

TRABAJO ACADÉMICAMENTE DIRIGIDO

DEPARTAMENTO DE FÍSICA DE LA TIERRA, ASTRONOMÍA Y ASTROFÍSICA II
UNIVERSIDAD COMPLUTENSE DE MADRID

Publicación de la base de datos de las
galaxias de la Exploración UCM
en el Observatorio Virtual

Curso 2009-2010
Sara Bertrán de Lis Mas

Supervisores:
María Arévalo Sánchez
Enrique Solano Márquez
Jesús Gallego Maestro
Jaime Zamorano Calvo

1. ÍNDICE

2. Introducción	3
2.1 La exploración UCM	3
2.2 Observatorio Virtual	4
3. Diseño y desarrollo del proyecto	5
3.1 Base de datos	5
3.2 Interfaz web	13
3.3 Servicio VO	19
4. Ejemplos de explotación científica	28
4.1 Muestra de LCBGs	28
4.2 Plano fundamental de galaxias con formación estelar	32
4.3 Estudio de la Exploración UCM	36
4.4 Trabajo con diferentes catálogos	39
5. Conclusiones y trabajo futuro	41
6. Bibliografía	42

2. INTRODUCCIÓN

En el mundo actual, donde las nuevas tecnologías han potenciado de forma extraordinaria las comunicaciones, se hace necesario el hecho de compartir información. La comunidad científica, en concreto el mundo de la astrofísica, no ajeno a estos cambios, ha puesto en marcha el proyecto Observatorio Virtual (VO).

La idea del presente trabajo surge de la voluntad de formar parte de dicho proyecto. Para ello se ha propuesto crear un servicio VO de la Exploración UCM de galaxias.

Este servicio va a permitir a la comunidad científica explotar nuestros datos de forma sencilla y eficiente. Esto es debido a que el astrónomo, como usuario de VO, realiza una única consulta, y es el sistema quien se encarga de buscar en todas las bases de datos asociadas y devolvernos el conjunto de resultados.

Así pues, con un servicio VO conseguimos una difusión máxima de nuestros datos, facilitando al investigador y a nosotros mismos el acceso a la información y su manipulación.

2.1 LA EXPLORACIÓN UCM

La exploración de la Universidad Complutense de Madrid (UCM) se inició con el objetivo de identificar y estudiar galaxias jóvenes con baja metalicidad, y así determinar las propiedades de la formación estelar en el Universo local.

En un principio se elaboraron dos listas con 264 candidatas en 17 campos diferentes, abarcando un total de 471.4 grados cuadrados, de las que finalmente sólo 191 han sido confirmadas como galaxias con líneas de emisión. (lista I: Zamorano et al. 1994 ApJS 95,387 y lista II Zamorano et al. 1996 ApJS 105,343).

El resultado de esta exploración es una muestra que reúne galaxias del Universo local ($z < 0.05$) con una actividad importante de formación estelar, con anchuras equivalentes de la línea de H α por encima de los 20Å. La Exploración incluye información relativa a

luminosidades, radios, tipo espectral, masas y otras propiedades físicas. Estos datos han sido obtenidos a partir de las observaciones realizadas con el telescopio 80/120 cm f/3 Schmidt del observatorio hispano-alemán de Calar Alto (Almería), por miembros del Departamento de Física de la Atmósfera, Astronomía y Astrofísica II de la Universidad Complutense de Madrid.

El resultado más relevante ha sido la primera determinación directa de la función de luminosidad en Halfa del Universo local y, a partir de ésta, de la densidad de tasa de formación estelar del Universo local.

2.2 OBSERVATORIO VIRTUAL

Los archivos astronómicos constituyen una herramienta básica para la astrofísica moderna ya que se hace uso continuado de ellos. Sin embargo, la eficiencia en la recuperación de datos se ve seriamente limitada por la falta de interoperabilidad entre ellos, sobre todo en la astronomía multifrecuencia. Por ello surge la necesidad de normalización de la información y la creación de unos protocolos para la extracción de datos.

El Observatorio Virtual es una iniciativa de colaboración internacional bajo los auspicios de la Alianza Internacional del Observatorio Virtual (IVOA), que surge como respuesta a este problema. Su principal objetivo es permitir el acceso electrónico global a los archivos astronómicos de las exploraciones de observatorios. También pretende facilitar técnicas de análisis de datos a través de una entidad de coordinación que provea estándares y herramientas de análisis de última generación.

Además otro importante objetivo es dar apoyo a la comunidad científica para utilizar la nueva infraestructura VO a través de la difusión, cursos y apoyo a los proyectos.

El proyecto español es *The Spanish Virtual Observatory* (SVO), que empezó en 2004 y cuyo equipo de desarrollo se encuentra en el Centro de Astrobiología (CAB), en Villafranca del Castillo. Es precisamente en este lugar donde se ha llevado a cabo el presente trabajo.

3. DISEÑO Y DESARROLLO DEL PROYECTO

Para crear un servicio VO necesitamos programar diferentes aplicaciones. En primer lugar debemos almacenar la información en una base de datos. A continuación crearemos un interfaz web para poder hacer consultas a esta base de datos desde cualquier lugar con acceso a Internet. Y finalmente, el servicio VO será también una aplicación de consulta a la base de datos, pero dicha consulta está preparada para ser realizada por otro programa en lugar de por un usuario. Esto último es lo que permitirá a una aplicación de VO acceder a nuestros datos.

3.1 BASE DE DATOS

La base de datos tiene que contener de forma ordenada aquella información que queremos que esté disponible para consultar. Dicha información se ordena en tablas. En nuestro caso hemos creado una base de datos relacional, lo cual implica que a través de las claves primarias y ajenas podemos relacionar los datos de diferentes tablas.

Nuestra base de datos ha sido construida a partir de una antigua base de datos, creada por Juan Carlos Muñoz Mateos para el trabajo académicamente dirigido “Base de Datos de la Exploración UCM”, entregado en el curso 2004-2005. Ésta ha ido sufriendo modificaciones a lo largo del tiempo: se le han añadido tablas con nuevos datos y otras se han modificado.

