaba la información dentro del catálogo original, puede hacerlo presionando sobre el hipervínculo. Al hacerlo, se abrirá una pequeña ventana que muestra toda la información disponible sobre esa fuente en particular. Para hacer este despliegue, se utiliza otro programa llamado `muestra-fuente.pl` que hace la siguiente consulta a la base de datos:

```
SELECT * FROM astrodb WHERE id=$id
```

Esa consulta selecciona todos los campos de la tabla `astrodb`, cuando el identificador es `id`. Como `id` es único, sólo mostrará la fuente seleccionada.

2.6. La precesión desde el año 2000 hasta el día de hoy

Desde el año 2000 han transcurrido tres años, pero la base de datos está en coordenadas J2000.0, por lo que fue necesario considerar si era necesario calcular la precesión entre ese momento y el día de hoy. Hay una fórmula muy simplificada, utilizada en el libro de Gil[6] que es la siguiente.

$$t = \frac{\text{dia_juliano_de_hoy} - \text{dia_juliano_del_1o_enero_2000}}{100 * \text{dias_del_año}}$$

$$t = \frac{2452875 - 2541545.0}{36525} \approx 0.036\text{s.}$$

Una desviación de esa magnitud, hace necesario introducir cálculos de corrección en este año. Sin embargo, si fuese necesario calcular la precesión de una fuente cualquiera, se puede usar la fórmula aproximada del texto de Duffett[3]

$$\alpha_1 = \alpha_0 + (3.07420 + 1.33589 \sin \alpha_0 \tan \delta_0) \times N$$

$$\delta_1 = \delta_0 + (20.0383 \cos \alpha_0) \times N$$

En donde N es el número de años transcurridos desde el año 2000. Si se toma, por ejemplo, la fuente con $\alpha_0 = 19:07:23$ y $\delta_0 = 19^\circ 55' 46''$, se tiene que, transcurridos veinte años, $\alpha_1 = 19:07:26$ y $\delta_1 = 20^\circ 2' 4.65''$, por lo que será necesario incorporar las fórmulas de precesión en algunos años.

Capítulo 3

Conclusiones

TESIS CON
FALLA DE ORIGEN

El presente trabajo presentó un método simple para consultar una base de datos de fuentes de radio astronómicas que puede ser utilizada, entre otras cosas, para calibrar al radio telescopio MEXART y para organizar las fechas de observación de los investigadores que utilizan el equipo.

Debido a la brevedad del tiempo utilizado para elaborar el trabajo, el alcance del proyecto es limitado, por ello, se han pensado hacer un grupo de extensiones al programa.

1. Importación de nuevos catálogos. Aunque se importaron siete catálogos que cubren la región del cielo cercana a la que observa MEXART, es posible agregar nuevos catálogos para que la base de datos cubra todos los radio objetos detectados por los equipos de diferentes observatorios radio astronómicos.

2. Precesión. Con el paso del tiempo, los objetos observados por MEXART diferirán de la información que se encuentra en el catálogo actual debido a la precesión terrestre. La diferencia es muy pequeña. En veinte años será de apenas tres segundos de tiempo en la ascensión recta y de seis minutos de arco en la

declinación. En cincuenta años, esta diferencia hará indispensable recalcular las posiciones de los astros utilizando las matrices de precesión.

3. Ascensión recta, declinación y otras búsquedas. Quizá la más importante ampliación a este proyecto es que la ventana de búsqueda no se restrinja a un rectángulo en el cenit, sino que se pueda ver otra porción del cielo a partir de un par de coordenadas de azimut y elevación, como si fuese un telescopio normal. MEXART, en una etapa posterior, podrá hacer este movimiento utilizando matrices de Butler o a través de transformadas de Fourier, y por ello, será necesario calibrarlo nuevamente.

Apéndice A

Cálculo de precesión por el método matricial

TESIS CON
FALLA DE ORIGEN

La forma exacta de calcular la precesión, consiste en hacer la siguiente operación matricial:

$$s = P s_0$$

En donde

$$s_0 = \begin{pmatrix} x_0 \\ y_0 \\ z_0 \end{pmatrix} = \begin{pmatrix} \cos \delta_0 \cos \alpha_0 \\ \cos \delta_0 \operatorname{sen} \alpha_0 \\ \operatorname{sen} \delta_0 \end{pmatrix}$$

$$s = \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} \cos \delta \cos \alpha \\ \cos \delta \operatorname{sen} \alpha \\ \operatorname{sen} \delta \end{pmatrix}$$

32APÉNDICE A. CÁLCULO DE PRECESIÓN POR EL MÉTODO MATRICIAL

, la matriz de rotación,

$$P = \begin{pmatrix} -\operatorname{sen} \zeta_A \operatorname{sen} z_A + \cos \zeta_A \cos z_A \cos \theta_A & \operatorname{sen} \zeta_A \cos z_A + \cos \zeta_A \operatorname{sen} z_A \cos \theta_A & \cos \zeta_A \operatorname{sen} \theta_A \\ -\cos \zeta_A \operatorname{sen} z_A - \operatorname{sen} \zeta_A \cos z_A \cos \theta_A & \cos \zeta_A \cos z_A - \operatorname{sen} \zeta_A \operatorname{sen} z_A \cos \theta_A & -\operatorname{sen} \zeta_A \operatorname{sen} \theta_A \\ -\cos z_A \operatorname{sen} \theta_A & -\operatorname{sen} z_A \operatorname{sen} \theta_A & \cos \theta_A \end{pmatrix}$$

y las variables

$$\zeta_A = 0^\circ.6402633T + 0^\circ.0000839T^2 + 0^\circ.0000050T^3,$$

$$z_A = \zeta_A + 0^\circ.0002197T^2,$$

$$\theta_A = 0^\circ.5567376T - 0^\circ.0001183T^2 - 0^\circ.0000117T^3$$

Apéndice B

TESIS CON
FALLA DE ORIGEN

Formato del archivo astroflux-db.sql

```
# Current Database: astroflux
CREATE DATABASE /*132312 IF NOT EXISTS*/ astroflux;
USE astroflux;
CREATE TABLE astroddb (
  id          int NOT NULL auto_increment,
  nf          varchar(20) NOT NULL default '',
  ar          double precision NOT NULL default '0.0',
  de          double precision NOT NULL default '0.0',
  flujo       double precision NOT NULL default '0.0',
  archivo     varchar(20) NOT NULL default '',
  linea       int NOT NULL default '0',
  original    varchar(254) NOT NULL default '',
  primary key (id)
) TYPE=MyISAM;
INSERT INTO astroddb VALUES (1, '4C0005-0615',
  0.0370955188100962,-0.104428866910994,   8.2,'4c.dat',1,
  '-06.01      0 5 56.6 -06 15 42      8.2  1   95.2 \
-66.5 a  3C3');
INSERT INTO astroddb VALUES (2, '4C0008-0613',
  0.0468475460056145,-0.103793760988741,   3.0,'4c.dat',2,
  '-06.02      0 8 10.7 -06 13 30      3.0  2   96.5 \
-66.7 b');
INSERT INTO astroddb VALUES (3,'4C0040-0630',
  0.186619330269494,-0.108898849050824,   5.3,'4c.dat',3,
  '-06.03      0 40 13.8 -06 30 48      5.3  1  116.9 \
-69.0 a');
INSERT INTO astroddb VALUES (4, '4C0046-0623',
  0.213751927932789,-0.106828694632486,   4.5,'4c.dat',4,
```

```
'-06.04      0 46 27.1 -06 23 36      4.5  2  121.2 \  
-69.0 b');  
INSERT INTO astrodb VALUES (5, '4C0125-0613',  
0.385303248993398, -0.104162219386384,  2.6, '4c.dat', 5,  
'-06.05      1 25 47.2 -06 13 36      2.6  21  147.1 \  
-67.1 be');  
INSERT INTO astrodb VALUES (6, '4C0135-0648',  
0.426682096676097, -0.114420876878662,  2.2, '4c.dat', 6,  
'-06.06      1 35 16.7 -06 48 36      2.2  2  153.2 \  
-66.7 b');  
INSERT INTO astrodb VALUES (7, '4C0200-0618',  
0.53646330662654, -0.105854219133456,  2.1, '4c.dat', 7,  
'-06.07      2  0 26.8 -06 18 18      2.1  1  165.1 \  
-62.9 a');
```

...

Apéndice C

genera-db.pl

TESIS CON
FALLA DE ORIGEN

```
#!/usr/bin/perl
# genera-db.pl genera la base de datos a partir de los catálogos

use strict;
use Astro::Coord;
use Astro::Time;

my $debug = 0;

# Los catálogos que se transformaron son 3C, 4C, 6C, 7C, 8C,
# MIYUN, WENSS@325MHz y WENSS@352MHz

# La información que se guardará en la base de datos es
# - Un número único consecutivo
# - El nombre de la fuente
# - Ascensión recta en J2000.0 (en radianes)
# - Declinación en J2000.0 (en radianes)
# - Flujo en Jy
# - Nombre del archivo
# - Número de línea
# - Línea original

# Se genera un arreglo asociativo %cat en donde está la
# descripción y forma de extracción de cada uno de los
# catálogos. El formato de cada elemento es el siguiente

#   archivo - Es el nombre del archivo que contiene los
#             datos del catálogo
#   titulo  - Es la descripción del catálogo
#   url     - Es el URL de donde se tomó el catálogo,
```

```

#         allí mismo suele estar la descripción de
#         los campos
#         formato - Es una cadena para usar con la función unpack()
#         radianes- Si existe, indica que la ascensión recta y la
#         declinación están en radianes
#         nombre  - Es la posición del arreglo autocampos en donde se
#         encuentra el nombre de la fuente, si no existe, es
#         necesario generarlo con la función genera_nombre()
#         ra      - Arreglo asociativo en donde se encuentran los
#         componentes de la ascensión recta
#                 h - horas
#                 m - minutos
#                 s - segundos
#         Si la bandera radianes está puesta, entonces sólo
#         hay un elemento
#         dec     - Arreglo asociativo en donde se encuentran los
#         componentes de la declinación
#                 g - grados
#                 m - minutos
#                 s - segundos
#         Si la bandera radianes está puesta, entonces sólo
#         hay un elemento
#         flux    - Es el flujo en Janskys (Jy), está en un arreglo
#         asociativo que puede contener lo siguiente
#                 int - El flujo entero
#                 peak - El flujo pico
#                 beam - El flujo del haz
#         prec    - Es la precisión con que está la información
#                 ra - Precisión de la ascensión recta
#                 dec - Precisión de la declinación
#         Está en número de cifras significativas

my %cat = (# Catálogos
  cat_3c => {
    archivo => "3C.dat",
    titulo  => "A Survey of radio sources at frequency 159 Mc/s.",
    url     => "http://cats.sao.ru/doc/3C.html",
    formato => "A5 x3 A2 x A2 x A4 x2 A x2 A3 x A2 x A3 x A2 x ".
              "A7 x A*",
    nombre => 0,
    ra     => {h=>1, m=>2, s=>3},
    dec    => {g=>5, m=>6, s=>7},
    flux   => {peak=>9},
    prec   => {ra=>7, dec=>7,},
  },

```

```

cat_3cr => {
  archivo => "3CR.dat",
  titulo => "A Survey of radio sources at frequency 159 Mc/s.",
  url => "http://cats.sao.ru/doc/3C.html",
  formato => "A9 x2 A2 x A2 x A4 x A3 x A2 x A3 x A+",
  nombre => 0,
  ra => {h=>1, m=>2, s=>3},
  dec => {g=>4, m=>5, s=>6},
  flux => {peak=>7},
  prec => {ra=>7, dec=>7},
},
cat_4c => {
  archivo => "4c.dat",
  titulo => "The 4C catalogue (1,2) lists the positions "
    "and flux densities of 4843 radio sources as "
    "observed with a transit interferometer of "
    "east-west baseline 469 wavelengths at 178 MHz",
  url => "http://cats.sao.ru/doc/4C.html",
  formato => "A6 x4 A2 x A2 x A4 x2 A3 x A2 x A2 x2 A7 x2 A+",
  ra => {h=>1, m=>2, s=>3},
  dec => {g=>4, m=>5, s=>6},
  flux => {peak=>7},
  prec => {ra=>7, dec=>6},
},
cat_6c1 => {
  archivo => "6c1.dat",
  titulo => "6C Cambridge survey of radio sources at 151 MHz",
  url => "http://cats.sao.ru/doc/6C.html",
  formato => "A2 x A2 x A2 x A3 x A2 x A2 x A6 x A6 x A4",
  ra => {h=>0, m=>1, s=>2},
  dec => {g=>3, m=>4, s=>5},
  flux => {peak=>6, int=>7},
  prec => {ra=>6, dec=>6},
},
cat_6c2 => {
  archivo => "6c2.dat",
  titulo => "6C Cambridge survey of radio sources at 151 MHz",
  url => "http://cats.sao.ru/doc/6C.html",
  formato => "A2 A2 A3 x A2 A2 A2 x3 A5 x2 A5 x3 A2 x2 A2",
  ra => {h=>0, m=>1, s=>2},
  dec => {g=>3, m=>4, s=>5},
  flux => {peak=>6, int=>7},
  prec => {ra=>7, dec=>6},
},
cat_6c3 => {