Así pues, en primer lugar se ha tenido que llevar a cabo una labor de estudio de la antigua base de datos y de selección de aquellas tablas cuyos datos estaban actualizados. Además se han incorporado otras tres tablas nuevas para la gestión de los archivos en formato fits de imágenes y espectros. Para ello se han recopilado las imágenes de la Exploración UCM en las diferentes bandas, y se ha comprobado que tuvieran astrometría y fotometría.

Por otra parte, para la nueva versión de la base de datos se ha escogido otro sistema gestor de bases de datos (SGBD): hemos pasado de MySQL a PostgreSQL. EL SGBD

es una interfaz de software que relaciona la base de datos física con los usuarios, y se encarga de tratar las peticiones de los usuarios. Se ha escogido PostgreSQL por diversos motivos: es de acceso gratuito, es compatible con Linux y es una herramienta muy potente para trabajar con grandes bases de datos. Así pues, toda la base de datos estará creada utilizando el lenguaje SQL (*structured query language*), que permite gestionar la información y especificar diversos tipos de operaciones, en las bases de datos relacionales.

Después del nuevo diseño, la base de datos actual consta de veinte tablas. A continuación mostramos una captura de pantalla de la lista de tablas:

```
[laeff-v2@weinberg ~]$ psql -h heisenberg-dev -U sara gucm
Password for user sara:
Welcome to psql 8.1.16 (server 8.1.3), the PostgreSQL interactive terminal.

gucom=> \d
          List of relations
Schema |      Name      | Type | Owner
-----+-----+-----+-----
public | archive_pag    | table | sara
public | cosmo          | table | sara
public | fits          | table | sara
public | fits_espectros | table | sara
public | gunn_r_alv    | table | sara
public | int_properties | table | sara
public | int_properties_f | table | sara
public | johnson_b_obs  | table | sara
public | johnson_b_pag  | table | sara
public | nir_pag       | table | sara
public | nombres       | table | sara
public | path          | table | sara
public | principal     | table | sara
public | schlegel_ext  | table | sara
public | schlegel_filters | table | sara
public | tables_       | table | sara
public | ucm_clave     | table | sara
public | ucm_espectros | table | sara
public | ucm_ha       | table | sara
public | ucm_sigma     | table | sara
(20 rows)
```

Cada un de estas tablas contiene la siguiente información:

Nombre tabla	Contenido
archive_pag	Datos de archivos: flujos IRAS, luminosidades en el FIR, flujos de 20cm y 21cm, masas dinámicas, magnitudes en UV, ect.
cosmo	Distancias calculadas a partir de NED y de los datos del survey UCM, distancias angulares.

ucm_sigma	Velocidades de dispersión.
fits	Ruta de acceso y nombres de los archivos de imágenes fits de cada galaxia en las bandas: U, B, R, gunn r, Ha, J, K.
fits_espectros	Ruta de acceso y nombres de los archivos de espectros fits de cada galaxia. En ocasiones hay más de un path por galaxia, correspondiente a diferentes partes de la misma.
gunn_r_alv	Datos fotométricos en gunn r tales como magnitudes, tipos morfológicos, radio, etc.
int_properties	Propiedades de las galaxias con formación estelar, como edad del brote de formación estelar, metalicidad, masa total de estrellas, SFR, etc.
int_properties_f	Contiene los mismos campos que la tabla int_properties, pero en ésta se agrupan aquellos datos que no se ajustan a los modelos.
johnson_b_obs	Datos observacionales en Johnson B: tiempo de exposición, seeing, escala de placa, parámetros de calibración, nivel de cielo, etc.
johnson_b_pag	Datos fotométricos en Johnson B: magnitud, tipo morfológico, radio, índice de color B-r, etc.
nir_pag	Magnitudes en infrarrojo cercano, en las bandas J y K.
nombres	Nombres de cada galaxia en los catálogos GALEX y SDSS, así como número de placa para cada una de ellas.
path	Directorios de las carpetas de imágenes y espectros.
principal	Nombres UCM y coordenadas ecuatoriales.
schlegel_ext	Extinciones galácticas en diferentes bandas, calculadas a partir de los mapas de extinción de Schlegel et al (1998) y las curvas de extinción con $R_v=3.1$ de Cardelli et al. (1989) y O'Donnell (1994).
schlegel_filters	Parámetros de extinción para diferentes bandas de los mapas de Schlegel et al. utilizando las curvas de extinción con $R_v=3.1$ de Cardelli et al. (1989) y O'Donnell (1994).
tables_	Explicación del contenido de las tablas de la base de datos.
ucm_clave	Explicación de las variables de cada una de las tablas de la base de datos.

ucm_espectros:	Datos de los espectros: intensidades de diferentes líneas, anchuras equivalentes, índices de color, tipo espectral, metalicidades, ect.
ucm_ha:	Datos de H alpha: luminosidades, anchuras equivalentes, flujos, ect.

La creación de cada tabla requiere una serie de sentencias en lenguaje SQL, como las que se muestran a continuación:

```
--
-- Table structure for table Schlegel_filters
--
CREATE TABLE Schlegel_filters (
  filter varchar(15) NOT NULL default '',
  Leff double precision default NULL,
  A_Av double precision default NULL,
  A_EBV double precision default NULL,
  PRIMARY KEY (filter)
) ;

--
-- Dumping data for table Schlegel_filters
--
INSERT INTO Schlegel_filters VALUES('Landolt_U', 3372, 1.664, 5.434);
INSERT INTO Schlegel_filters VALUES('Landolt_B', 4404, 1.321, 4.315);
INSERT INTO Schlegel_filters VALUES('Landolt_V', 5428, 1.015, 3.315);
INSERT INTO Schlegel_filters VALUES('Landolt_R', 6509, 0.819, 2.673);
INSERT INTO Schlegel_filters VALUES('Landolt_I', 8090, 0.594, 1.94);
INSERT INTO Schlegel_filters VALUES('CTIO_U', 3683, 1.521, 4.968);
INSERT INTO Schlegel_filters VALUES('CTIO_B', 4393, 1.324, 4.325);
INSERT INTO Schlegel_filters VALUES('CTIO_V', 5519, 0.992, 3.24);
INSERT INTO Schlegel_filters VALUES('CTIO_R', 6602, 0.807, 2.634);
INSERT INTO Schlegel_filters VALUES('CTIO_I', 8046, 0.601, 1.962);
INSERT INTO Schlegel_filters VALUES('UKIRT_J', 12660, 0.276, 0.902);
INSERT INTO Schlegel_filters VALUES('UKIRT_H', 16732, 0.176, 0.576);
INSERT INTO Schlegel_filters VALUES('UKIRT_K', 22152, 0.112, 0.367);
INSERT INTO Schlegel_filters VALUES('UKIRT_L', 38079, 0.047, 0.153);
INSERT INTO Schlegel_filters VALUES('Gunn_g', 5244, 1.065, 3.476);
INSERT INTO Schlegel_filters VALUES('Gunn_r', 6707, 0.793, 2.59);
INSERT INTO Schlegel_filters VALUES('Gunn_i', 7985, 0.61, 1.991);
INSERT INTO Schlegel_filters VALUES('Gunn_z', 9055, 0.472, 1.54);
INSERT INTO Schlegel_filters VALUES('SpinRad_RS', 6993, 0.755, 2.467);
INSERT INTO Schlegel_filters VALUES('APM_bj', 4690, 1.236, 4.035);
INSERT INTO Schlegel_filters VALUES('Stromgren_u', 3502, 1.602, 5.231);
INSERT INTO Schlegel_filters VALUES('Stromgren_b', 4676, 1.24, 4.049);
INSERT INTO Schlegel_filters VALUES('Stromgren_v', 4127, 1.394, 4.552);
INSERT INTO Schlegel_filters VALUES('Stromgren_beta', 4861, 1.182, 3.858);
INSERT INTO Schlegel_filters VALUES('Stromgren_y', 5479, 1.004, 3.277);
INSERT INTO Schlegel_filters VALUES('Sloan_u', 3546, 1.579, 5.155);
INSERT INTO Schlegel_filters VALUES('Sloan_g', 4925, 1.161, 3.793);
INSERT INTO Schlegel_filters VALUES('Sloan_r', 6335, 0.843, 2.751);
INSERT INTO Schlegel_filters VALUES('Sloan_i', 7799, 0.639, 2.086);
INSERT INTO Schlegel_filters VALUES('Sloan_z', 9294, 0.453, 1.479);
INSERT INTO Schlegel_filters VALUES('WFPC2_F300W', 3047, 1.791, 5.849);
INSERT INTO Schlegel_filters VALUES('WFPC2_F450W', 4711, 1.229, 4.015);
INSERT INTO Schlegel_filters VALUES('WFPC2_F555W', 5498, 0.996, 3.252);
INSERT INTO Schlegel_filters VALUES('WFPC2_F606W', 6042, 0.885, 2.889);
INSERT INTO Schlegel_filters VALUES('WFPC2_F702W', 7068, 0.746, 2.435);
INSERT INTO Schlegel_filters VALUES('WFPC2_F814W', 8066, 0.597, 1.948);
INSERT INTO Schlegel_filters VALUES('DSS_II_g', 4814, 1.197, 3.907);
INSERT INTO Schlegel_filters VALUES('DSS_II_r', 6571, 0.811, 2.649);
INSERT INTO Schlegel_filters VALUES('DSS_II_i', 8183, 0.58, 1.893);
```


En la imagen superior vemos en primer lugar la creación de la tabla “fits”. En la parte inferior vemos las órdenes correspondientes para rellenar esta tabla. Cada fila contiene primero el nombre de la galaxia y después el directorio de la imagen en cada uno de los diferentes filtros. Este directorio está constituido primero por el nombre de la carpeta de imágenes de la banda y seguido del nombre de archivo. Por ejemplo “Images_U/b0000+2140.fits”. Así sabemos que la carpeta que contiene las imágenes en banda Johnson U se llama Images_U y los archivos tienen en primer lugar una letra minúscula referente al filtro, a continuación el nombre ucm (sin ucm delante) y después la extensión “.fits”

En el caso de los espectros, como no tenemos carpetas con diferentes filtros, solamente ponemos el nombre del archivo, como podemos ver a continuación en el código correspondiente a la creación de la tabla “fits_espectros”:

```
CREATE TABLE fits_espectros (
  ucname varchar(10) NOT NULL default '',
  spct_total varchar(30) default NULL,
  spct_1 varchar(30) default NULL,
  spct_2 varchar(30) default NULL,
  spct_3 varchar(30) default NULL,
  spct_4 varchar(30) default NULL,
  spct_5 varchar(30) default NULL,
  spct_6 varchar(30) default NULL,
  spct_7 varchar(30) default NULL,
  spct_2D varchar(30) default NULL,
  PRIMARY KEY (ucname)
);

INSERT INTO fits_espectros VALUES ('0000+2140', 'UCM0000+2140E.fits', NULL, NULL, NULL, NI
INSERT INTO fits_espectros VALUES ('0003+1955', 'UCM0003+1955E.fits', NULL, NULL, NULL, NI
INSERT INTO fits_espectros VALUES ('0003+2200', 'UCM0003+2200E.fits', NULL, NULL, NULL, NI
INSERT INTO fits_espectros VALUES ('0003+2215', 'UCM0003+2215E.fits', NULL, NULL, NULL, NI
INSERT INTO fits_espectros VALUES ('0005+1802', 'UCM0005+1802P.fits', NULL, NULL, NULL, NI
INSERT INTO fits_espectros VALUES ('0006+2332', 'UCM0006+2332E.fits', NULL, NULL, NULL, NI
```

Cuando queramos obtener la dirección completa de un archivo, tendremos que concatenar la dirección de la tabla “path” con la que nos da la tabla “fits” o “fits espectros”, según queramos imagen o espectro.