```

**TESIS CON
FALLA DE ORIGEN**

```

archivo => "6c3.dat",
titulo => "6C Cambridge survey of radio sources at 151 MHz",
url => "http://cats.sao.ru/doc/6C.html",
formato => "A2 x A2 x A4 x2 A2 x A2 x2 A5 x2 A5 x2 A3 "
        "A1 A2 A2 A3",
ra => {h=>0, m=>1, s=>2},
dec => {g=>3, m=>4, s=>5},
flux => {peak=>6, int=>7},
prec => {ra=>7, dec=>6,},
    },
    cat_6c4 => {
archivo => "6c4.dat",
titulo => "6C Cambridge survey of radio sources at 151 MHz",
url => "http://cats.sao.ru/doc/6C.html",
formato => "x A8 x2 A8 x A10 x2 A9 x3 A6 x3 A6 x2 A2 x2 A7",
radianes => 1,
ra => 0,
dec => 1,
flux => {peak=>4, int=>5},
prec => {ra=>7, dec=>7,},
    },
    cat_6c5 => {
archivo => "6c5.dat",
titulo => "6C Cambridge survey of radio sources at 151 MHz",
url => "http://cats.sao.ru/doc/6C.html",
formato => "x A8 x2 A8 x A10 x2 A9 x3 A6 x3 A6 x2 A2 x2 A7",
radianes => 1,
ra => 0,
dec => 1,
flux => {peak=>4, int=>5},
prec => {ra=>7, dec=>7,},
    },
    cat_6c6 => {
archivo => "6c6.dat",
titulo => "6C Cambridge survey of radio sources at 151 MHz",
url => "http://cats.sao.ru/doc/6C.html",
formato => "x A8 x2 A8 x A10 x2 A9 x3 A6 x3 A6 x2 A2 x2 A7",
radianes => 1,
ra => 0,
dec => 1,
flux => {peak=>4, int=>5},
prec => {ra=>7, dec=>7,},
    },
    cat_7c1a => {
archivo => "7C1_062845",

```

```

titulo => "The 7C Survey at 151 MHz",
url     => "http://cats.sao.ru/doc/7C.html",
formato => "x A2 x A2 x A5 x2 A3 x A2 x A5 x3 A6 x3 A*",
ra      => {h=>0, m=>1, s=>2},
dec     => {g=>3, m=>4, s=>5},
flux    => {peak=>6},
prec    => {ra=>8, dec=>8,},
    },
    cat_7c1b => {
archivo => "7C1_102841",
titulo  => "The 7C Survey at 151 MHz",
url     => "http://cats.sao.ru/doc/7C.html",
formato => "x A2 x A2 x A5 x2 A3 x A2 x A5 x3 A6 x3 A*",
ra      => {h=>0, m=>1, s=>2},
dec     => {g=>3, m=>4, s=>5},
flux    => {peak=>6},
prec    => {ra=>8, dec=>8,},
    },
    cat_7c2 => {
archivo => "7C2_170065.dat",
titulo  => "The 7C Survey at 151 MHz",
url     => "http://cats.sao.ru/doc/7C.html",
formato => "A11 x3 A2 x A2 x A4 x2 A2 x A2 x A2 x A10 x2 ".
           "A8 x2 A5 x2 A5 A*",
nombre => 0,
ra      => {h=>1, m=>2, s=>3},
dec     => {g=>4, m=>5, s=>6},
flux    => {beam=>9, peak=>10},
prec    => {ra=>7, dec=>6,},
    },
    cat_7c3a => {
archivo => "7C3_Lasy.tb2",
titulo  => "The 7C Survey at 151 MHz",
url     => "http://cats.sao.ru/doc/7C.html",
formato => "A11 x2 A2 x A2 x A4 x A3 x A2 x A2 x A10 x A9 ".
           "x A4 x2 A4 A*",
nombre => 0,
ra      => {h=>1, m=>2, s=>3},
dec     => {g=>4, m=>5, s=>6},
flux    => {beam=>9, peak=>10},
prec    => {ra=>7, dec=>6,},
    },
    # No se procesó el catálogo 7C3_Lasy.tb3 porque
    # no contiene información sobre los flujos
    cat_7c4 => {

```

```

archivo => "7C4_RA09-16.dat",
titulo => "The 7C Survey at 151 MHz",
url => "http://cats.sao.ru/doc/7C.html",
formato => "A2 x A2 x A4 x2 A3 x A2 x A2 x3 A10 x2 A9 x4 "
        "A5 x A5 A*",
ra => {h=>0, m=>1, s=>2},
dec => {g=>3, m=>4, s=>5},
flux => {beam=>8, peak=>9},
prec => {ra=>7, dec=>6,},
    },
    cat_7c5 => {
archivo => "7C5_Pooley.dat",
titulo => "The 7C Survey at 151 MHz",
url => "http://cats.sao.ru/doc/7C.html",
formato => "A1 x A2 A2 A4 A3 A2 A2 x A8 A7 x2 A4 x2 A4 x "
        "A5 A*",
ra => {h=>1, m=>2, s=>3},
dec => {g=>4, m=>5, s=>6},
flux => {peak=>11},
prec => {ra=>7, dec=>6,},
    },
    cat_8c => {
archivo => "8C.dat",
titulo => "Rees 38-MHz survey",
url => "http://cats.sao.ru/doc/8C.html",
formato => "A2 x A2 x A2 x2 A2 x A2 x A2 x A x A7 x A7 A*",
ra => {h=>0, m=>1, s=>2},
dec => {g=>3, m=>4, s=>5},
flux => {peak=>7, int=>8},
prec => {ra=>6, dec=>6,},
    },
    cat_miyun => {
archivo => "miyun.dat",
titulo => "The Miyun 232 MHz survey II: the main list",
url => "http://cats.sao.ru/doc/MIYUN.html",
formato => "x3 A10 x2 A2 x A2 x A4 x2 A2 x A2 x A4 x2 A8".
        "x2 A8 A*",
nombre => 0,
ra => {h=>1, m=>2, s=>3},
dec => {g=>4, m=>5, s=>6},
flux => {peak=>7, int=>8},
prec => {ra=>7, dec=>7,},
    },
    cat_wenss => {
archivo => "final_list_polar.dat",

```

```

titulo => "WENSS main catalogue (sources in the region ".
         "declination 28 to 76 degrees)",
url     => "http://cats.sao.ru/doc/wenss.html",
formato => "A15 x A2 x A2 x A5 x2 A2 x A2 x A4 x2 A11 x2 ".
         "A10 x2 A3 x2 A4 x2 A4 A*",
nombre => 0,
ra     => {h=>1, m=>2, s=>3},
dec   => {g=>4, m=>5, s=>6},
flux  => {beam=>10, peak=>11},
prec  => {ra=>8, dec=>7,},
      },
);