Para rellenar estas tablas con los directorios y nombres de archivos se ha utilizado una aplicación Java, creada por el Observatorio Virtual Español (SVO) y modificada para el caso que nos ocupa. Esta aplicación lee un archivo de texto que creamos con los nombres ucm de las galaxias (sin el ucm delante) y a partir de él nos devuelve por

pantalla el comando para crear cada una de las filas de la tabla. Esta aplicación Java consta de dos clases llamadas `SacaDatos.java` e `Insertator.java`:

```

Insertator.java X
import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.sql.SQLException;

public class Insertator {

    public Insertator() {
    }

    public void insertSentences(FileReader fr, FileWriter fw) {
        try {
            BufferedReader bf = new BufferedReader(fr);
            String linea="";
            int i =0;

            BufferedWriter buf =new BufferedWriter(fw);
            //linea = bf.readLine(); // primera linea son cabeceras
            while((linea=bf.readLine())!=null){

                String insert = "";

                // INSERT INTO

                insert = "INSERT INTO fits_espectros VALUES " +
                    " (' + linea + ', 'UCM" + linea + ".fits', NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL);\n";

                buf.write(insert);
                System.out.print(insert);
                //if (i>2) break;
                i++;
            }
            buf.close();
            fw.close();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}

```

Esta clase crea y devuelve por pantalla, por cada línea leída de un archivo de texto, una nueva línea llamada “insert”, que contiene la sentencia SQL para crear una nueva fila de una tabla. “Insert” está formada por texto y también por una variable llamada “línea”, que contiene cada línea de texto leída del archivo.

La clase `SacaDatos` contiene la función `main`, que es la que va a ser ejecutada. Instancia un objeto de la clase `Insertator`, descrita anteriormente, y ejecuta su método `insertSentences` para construir las sentencias SQL. Su código se muestra a continuación:

```
SacaDatos.java ✕
import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.sql.SQLException;

public class SacaDatos {

    public static void main(String[] args) throws FileNotFoundException, IOException, SQLException {

        //Leemos el fichero
        FileReader fr = new FileReader(args[0]);

        FileWriter fw = new FileWriter(args[1]);
        Insertator insert = new Insertator();

        insert.insertSentences(fr, fw);

    }
}
```

Así, con estas dos clases conseguimos de forma muy sencilla crear las sentencias SQL para las tablas de imágenes y espectros, sin lugar a posibles errores de lectura de los nombres de archivo.

En total nuestra base de datos ha necesitado un total de 3982 líneas de código en lenguaje SQL para ser creada. La base de datos completa, así como el resto de archivos que constituyen este trabajo, se adjuntan en un CD con esta memoria.

3.2 INTERFAZ WEB

Una vez construida la base de datos, tenemos que hacer una aplicación web de consulta a la misma. Esta aplicación sirve para que los usuarios puedan consultar directamente la base de datos.

Esta aplicación web ha sido creada utilizando lenguaje HTML (*HyperText Markup Languaje*) para definir la estructura de la página y JavaScript y JSP (*JavaServer Pages*), que permiten crear páginas web dinámicas incorporando lenguaje Java. Se ha utilizando un servidor Apache Tomcat, que es un servidor web con soporte para JSP.

Hemos creado dos archivos: inicio.jsp y resultado.jsp. El primero de ellos contiene el formulario de inicio, cuyo resultado visible para el usuario tiene el siguiente aspecto:



ACCESS TO THE UCM SURVEY

Enter object name or coordinates:

Object name (nnnn+nnnn):	<input type="text"/>
RA (deg):	<input type="text"/>
DEC (deg):	<input type="text"/>
Radius (deg):	<input type="text" value="0.5"/>
All galaxies:	<input type="checkbox"/>

General Information:

Coordinates:	<input type="checkbox"/>
Other names:	<input type="checkbox"/>

Near-infrared Photometry:

J:	<input type="checkbox"/>
K:	<input type="checkbox"/>

Optical Spectroscopy:

Line intensities:	<input type="checkbox"/>
Equivalent widths:	<input type="checkbox"/>
Fluxes:	<input type="checkbox"/>
Other data:	<input type="checkbox"/>

Optical Photometry:

Johnson B:	<input type="checkbox"/>
Gunn r:	<input type="checkbox"/>
Color indexes:	<input type="checkbox"/>
H α :	<input type="checkbox"/>

Galactic extinction:

Landolt:	<input type="checkbox"/>
UKIRT:	<input type="checkbox"/>
Gunn:	<input type="checkbox"/>

Derived properties:

Abundances:	<input type="checkbox"/>
Stellar population models:	<input type="checkbox"/>
Cosmology:	<input type="checkbox"/>

mark/unmark all:

Como se puede apreciar, este formulario nos permite hacer una búsqueda bien por nombre, utilizando el nombre ucm pero sin el ‘ucm’ delante, o bien por coordenadas. También existe la opción de que devuelva todas las galaxias. Además, podemos escoger qué datos de la base de datos queremos que nos sean devueltos.

Este formulario, así como la cabecera de la página se han creado en lenguaje HTML. Los archivos son de tipo JSP.

En segundo lugar, el archivo resultado.jsp lleva a cabo la conexión con la base de datos, la consulta y la presentación de los resultados. Este proceso tiene diferentes fases.

La primera de ellas es la definición de variables. Aquí recogemos los parámetros que se han introducido en la página de inicio: nombre, coordenadas y radio. Se comprueba además si alguno de ellos es nulo y en ese caso se le asigna un valor por defecto. Estas variables son, por el momento, de tipo *string* (texto).

```
//=====
//=  DEFINICION DE VARIABLES
//=====

////////////////////////////////////
String fra = request.getParameter("coor_ra");
if (fra == (null) || fra == ("") || fra.length() == 0) fra = "";

////////////////////////////////////
String fdec = request.getParameter("coor_dec");
if (fdec == (null) || fdec == ("") || fdec.length() == 0) fdec = "";

////////////////////////////////////
String frad = request.getParameter("coor_rad");
if (frad == (null) || frad == ("") || frad.length() == 0) frad = "";

////////////////////////////////////
String name = request.getParameter("id");
if (name == (null) || name == ("") || name.length() == 0) name = "";
```

La segunda fase es el control de errores. Aquí creamos los parámetros de ascensión recta (ra), declinación (dec) y radio de búsqueda (rad), y les asignamos un valor nulo por defecto. En caso de no haber introducido ningún valor de entrada, no haber seleccionado la opción “*All galaxies*”, o bien haber introducido de forma incorrecta el formato de las coordenadas, de forma que no se puedan transformar a variables numéricas, entonces automáticamente aparece un mensaje de error y un enlace a la página de inicio. Veamos el código:

```
//=====
//= CONTROL DE ERRORES
//=====

String err = "";

Double ra=0.0;
Double dec=0.0;
Double rad=0.0;

String check0 = request.getParameter("all");

try{
    ra = new Double(fra);
    dec = new Double(fdec);
    rad = new Double(frad);
}catch(NumberFormatException e){
    err="Formato incorrecto de coordenadas";
}

if(err=="Formato incorrecto de coordenadas" && name==" && check0==null){
    %<b><br><font size=3 color="royalblue"> Please, set search conditions <A HREF="inicio.jsp" >here</A></font></b><%;
}else{
```

La tercera fase es la conexión con la base de datos, que se realiza proporcionando el nombre de la base de datos, usuario, contraseña y *host*. Además se incluye en el código una aclaración sobre la construcción de las consultas a la base de datos, o *queries*:

```
//=====
// CONEXION CON LA BASE DE DATOS
//=====

String host = "heisenberg-dev";
String dbName = "gucm";
String userid = "sara";
String passwd = "sara";

Connection conexion = null; //interface para la connexion con las BD.

String bdurl = "jdbc:postgresql://" + host + ":5432/" + dbName ;

try {
    Class.forName("org.postgresql.Driver");
    conexion = DriverManager.getConnection(bdurl, userid, passwd );
} catch (SQLException errorconexion) {
    errorconexion.printStackTrace();
}

//Construcción de la Query. Las variables incógnita, marcadas como ? en las query, serán siempre:
//Var. 1 = declinación
//Var. 2 = ascensión recta
//Var. 3 = decclinación
//Var. 4 = radio del cono de búsqueda
//Var. 5 = ucmmname (sin ucm delante, sólo números)

//La fórmula que aparece en la query, en la parte de WHERE sirve para considerar siempre un mismo área
//de búsqueda independientemente de la declinación. Se ha utilizado porque nuestro catálogo es pequeño.
```

Una vez completados estos pasos, la última fase es el cuerpo de las tablas de resultados. Vamos a mostrar el ejemplo de la primera tabla, “*Coordinates*”.

En primer lugar se comprueba si en la página de inicio se ha marcado el *checkbox* para que se muestren estos resultados, en caso afirmativo se continúa. El siguiente paso es hacer la cabecera de la tabla de resultados, es decir la primera fila de la tabla con los nombres de los campos. Esto se hace con lenguaje HTML:

```
//=====
// TABLA COORDENADAS:
//=====

String check1 = request.getParameter("coor"); //Se recoge la información sobre el estado del checkbox de coordenadas
if(check1!=null && check1.length()>0){ //Si no es nulo, se procede.

    //Construimos la cabecera de la tabla:
    <br><br><b><font size=3 color="royalblue"> Coordinates: </font></b><br><br>

    <table width="500" border="1" cellspacing="0" cellpadding="3">
    <tr>
        <td><i>UCM name</i></td>
        <td><i>RA (deg)</i></td>
        <td><i>DEC (deg)</i></td>
        <td><i>RA (hh mm ss.s)</i></td>
        <td><i>DEC (dd mm ss)</i></td>
    </tr>
    <%
```

A continuación se construye la *query*. Ésta se crea como una variable de tipo texto (*string*) que a continuación se ejecutará. Se hace en dos pasos diferentes, en el primero se hace la parte inicial de la *query*: “SELECT variable1, variable2 FROM NombreTabla”.

El segundo paso dependerá de si hemos seleccionado la opción “All galaxies”, en cuyo caso se cierra la *query* con punto y coma sin agregar nada más, o bien si hemos introducido datos para una galaxia en concreto. En ese caso la *query* se completa con una parte que impone una condición, “WHERE”. Esta condición tiene dos partes separadas con un OR, la primera correspondiente a coordenadas, la segunda al identificador. En la primera tenemos una fórmula que selecciona aquellas galaxias cuyas coordenadas se encuentren dentro del radio escogido, corrigiendo efectos de declinación en el área definida por nuestro radio. Es decir, el área de búsqueda será la misma estemos en el ecuador o en el polo. La segunda parte del OR selecciona aquellas galaxias cuyo identificador coincida exactamente con el escrito en el formulario de inicio.

Al utilizar un *OR* garantizamos que si se han introducido a la vez coordenadas e identificador, y no coinciden, saldrán ambos resultados.

```

//Construimos la query:
String query = "SELECT ucname, ra_deg, dec_deg, RA, DECL FROM principal p ";
PreparedStatement statement = null;

if(check0!=null && check0.length()>0){
    //Si el checkbox de todas las galaxias está marcado
    query += ";";
    statement = conexion.prepareStatement(query);
}
else{
    //Si el checkbox de todas las galaxias no está marcado
    query += "WHERE "+
        " (acos( cos(radians(p.dec_deg))*cos(radians( ? ))*cos(radians(p.ra_deg - ? )) + " +
        " sin(radians(p.dec_deg))*sin(radians( ? )) ) < radians( ? ) OR (p.ucname=?));";

    statement = conexion.prepareStatement(query);

    //Asignamos valores a las incógnitas de la query
    statement.setDouble(1,dec.doubleValue());
    statement.setDouble(2,ra.doubleValue());
    statement.setDouble(3,dec.doubleValue());
    statement.setDouble(4,rad.doubleValue());
    statement.setString(5, name);
}

```

Después de esto ejecutamos la *query* y almacenamos los resultados:

```

//Se ejecuta la query:
ResultSet resultados = statement.executeQuery();