```

```
# Archivos de salida
```

```
my $fsalida = "/var/www/html/astroflux-db.txt";
open(S,">$fsalida") || die "No puedo abrir $fsalida $!";
```

```
# Otro archivo para generar la base de datos
```

```
my $salida_sql = "/var/www/html/astroflux-db.sql";
open(Q,">$salida_sql") || die "No puedo abrir $salida_sql $!";
print Q q{
```

```
# Current Database: astroflux
```

```
# Base de datos que utiliza las extensiones espaciales
```

```
CREATE DATABASE /*:32312 IF NOT EXISTS*/ astroflux;
```

```
USE astroflux;
```

```
CREATE TABLE astrodb (
```

```

  id      int NOT NULL auto_increment,
  nf      varchar(20) NOT NULL default '',
  ar      double precision NOT NULL default '0.0',
  de      double precision NOT NULL default '0.0',
  flujo   double precision NOT NULL default '0.0',
  archivo varchar(20) NOT NULL default '',
  linea   int NOT NULL default '0',
  original varchar(254) NOT NULL default '',
  primary key (id)
) TYPE=MyISAM;
```

```
);
```

```
my ($catalogo, $titulo, $url);
```

```
my (@autocampos);
```

```
my $id = 1;
```

```
foreach $catalogo (sort keys %cat) {
```

```

    $titulo=$cat{$catalogo}{titulo};

```

TESIS CON
FALLA DE ORIGEN

```

$url=${cat}{catalogo}{url};

print "# [{catalogo}] [{titulo}] [{url}]\n";
open(F,"<${cat}{catalogo}{archivo}" ||
die "No puedo abrir ${cat}{catalogo}{archivo} ${!";

while(<F>) {
chomp;
next if /^#/;
next if /~$/;

my ($x_ra, $x_dec, $JRA, $JDec, $JRAs, $JDecs, $x_flux);
my ($Nf, $RAh, $RAm, $RAs, $DEx, $DEd, $DEm, $DEs, $compactflux);

# El arreglo autocampos trae la línea completa desempaquetada de
# acuerdo con el formato establecido en el arreglo asociativo, a
# cada campo definido en el formato, se le pone en un elemento
# del arreglo
@autocampos = unpack(${cat}{catalogo}{formato}, $_);

unless (defined ${cat}{catalogo}{radianes}) {
    $x_ra = "@autocampos[${cat}{catalogo}{ra}{h}] " .
"@autocampos[${cat}{catalogo}{ra}{m}] " .
"@autocampos[${cat}{catalogo}{ra}{s}]";
    $x_ra =~ s/ +/ /g; # Elimina espacios sobrantes
    print "x_ra = $x_ra\n" if ($debug);
    $x_dec = "@autocampos[${cat}{catalogo}{dec}{g}] " .
"@autocampos[${cat}{catalogo}{dec}{m}] " .
"@autocampos[${cat}{catalogo}{dec}{s}]";
    $x_dec =~ s/ +/ /g; # Elimina espacios sobrantes
    print "x_dec = $x_dec\n" if ($debug);

    ($JRA, $JDec) = fk4fk5(str2turn($x_ra,'H'),
str2turn($x_dec,'D'));
} else {
    ($JRA, $JDec) =
fk4fk5(rad2turn(
@autocampos[${cat}{catalogo}{ra}]},
rad2turn(@autocampos[${cat}{catalogo}{dec}]));
}

$JRAs = turn2str($JRA,'H',1);

```

```

$JDecs = turn2str($JDec,'D',0);
$x_flux = @autocampos[$cat{$catalogo}{flux}{peak}];

unless(defined $cat{$catalogo}{nombre}) {
    $Nf = genera_nombre($catalogo);
} else {
    $Nf = @autocampos[$cat{$catalogo}{nombre}]
}

# Aquí se generan dos archivos, uno de texto simple en donde
# está toda la información en ASCII y el otro que contiene el
# código SQL que se utiliza para insertar la información a la
# base de datos MySQL

print S "$Nf|$JRAs|$JDecs|$x_flux|$cat{$catalogo}{archivo}".
        "|$|$_\n";
print Q "INSERT INTO astroddb ".
        "VALUES ($id, '$Nf'," .str2rad($JRAs,'H') .
        ", ".str2rad($JDecs,'D') .
        ", $x_flux, '$cat{$catalogo}{archivo}', $., '$_');\n";
$id++;
    }
    close(F);
}

close(S);
close(Q);

# Esta función genera el nombre de la fuente cuando
# no esté en el catálogo
sub genera_nombre {
    my $catalogo = shift @_;
    my ($Nf, $signo);

    unless(defined $cat{$catalogo}{radianes}) {
        $signo = (@autocampos[$cat{$catalogo}{dec}{g}] >= 0) ? '+' : '-';
        $Nf = uc substr($catalogo, 4, 2) .
            sprintf("%02d%02d",
                @autocampos[$cat{$catalogo}{ra}{h}],
                @autocampos[$cat{$catalogo}{ra}{m}]) .
            $signo .
            sprintf("%02d%02d",
                abs(@autocampos[$cat{$catalogo}{dec}{g}]),
                @autocampos[$cat{$catalogo}{dec}{m}]);
    } else {

```

```
my $nstrra = rad2str(@autocampos[$cat{$catalogo}{ra}], 'H', 0);
my $nstrde = rad2str(@autocampos[$cat{$catalogo}{dec}], 'D', 0);
my ($nstrrah, $nstrram, $descarta1) = split /:/, $nstrra;
my ($nstrdeg, $nstrdem, $descarta2) = split /:/, $nstrde;
$signo = ($nstrdeg >= 0) ? '+' : '-';
$Nf = uc substr($catalogo, 4, 2) .
    sprintf("%02d%02d", $nstrrah, $nstrram) .
    $signo .
    sprintf("%02d%02d", abs($nstrdeg), $nstrdem);
}

return $Nf;
}
```