```

Luego creamos las variables de los resultados, y les asignamos el valor correspondiente del resultado de la *query*. Una vez hecho esto podemos acabar de completar la tabla HTML con los resultados. El proceso a partir de la ejecución de la *query* está incluido dentro de un bucle *while*, de tal forma que si para nuestra búsqueda hay más de un resultado, cada uno de ellos se irá mostrando en una fila diferente de la tabla.

```

//Recogemos los resultados de la consulta y los asignamos a la variable correspondiente:
while (resultados.next()) {
    String ucname      = resultados.getString("ucname");

    String RA          = resultados.getString("RA");

    String DECL        = resultados.getString("DECL");

    Double ra_deg      = new Double(resultados.getDouble("ra_deg"));
    if(resultados.isNull()) ra_deg=null;

    Double dec_deg     = new Double(resultados.getDouble("dec_deg"));
    if(resultados.isNull()) dec_deg=null;

    //Construcción del resto de la tabla de resultados. Esta tabla tendrá tantas filas como galaxias haya dentro
    //del radio.
    //La orden <%= ... > imprime en pantalla el valor de la variable que se escriba en ...

    <tr>
        <td><%=ucname %></td>
        <td><%=ra_deg %></td>
        <td><%=dec_deg %></td>
        <td><%=RA %></td>
        <td><%=DECL %></td>
    </tr>
}

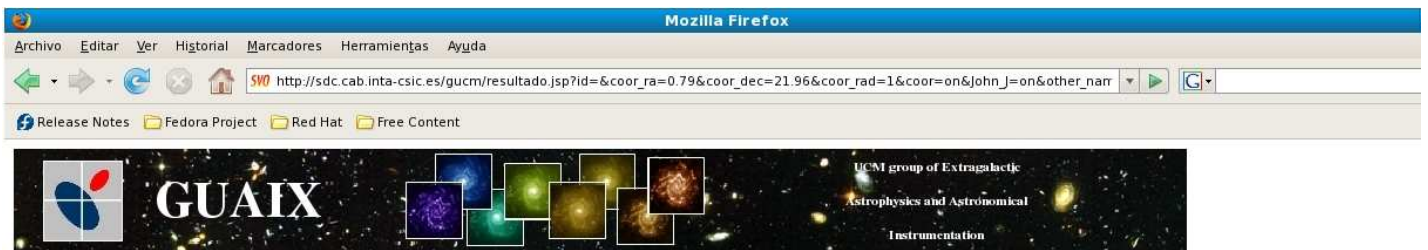
```

Por último agregamos comentarios al pie de la tabla, tales como las referencias con un enlace al artículo, o bien alguna aclaración sobre una variable en particular:

```
//Cerramos la tabla y ponemos al final las referencias:
</table>

<font color="grey"> Referencias: <A href="http://adsabs.harvard.edu/abs/1994ApJS...95...387Z">1994ApJS...95...387Z</A>
}
```

Este proceso se repite de forma muy similar en cada una de las tablas de resultados. Para ilustrar lo que el usuario recibe al realizar una búsqueda, mostramos a continuación la respuesta de una búsqueda cuyo resultado son tres galaxias:



RESULTS TABLES:

Coordinates:

UCM name	RA (deg)	DEC (deg)	RA (hh mm ss.s)	DEC (dd mm ss)
0000+2140	0.790417	21.9608	00 03 09.7	+21 57 39
0003+2200	1.40833	22.2825	00 05 38.0	+22 16 57
0003+2215	1.46792	22.5358	00 05 52.3	+22 32 09

References: [1994ApJS...95...387Z](#), [1996ApJS...105...343Z](#)

Other names:

UCM name	RA (deg)	DEC (deg)	Galex	Sloan
0000+2140	0.790417	21.9608	null	null
0003+2200	1.40833	22.2825	null	null
0003+2215	1.46792	22.5358	null	null

Line intensities (relative to H_beta=100):

UCM name	RA (deg)	DEC (deg)	[OII] 3727	Hγ 4340	Hβ 4861	[OIII] 4959	[OIII] 5007	HeI 5876	[OI] 6300	[NII] 6548	Hα 6563	[NII] 6584	[SII] 6716	[SII] 6731	Hα/Hβ (c)
0003+2200	1.40833	22.2825	null	100.0	null	null	180.0	null	null	null	740.0	244.0	null	null	502.0
0003+2215	1.46792	22.5358	null	100.0	null	null	null	null	null	null	863.0	433.0	90.0	76.0	562.0
0000+2140	0.790417	21.9608	27.0	100.0	159.0	66.0	211.0	null	46.0	null	879.0	359.0	95.0	78.0	755.0

(c): corrected for stellar absorption

Reference: [1996A&AS...120...323G](#)

La URL de este interfaz web es: <http://sdc.cab.inta-csic.es/gucm/>

3.3 SERVICIO VO

El servicio de VO es también, como la anterior página web, un servicio de consulta a la base de datos. La diferencia es que la página web está hecha para ser consultada por usuarios, y el servicio VO está hecho fundamentalmente para ser consultado automáticamente.

El protocolo para obtener los datos se denomina *ConeSearch*, ya que a partir de unas coordenadas y una distancia angular definimos un cono en el espacio. Los objetos dentro de dicho cono nos serán devueltos como resultados de la búsqueda.

Este protocolo proporciona ciertos requisitos para nuestro servicio. Los datos de entrada son la ascensión recta, declinación y una distancia angular, que se obtienen desde la URL (*HTTP GET Request*) y con los que se define el cono en el espacio. La respuesta de la búsqueda tiene que ser dada en formato VOTable, que es un estándar en XML (*Extensible Markup Language*) para la representación de tablas.

Una VOTable tiene que incluir los parámetros de ascensión recta, declinación e identificador. Además la tabla puede contener otros campos, que serán especificados mediante su UCD, con una pequeña descripción y tipo de variable que contienen.

Las iniciales UCD corresponden a *Unified Content Descriptors*. Sirven para identificar de forma estandarizada los parámetros que vamos a utilizar, y se crean uniendo palabras según unas normas de construcción, que aparecen descritas en la página web de IVOA (ver bibliografía).

En caso de no haber datos de entrada, o bien éstos sean dados con un formato incorrecto, la respuesta debe ser una VOTable con un único parámetro de error que especifique la naturaleza del mismo.

El esquema del proyecto es el siguiente:



Vemos que en la parte inferior aparecen los archivos inicio.jsp y resultado.jsp correspondientes a la aplicación web. Éstos no tienen ninguna relación con el servicio VO y sin embargo, los hemos creado por comodidad en el mismo proyecto de Eclipse, que es el entorno de desarrollo integrado que se ha utilizado para realizar este proyecto.

Para crear el servicio VO hemos utilizado Java y JSP. A continuación iremos comentando cada uno de los archivos que conforman esta aplicación.