Apéndice D

muestra-fuente.pl

```
#!/usr/bin/perl -T -w

# muestra-fuente.pl
# Muestra todo el detalle sobre una fuente dada

use Astro::Coord;
use Astro::Time;
use CGI qw(:standard);
use DBI;

print header();
print start_html("Muestra fuente");

my $id;
if (!param()) {
    print "Este programa requiere de un parámetro<br>";

    $id = 55;
} else {
    $id = param("id");
}

my $astrodb = DBI->connect('dbi:mysql:astroflux',
                          'max', 'perlita') ||
    die "No me puedo conectar a astroflux $DBI::errstr\n";

my $query="select * from astrodb where id=$id";

my $sth=$astrodb->prepare($query);
```

```

$sth->execute();

@row = $sth->fetchrow_array;

my ($ar,$de);

$ar = turn2str(rad2turn($row[$sth->{NAME_lc_hash}{ar}]),'H',0);
$de = turn2str(rad2turn($row[$sth->{NAME_lc_hash}{de}]),'D',0,
    'deg');

print qq {

<TABLE BORDER=1>
<TR><TH ALIGN=LEFT>Nombre de la fuente</TH>
    <TD>$row[$sth->{NAME_lc_hash}{nf}]</TD></TR>
<TR><TH ALIGN=LEFT>Ascensión recta J2000.0</TH>
    <TD>$ar</TD></TR>
<TR><TH ALIGN=LEFT>Declinación J2000.0</TH>
    <TD>$de</TD></TR>
<TR><TH ALIGN=LEFT>Flujo entero</TH>
    <TD>$row[$sth->{NAME_lc_hash}{flujo}]</TD></TR>
<TR><TH ALIGN=LEFT>Archivo original</TH>
    <TD>$row[$sth->{NAME_lc_hash}{archivo}]</TD></TR>
<TR><TH ALIGN=LEFT>Línea</TH>
    <TD>$row[$sth->{NAME_lc_hash}{linea}]</TD></TR>
<TR><TH ALIGN=LEFT>Línea original del catálogo</TH>
    <TD>$row[$sth->{NAME_lc_hash}{original}]</TD></TR>
</TABLE>
};

print end_html();

1;

```

Apéndice E

busca-fuentes.pl

```
#!/usr/bin/perl -T -w

# busca-fuentes.pl
# Busca las fuentes que se encuentran en el cenit
# a partir de una hora y fecha dadas

use Astro::Coord;
use Astro::Time;
use CGI qw(:standard escapeHTML *table);
use DBI;

my ($sec,$min,$hour,$mday,$mon,$year,$wday,$yday,$isdst);
my ($jd,$mjd,$fecha,$tiempouniversal,$tl,$t1s);
my ($longitud,$latitud);
my ($radio);
my ($ar,$dec);

my $astrodb = DBI->connect('dbi:mysql:astroflux',
                          'max','perlita') ||
    die "No me puedo conectar a astroflux $DBI::errstr\n";

$longitud = "-101 41 0";
$latitud = "19 48 0";

encabezado();

# Si el programa trae parámetros, aquí se procesan
# Sólo se procesa el tiempo local sidéreo, puesto que
# ya fue calculado por el código en JavaScript
if(param()) {
```

```

% Primero borra los mensajes
$mensajes = '
$fecha = param("Fecha");
$tiempouniversal = param("TU");
$t1 = param("TL");
$tl1 = param("TL1");
$radio = param("Radio");
$limiteflujo = param("LimiteFlujo");
} else {
# Aquí van los parámetros por omisión cuando
# el programa corre por primera vez
($sec,$min,$hour,$mday,$mon,$year,$wday,$yday,$isdst) =
localtime();
$mjd = now2mjd();
$j1d = mjd2jd($mjd);
$fecha = sprintf("%02d/%02d/%04d", $mday,$mon+1,$year+1900);
$tiempouniversal = t12tu($hour, $min, $sec, -6, $isdst);
$t1 = sprintf("%02d:%02d:%02d", $hour, $min, $sec);
$tl1 = turn2str(mjd21st($mjd, str2turn($longitud, 'D')),
'H', 0);
$radio = "002 00 00";
$limiteflujo = 1;
}

# Genera la forma de captura de información del usuario
captura();

# Puesto que el radio telescopio apunta al cenit,
# se buscan los objetos cuya ascensión recta sea $tl1
# y la declinación coincide con la latitud del observador
# en un rectángulo de 2 * $radio

$lar = $tl1;
$dec = $latitud;
busca_objetos($lar, $dec, $radio);

# Termina el HTML
pie();

$astrodb->disconnect
|| warn "No me puedo desconectar de astroflux ".
"$DBI::errstr\n";

1; # Aquí termina el programa

```

```

# Sección de subrutinas

# Convierte el tiempo local a tiempo universal
# Sección 9 del Duffett-Smith, pp. 13
sub t12tu {
    my ($h,$m,$s,$tz,$dst) = @_ ;

    if ($isdst) { $h-- ; }
    $h -= $tz ;

    if ($h > 24) { $h -= 24 ; }
    if ($h < 0) { $h += 24 ; }

    return sprintf("%02d:%02d:%02d", $h, $m, $s) ;
}

# Imprime el encabezado del HTML, aquí está todo
# el código en JavaScript
sub encabezado {
    my $titulo = "Búsqueda de fuentes astronómicas";
    my $onLoad = "document.búsqueda.Mensajes.value=' ' ;".
        "document.búsqueda.Fecha.focus(); ";
    my $JavaScript = q 0

    // Se pasa a JavaScript la longitud en grados decimales ;
    var longitud = 0 . str2deg($longitud,'D') . q 0 ;

    // Funciones de validación

    // Esta función valida a toda la forma pero,
    // por el momento, no se utiliza
    function validaForma(elem) {
return true;
    }

    // Primero se verifica que sea una fecha válida
    function validaFecha(elem) {
muestraMensaje(elem,'Búsqueda no actualizada, presione Buscar');
if(revisaFecha(elem)) {
    // Se recalcula con base en el Tiempo Local ;
    var aTL = new Array();
    aTL = isTime(document.forms[elem.form.name].elements['TL']);
    var aTU = new Array();
    aTU = t12tu(elem,aTL);
    colocaTiempo(elem,'TU',aTU);
}
}

```

```

    var aTLS = new Array();
    aTLS = calculaTLS(elem,aTU);
    colocaTiempo(elem, 'TLS', aTLS);
} else {
    return false;
}
return true;
}

// Tiempo Universal
function validaTU(elem) {
var aTU = new Array ();
isEmpty(elem);

aTU = isTime (elem);
if (! aTU) {
    focusElement (elem.form.name, 'TU');
    return false;
}
colocaTiempo (elem, 'TU', aTU);

// Ahora lo que tengo que hacer es actualizar el TL y el TLS;
var aTL = new Array ();
var aTLS = new Array ();

var i=0;
while(i<aTU.length) {
    aTL[i] = aTU[i++];
}

// Se obtiene la corrección del horario de verano (en horas) ;
var correccion = zonaDeTiempo(elem);
if ( aTL[0] < correccion ) {
    aTL[0] = parseInt(aTL[0]) + 24;
}
aTL[0] -= correccion;
colocaTiempo (elem, 'TL', aTL);

aTLS = calculaTLS (elem,aTU);
colocaTiempo (elem, 'TLS', aTLS);

muestraMensaje (elem, 'Búsqueda no actualizada, presione Buscar');

return true;
}

```

```

        // Valida el Tiempo Local ;
        function validaTL(elem) {
            isEmpty(elem);

            var aTL = new Array();
            aTL = isTime(elem);
            if (! aTL) {
                focusElement(elem.form.name, 'TL');
                return false;
            }
            colocaTiempo(elem, 'TL', aTL);

            // Convierte el Tiempo Local a Tiempo Universal ;
            var aTU = new Array();
            aTU = tl2tu(elem, aTL);
            colocaTiempo(elem, 'TU', aTU);

            var aTLS = new Array();
            aTLS = calculaTLS(elem, aTU);
            colocaTiempo(elem, 'TLS', aTLS);

            muestraMensaje(elem, 'Búsqueda no actualizada, presione Buscar');

            return true;
        }

        // Valida el Tiempo Local Sidéreo ;
        function validaTLS(elem) {
            var aTLS = new Array();
            isEmpty(elem);

            aTLS = isTime(elem);
            if (! aTLS) {
                focusElement(elem.form.name, 'TLS');
                return false;
            }
            colocaTiempo(elem, 'TLS', aTLS);

            // Se convierte el Tiempo Local Sidéreo a Tiempo Universal ;

            // Primero convierto el TLS a GST ;
            // Op. Cit. pp. 21 Sección 15 ;
            var tls = hms2dec(aTLS);
            var gst = tls - (longitud / 15);

```

```

while(gst > 24) gst -= 24;
while(gst < 0) gst += 24;

// Luego convierto de GST a UT ;
// Op. Cit. pp. 18 Sección 13 ;
var jd = julianDate(
    document.forms[elem.form.name].elements['Fecha'].value);
// alert('Día juliano='+jd);
var s = jd - 2451545.0;
var t = s / 36525.0;
var tO = 6.697374558 + (2400.051336 * t) + (0.000025862 * t * t);
while ( tO < 0 ) tO += 24;
while ( tO > 24 ) tO -= 24;

var tu = gst - tO;
while ( tu < 0 ) tu += 24;
while ( tu > 24 ) tu -= 24;
tu *= 0.9972695663;

var aTU = new Array();
aTU = dec2hms(tu);
colocaTiempo(elem, 'TU', aTU);

var aTL = new Array();
var i=0;
while(i<aTU.length)
    aTL[i] = aTU[i++];

// Se obtiene la corrección del horario de verano (en horas) ;
var correccion = zonaDeTiempo(elem);

aTL[0] -= correccion;
while( aTL[0] < 24 )
    aTL[0] += 24;
while( aTL[0] > 24 )
    aTL[0] -= 24;
colocaTiempo(elem, 'TL', aTL);

muestraMensaje(elem, 'Búsqueda no actualizada, presione Buscar');

return true;
}

// Validación de fecha según el O'Reilly: JavaScript and
// DHTML Cookbook

```

```

function revisaFecha(elem) {
var m, d, a;
var f=elem.value;
var er = /\b\d{1,2}[/-]\d{1,2}[/-]\d{4}\b/;
if (er.test(f)) {
var delimChar = (f.indexOf("/") != -1) ? "/" : "-";
var delim1 = f.indexOf(delimChar);
var delim2 = f.lastIndexOf(delimChar);
d = parseInt(f.substring(0,delim1),10);
m = parseInt(f.substring(delim1+1,delim2),10);
y = parseInt(f.substring(delim2+1),10);
var pruebaFecha = new Date(y, m-1, d);
if ( pruebaFecha.getDate() == d ) {
if ( pruebaFecha.getMonth()+1 == m ) {
if ( pruebaFecha.getFullYear() == y ) {
return true;
} else {
rechaza(elem,"El año es incorrecto");
}
} else {
rechaza(elem,"El mes es incorrecto");
}
} else {
rechaza(elem,"El día es incorrecto");
}
} else {
rechaza(elem,"El formato de la fecha es incorrecto,\
debe ser dd/mm/aaaa");
}
return false;
}

```

```

function isEmpty(elem) {
var str = elem.value;
var re = /.+;/;
if(str.match(re)){
return true;
} else {
rechaza(elem,"Este campo no puede estar vacío");
return false;
}
}

```

```

function isTime(elem) {
var t = elem.value;

```

```

var h, m, s;
var er = /\b\d{1,2}:\d{1,2}:\d{1,2}\b/;

if(er.test(t)){
    var delim1 = t.indexOf(':');
    var delim2 = t.lastIndexOf(':');
    h = parseInt(t.substring(0,delim1),10);
    m = parseInt(t.substring(delim1+1,delim2),10);
    s = parseInt(t.substring(delim2+1),10);
    if ( h >= 0 && h < 25 ) {
        if ( m >= 0 && m < 60 ) {
            if ( s >=0 && s < 60 ) {
                var aTime = new Array();
                aTime[aTime.length] = h;
                aTime[aTime.length] = m;
                aTime[aTime.length] = s;
                return aTime;
            } else {
                rechaza(elem,'Los segundos deben estar entre 0 y 59');
            }
        } else {
            rechaza(elem,'Los minutos deben estar entre 0 y 59');
        }
    } else {
        rechaza(elem,'La hora debe estar entre 0 y 24');
    }
} else {
    rechaza(elem,'Este campo debe tener el formato HH:MM:SS');
}
return false;
}

// Funciones de validación menores
function validaRadio(elem) {
muestraMensaje(elem,'Búsqueda no actualizada, presione Buscar');
var d, m, s;
var r=elem.value;
var er = /\b\d{1,3}[ ]\d{1,2}[ ]\d{1,2}\b/;

if (er.test(r)) {
    var delimChar = ' ';
    var delim1 = r.indexOf(delimChar);
    var delim2 = r.lastIndexOf(delimChar);
    d = parseInt(r.substring(0,delim1),10);
    m = parseInt(r.substring(delim1+1,delim2),10);

```

```

    s = parseInt(r.substring(delim2+1),10);
    if ( d >= 0 && d < 360 ) {
    if ( m >= 0 && m < 60 ) {
        if ( s >= 0 && s < 60 ) {
        return true;
        } else {
        rechaza(elem,"Los segundos deben estar entre 0 y 59");
        }
        } else {
        rechaza(elem,"Los minutos solo pueden estar entre 0 y 59");
        }
        } else {
        rechaza(elem,"Los grados solo pueden estar entre 0 y 360");
        }
        } else {
        rechaza(elem,"El formato del radio es incorrecto,\
debe ser ggg mm ss");
        }
    }
    return false;
}

```

```

function validaLimiteDeFlujo(elem) {
    isEmpty(elem);
    if(isNaN(elem.value)){
        rechaza(elem,"Debe ser un valor numérico");
        return false;
    } else {
        muestraMensaje(elem,'Búsqueda no actualizada,\
presione Buscar');
        return true;
    }
}