VOObject.java recoge los parámetros de entrada, que son ascensión recta, declinación y radio (RA, DEC y SR). En el caso de un servicio VO los datos no son introducidos manualmente, sino que se obtienen desde la URL, que debe estar compuesta por las siguientes partes:

`http://<server-address>/<path>? [<extra-GET-arg>&[...]]`

`http://mycone.org/cgi-bin/search?RA=180.567&DEC=-30.45&SR=0.0125`

Recogemos así los datos y con ellos se realiza un control de errores, se convierten a tipo numérico y se almacenan. El código Java se muestra a continuación. Esta clase tiene además accesores (*get*) y mutadores (*set*) para estas variables, que no mostramos en la imagen inferior.

```

access.java  *VOObject.java  *metadata.jsp  vosearch.jsp
public VOObject (HttpServletRequest request){

    if (request.getParameter("RA").trim().length()==0 || request.getParameter("RA")== null){
        ra = null;
    }else{
    try{

        //String regexp ="([0-9]+(?:\.[0-9]+){0,1})";

        //ra = Double.valueOf(request.getParameter("RA").trim().replaceAll(regexp, "$1")).doubleValue();
        ra = Double.valueOf(request.getParameter("RA").trim());

    }catch(Exception e){
        //No se puede
        ra = null;
    }
    }

    if (request.getParameter("DEC").trim().length()==0 || request.getParameter("DEC")== null){
        dec = null;
    }else{
    try{

        //String regexp ="([+-]{0,1}[0-9]+(?:\.[0-9]+){0,1})";

        //dec = Double.valueOf(request.getParameter("DEC").trim().replaceAll(regexp, "$1")).doubleValue();
        dec = Double.valueOf(request.getParameter("DEC").trim());

    }catch(Exception e){
        //No se puede
        dec = null;
    }
    }

    if (request.getParameter("SR").trim().length()==0 || request.getParameter("SR")== null){
        sr = null;
    }else{
    sr=Double.valueOf(request.getParameter("SR").trim());
    }
}

```

En **ResultVO.java** se han creado las variables para almacenar el resultado de la búsqueda en la base de datos. Esta clase contiene además accesores (*get*) y mutadores (*set*) a todas las estas variables.

```

ResultVO.java
package classes;

public class ResultVO {

    String id = null;
    Double ra = null;
    Double dec = null;

    String Sp_type = null;
    Double L_Ha_spe = null;
    Double Mb_pag = null;
    Double Mb_pag_e = null;
    Double EW_Ha = null;
    Double b_ae_iso_kpc = null;
    Double b_ae_iso_kpc_e = null;
    Double log_SFR_M = null;
    Double log_SFR_M_e = null;
    Double log_M = null;
    Double log_M_e = null;
    Double oh_o = null;
    Double I_3727_c = null;

    public ResultVO (){}
}

/**
 * @return the ra
 */
public Double getRa() {
    return ra;
}

/**
 * @param ra the ra to set
 */
public void setRa(Double ra) {
    this.ra = ra;
}
}

```

UCMAccess.java accede a la base de datos de una forma muy similar a como accedíamos en la aplicación web, pero en este caso la *query* es única y pide de una sola vez todos los parámetros que queremos que devuelva el servicio VO, para aquellas galaxias que se ajusten a los datos de entrada: coordenadas y radio. Además, en lugar de presentar los datos en una tabla HTML se almacenan utilizando la clase ResultVO.

vosearch.jsp es el cuerpo del Servicio VO y un archivo JSP. En primer lugar se lleva a cabo un control de errores de tal forma que redirecciona a una página de error, error.jsp, si alguno de los datos de entrada (ascensión recta, declinación o radio) son nulos. A continuación hay un segundo control de errores si los datos de coordenadas se salen de


```

if (parametrosFormulario.getRA() != null){
  <PARAM name="INPUT:RA" ucd="POS_EQ_RA_MAIN" value="<%=parametrosFormulario.getRA()%>" datatype="double" arraysize="*">
  <DESCRIPTION>
  A right-ascension in the ICRS coordinate system for the position of the center of the cone to search,
  given in decimal degrees.
  </DESCRIPTION>
  </PARAM>
}

if(parametrosFormulario.getDEC() != null){
  <PARAM name="INPUT:DEC" ucd="POS_EQ_DEC_MAIN" value="<%=parametrosFormulario.getDEC()%>" datatype="double" arraysize="*">
  <DESCRIPTION>
  A declination in the ICRS coordinate system for the position of the center of the cone to search,
  given in decimal degrees.
  </DESCRIPTION>
  </PARAM>
}

if(parametrosFormulario.getSR() != null){
  <PARAM name="INPUT:SR" ucd="OBS_ANG-SIZE" value="<%=parametrosFormulario.getSR()%>" datatype="double" arraysize="*">
  <DESCRIPTION>
  The radius of the cone to search, given in decimal degrees.
  </DESCRIPTION>
  </PARAM>
}

```

Vemos que por cada parámetro aparece una breve descripción y además se asocia su correspondiente UCD.

A continuación presentamos en una tabla los UCDs que nosotros hemos utilizado, así como una descripción de los mismos:

UCD	Descripción
ID_MAIN	IAU-compatible name of source
POS_EQ_RA_MAIN	Right Ascension
POS_EQ_DEC_MAIN	Declination
src.spType	Spectroscopic type
phys.luminosity;em.line.Halpha	H α luminosity
phys.magAbs;em.opt.B	Johnson B absolute magnitude
stat.error;phys.magAbs;em.opt.B	Uncertainty in Johnson B absolute magnitude
spect.line.eqWidth;em.line.Halpha	H α equivalent width

phys.size.radius;em.opt.B	Johnson B linear effective radius (isophotal)
stat.error;phys.size.radius;em.opt.B	Uncertainty in Johnson B linear effective radius (isophotal)
phys.SFR	Logarithm of specific star formation rate (SFR per unit of total stellar mass)
stat.error;phys.SFR	Uncertainty in logarithm of specific star formation rate (SFR per unit of total stellar mass)
phys.mass	Logarithm of the total stellar mass
stat.error;phys.mass	Uncertainty in logarithm of the total stellar mass
phys.abund.Z;em.line.OIII	Metallicity derived from [OIII]5007/H β
spect.line.intensity;em.line.OII	[OII]3727 extinction corrected intensity relative to H β corrected, H β _c=100

Continuamos con el archivo vosearch.jsp. Por último se construye la cabecera de la VOTable, donde se da el nombre del campo, una breve descripción y el correspondiente UCD. Después se rellenan las filas de la tabla con los resultados de la búsqueda, de la misma forma que lo hacíamos en la aplicación web.