```

```

// Funciones comunes
// Pasar de Tiempo Local a Tiempo Universal
function t12tu(elem,aTL) {
    var aTU = new Array();
    var i = 0;
    while(i<aTL.length) {
        aTU[i] = aTL[i++];
    }
}

```

```

var correccion = zonaDeTiempo(elem);
aTU[0] = parseInt(aTU[0]) + correccion;
while( aTU[0] > 24 ) aTU[0] -= 24;

```

```

while( aTU[0] < 0 ) aTU[0] += 24;
return aTU;
}

// Calcula el Tiempo Local Sidéreo según el
// algoritmo de Practical Astronomy with your
// Calculator pp. 17 ;
function calculaTLS(elem,aTU) {
var aTLS = new Array();
var jd = julianDate(
    document.forms[elem.form.name].elements['Fecha'].value);
var s = jd - 2451545.0;
var t = s / 36525.0;
var tO = 6.697374558 + (2400.051336 * t) + (0.000025862 * t * t);

while ( tO < 0 ) tO += 24;
while ( tO > 24 ) tO -= 24;
var tu = hms2dec(aTU);

var gst = tO + tu * 1.002737909;
while (gst < 0) gst = gst + 24;
while (gst > 24) gst = gst - 24;

// Ahora al GST hay que convertirlo a LST ;
// Luego la convierto a horas según el algoritmo de
// PAC pp. 20
var lst = gst + (longitud / 15.0);
while (lst > 24) lst -= 24;
while (lst < 0) lst += 24;
aTLS = dec2hms(lst);
return aTLS;
}

// Convierte las horas, minutos y segundos a tiempo
// decimal en horas
// Practical Astronomy with your Calculator pp. 10
// Converting hours, minutes and seconds to decimal hours
function hms2dec(aTime) {
var h, m, s;
h = Math.floor(aTime[0]);
m = Math.floor(aTime[1]);
s = Math.floor(aTime[2]);
var t = h + (m + (s / 60) ) / 60;
return t;
}

```

```

// Convierte las horas decimales a horas, minutos y segundos
// Op. Cit. pp. 10
function dec2hms(tdec) {
var aTime = new Array ();
var h, m, s;
var a;
h = Math.floor(tdec);
a = (tdec - h) * 60;
m = Math.floor(a);
a = (a - m) * 60;
s = Math.floor(a);

aTime[0] = h;
aTime[1] = m;
aTime[2] = s;

return aTime;
}

// Op. Cit. pp. 6
function julianDate(fecha) {
var dia, m, y;

dia = fecha.substring(0,2);
m = fecha.substring(3,5);
y = fecha.substring(6,10);

var a, b, c, d;
var yp, mp;

if ( m == 1 || m == 2 ) {
    yp = y - 1;
    mp = Math.floor(m) + 12;
} else {
    yp = Math.floor(y);
    mp = Math.floor(m);
}

// Como las fechas son posteriores al 15 de octubre de 1582
// se excluyen algunas comprobaciones
a = Math.floor(yp / 100.0);
b = 2 - a + Math.floor( a / 4.0 );
if ( yp < 0 ) {
    c = Math.floor( (365.25 * yp) - 0.75 );

```

```

} else {
    c = Math.floor( 365.25 * yp );
}
d = Math.floor( 30.6001 * (mp + 1) );
return (b + c + d + Math.floor(dia) + 1720994.5);
}

// Función para ver la cantidad de horas por sumar o restar
// dependiendo de la zona horaria
// Depende de la configuración horaria de la máquina
// en donde se consulta
function zonaDeTiempo(elem) {
var m, d, a;
var delim1, delim2;
var f = document.forms[elem.form.name].elements['Fecha'].value;
var er = /\b\d{1,2}[\/-]\d{1,2}[\/-]\d{4}\b/;
if (er.test(f)) {
    var delimChar = (f.indexOf("/") != -1) ? "/" : "-";
    var delim1 = f.indexOf(delimChar);
    var delim2 = f.lastIndexOf(delimChar);
    d = parseInt(f.substring(0, delim1), 10);
    m = parseInt(f.substring(delim1+1, delim2), 10);
    y = parseInt(f.substring(delim2+1), 10);
    var Fecha = new Date(y, m-1, d);
    var nTZs = Fecha.getTimezoneOffset();
    return nTZs / 60;
} else {
    return 0;
}
}

// Coloca un tiempo en la casilla indicada
function colocaTiempo(elem, campo, aTime) {
document.forms[elem.form.name].elements[campo].value =
pad2(aTime[0]) + ':' + pad2(aTime[1]) + ':' + pad2(aTime[2]);
}

// Coloca un mensaje en su casilla correspondiente
function muestraMensaje(elem, sMensaje) {
document.forms[elem.form.name].elements['Mensajes'].value =
sMensaje;
}

// Rellena con un cero a la izquierda cuando es necesario

```

```

function pad2(str) {
n = parseInt(str);
if(n<10)
return '0'+n;
else
return n;
}

// Rechaza una modificación
function rechaza(elem,causa) {
alert(causa);
setTimeout("focusElement('"+elem.form.name+
"','"+elem.name+"')",0);
return false;
}

// Pone el cursor en la casilla especificada
function focusElement(formName, elemName) {
var elem = document.forms[formName].elements[elemName];
elem.focus();
elem.select();
}

// Abre en otra ventana la información sobre una fuente
function openWin(URL) {
aWindow = window.open(URL,'informacion',
'toolbar=no,width=700,height=300,\
status=no,scrollbars=no,resize=no,\
menubar=no');
}

// Fin del código en JavaScript
Û;

print header();
print start_html(-title=>$titulo, -script=>$JavaScript,
-onload=>$onLoad);
print hi($titulo);
}

sub pie {
print end_html();
}

sub captura {

```

**ESTA TESIS NO SALE
DE LA BIBLIOTECA**

```

print start_form(-name=>'busqueda',
  -method=>'post',
  -action=>"busca-fuentes.pl",
  -onsubmit=>"return validaForma(this)");

print table( {-border=>0},
Tr({-align=>'left', -valign=>'top'},
  [
    td(['Fecha ',
textfield(
  -name=>'Fecha',
  -default=>$fecha,
  -size=>10,
  -maxlength=>10,
  -onChange=>'validaFecha(this)',
  ),
'&nbsp;'],
'Tiempo universal ',
textfield(
  -name=>'TU',
  -default=>$tiempouniversal,
  -size=>8,
  -maxlength=>8,
  -onChange=>'validaTU(this)',
  ),
'&nbsp;'],
td(['Radio',
  textfield(
    -name=>'Radio',
    -default=>$radio,
    -size=>9,
    -maxlength=>9,
    -onChange=>'validaRadio(this)',
  ),
  '(ggg mm ss)',
  ]),
]),
td([
  ' ', '(dd/mm/aaaa)',
  '&nbsp;'],
'Tiempo local ',
textfield(
  -name=>'TL',
  -default=>$tl,
  -size=>8,

```

```

        -maxlength=>8,
        -onChange=>'validaTL(this)',
    ),
    '&nbsp;';,
    '&nbsp;';,
    'Límite de flujo [Jy]',
    textfield(
        -name=>'LimiteFlujo',
        -default=>$limiteflujo,
        -size=>6,
        -maxlength=>8,
        -onChange=>'validaLimiteDeFlujo(this)',
    ),
    )],
    td({
        '',' ',
        '&nbsp;';,
        'Tiempo local sidéreo ',
        textfield(
            -name=>'TLS',
            -default=>$tls,
            -size=>8,
            -maxlength=>8,
            -onChange=>'validaTLS(this)',
        ),
    )],
    td({-colspan=>'5'}, [
        'Mensajes &nbsp;&nbsp;&nbsp;';.
        textfield(
            -name=>'Mensajes',
            -default=>$mensajes,
            -size=>40,
            -maxlength=>40,
            -onChange=>'validaMensajes(this)',
        ),
    ]),
    td([submit("Buscar"),
        ' ',
        ' ',
        defaults("Actualizar hora")]),
    ]
    );
    print end_form();
}

```

```

sub busca_objetos {
my ($ar, $dec, $radio) = @_;
my ($r_ar, $r_dec, $delta_ra, $delta_dec);
my @row;
my ($query, $sth);
my $nombre_campo;
my $i;

$r_ar = str2rad($ar, 'H');
$r_dec = str2rad($dec, 'D');
$delta_dec = str2rad($radio, 'D');
$delta_ra = $delta_dec * cos ($r_dec);

# Rectángulo de búsqueda
my $ar_inf = $r_ar-$delta_ra;
my $ar_sup = $r_ar+$delta_ra;
my $dec_inf = $r_dec-$delta_dec;
my $dec_sup = $r_dec+$delta_dec;

# Construye la búsqueda y la ejecuta
$query = qq {
SELECT id,nf,ar,de,flujo,
      round(degrees(sqrt(pow((ar-$r_ar) *
      cos((de+$r_dec)/2), 2) +
      pow(($r_dec-de), 2))),3)
AS da,archivo,linea
FROM astrodb WHERE ar > $ar_inf AND ar < $ar_sup
AND
  de > $dec_inf AND de < $dec_sup
AND
flujo >= $limiteflujo
ORDER BY da,ar,de
};
print "\n<!-- $query <br>-->\n";
$sth=$astrodb->prepare($query);
$sth->execute();

# Imprime el encabezado de la tabla
print q {
<TABLE BORDER=1>
<TR ALIGN=CENTER VALIGN=TOP>
<TH>Nombre</TH><TH>Ascensión Recta<BR>J2000.0</TH>
<TH>Declinación<BR>J2000.0</TH><TH>Flujo entero</TH>
<TH>Distancia angular<br>(grados)</TH>

```

```

<TH>Catálogo</TH><TH>Línea</TH>
);

# Imprime lo que encontró
@row = $sth->fetchrow_array;
do {
print "<TR>";
for ($i=1;$i<$sth->{NUM_OF_FIELDS};$i++) {
print "<TD ALIGN=";
$nombre_campo = $sth->{NAME}->[$i];
if ( $nombre_campo eq 'nf' || $nombre_campo eq 'archivo' ) {
print "LEFT";
} else { print "RIGHT"; }
print ">";
SW: {
if ($nombre_campo eq 'nf') {
print "<A HREF=\"javascript:openWin('muestra-fuente.pl?\"
'id=$row[0]');\">$row[$i]</A>"; last SW; }
if ($nombre_campo eq 'ar') {
print turn2str(rad2turn($row[$i]),'H',0); last SW; }
if ($nombre_campo eq 'de') {
print turn2str(rad2turn($row[$i]),'D',0,'deg'); last SW; }
print $row[$i];
}
print "</TD>";
}
print "</TR>\n";
} while (@row = $sth->fetchrow_array);
print "</TABLE>\n";
}

```


Bibliografía

- [1] Oleg Bartunov, D'Arcy Cain, Peter Eisentraut, Oliver Elphick, Daniel Horak, Hiroshi Inoue, Tatsuo Ishii, Christopher Kings-Lynne, Thomas Lockhart, Michael Meskes, Peter Mount, Lamar Owen, Dave Page, Stephan Szabo, and Constantin Teodorescu. Postgresql. <http://www.postgresql.org>, 2003.
- [2] C. J. Date. *Introducción a los sistemas de bases de datos*. Pearson Educación, 2001.
- [3] Peter Duffett-Smith. *Practical Astronomy with your Calculator*. Cambridge University Press, 1988.
- [4] W. Fricke, A. Kopff, W. Gliese, F. Gondolatsch, T. Lederle, H. Nowacki, W. Strobel, and P. Stumpff. Fourth fundamental catalogue (fk4). *Veröffentlichungen des Astronomischen Rechen-Instituts Heidelberg*, 10:144, 1963.
- [5] W. Fricke, H. Schwan, and T. Lederle. Fifth fundamental catalogue (fk5). *Veröffentlichungen des Astronomischen Rechen-Instituts Heidelberg*, 32:106, 1988.

- [6] María de Gracia Gil Cruz, Antonio José; Rodríguez Caderot. *Problemas resueltos de astronomía*. Equipo Sirius, S.A., 2000.
- [7] J.A. González-Esparza, A. Carrillo, E. Andrade, R. Perez, and S. Kurtz. Construction of an interplanetary scintillation array in Mexico. *Geofísica Internacional*, pages 1–23, 2002.
- [8] Danny Goodman. *JavaScript & DHTML Cookbook*. Nutshell Handbook. O'Reilly & Associates, 981 Chestnut Street, Newton, MA 02164, USA, 1st edition, 2003.
- [9] K. John Gough. *Syntax analysis and software tools*. International computer science series. Addison-Wesley Publishing Company, Sydney, 1st. edition, 1988.
- [10] Robin M. Green. *Spherical Astronomy*. Cambridge Press, 1998.
- [11] Chris Phillips. Astro-0.62. <http://www.atnf.csiro.au/people/Chris.Phillips/>, 2003.
- [12] Larry Wall, Randal L. Schwartz, Tom Christiansen, and Stephen Potter. *Programming Perl*. Nutshell Handbook. O'Reilly & Associates, 981 Chestnut Street, Newton, MA 02164, USA, 2nd edition, 1996.
- [13] Michael (Monty) Widenius, David Axmark, Jani Tolonen, Sinisa Milivojevic, Tonu Samuel, and Sasha Pachev et. al. Mysql. <http://www.mysql.com>, 2003.