Se muestra a continuación la cabecera de la tabla, escrita en formato XML:

```
<TABLE id="UCM Survey">
<DESCRIPTION> </DESCRIPTION>

<FIELD name="unique_id" datatype="char" arraysize="*" ucd="ID_MAIN">
<DESCRIPTION> IAU-compatible name of source </DESCRIPTION>
</FIELD>

<FIELD name="ra" datatype="double" unit="degree" ucd="POS_EQ_RA_MAIN">
<DESCRIPTION> Right Ascension </DESCRIPTION>
</FIELD>

<FIELD name="dec" datatype="double" unit="degree" ucd="POS_EQ_DEC_MAIN">
<DESCRIPTION> Declination </DESCRIPTION>
</FIELD>

<FIELD name="Sp_type" datatype="char" arraysize="*" ucd="src.spType">
<DESCRIPTION> Spectroscopic type </DESCRIPTION>
</FIELD>

<FIELD name="L_Ha_spe" datatype="double" unit="10e8 Lsun" ucd="phys.luminosity;em.line.Halpha">
<DESCRIPTION> Halpha luminosity </DESCRIPTION>
</FIELD>
```

```

<FIELD name="Mb_pag" datatype="double" unit="magnitudes" ucd="phys.magAbs;em.opt.B">
<DESCRIPTION> Johnson B absolute magnitude </DESCRIPTION>
</FIELD>

<FIELD name="Mb_pag_e" datatype="double" unit="magnitudes" ucd="stat.error;phys.magAbs;em.opt.B">
<DESCRIPTION> Uncertainty in Johnson B absolute magnitude </DESCRIPTION>
</FIELD>

<FIELD name="EW_Ha" datatype="double" unit="Angstroms" ucd="spect.line.eqwidth;em.line.Halpha">
<DESCRIPTION> Halpha equivalent width </DESCRIPTION>
</FIELD>

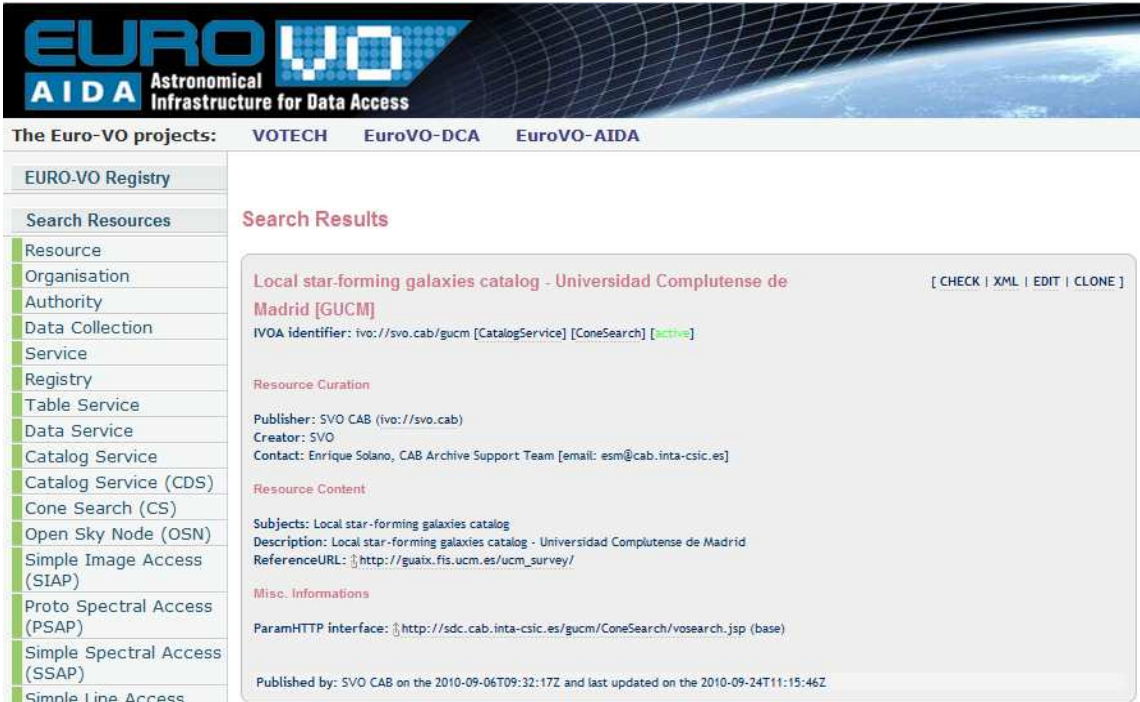
<FIELD name="b_ae_iso_kpc" datatype="double" unit="kpc" ucd="phys.size.radius;em.opt.B">
<DESCRIPTION> Johnson B linear effective radius (isophotal) </DESCRIPTION>
</FIELD>

<FIELD name="b_ae_iso_kpc_e" datatype="double" unit="kpc" ucd="stat.error;phys.size.radius;em.opt.B">
<DESCRIPTION> Uncertainty in Johnson B linear effective radius (isophotal) </DESCRIPTION>
</FIELD>

<FIELD name="log_SFR_M" datatype="double" unit="log(10e-11 y^-1 Msun^-1)" ucd="phys.SFR">
<DESCRIPTION> Logarithm of specific star formation rate (SFR per unit of total stellar mass) </DESCRIPTION>
</FIELD>

```

Una vez implementado este servicio, debemos registrarlo en EURO-VO para que entre en funcionamiento. Para ello debemos dar diferentes datos, como el nombre del servicio: *Local star-forming galaxies catalog – Universidad Complutense de Madrid*. También datos sobre el creador del servicio: *Spanish Virtual Observatory (SVO)*, cuya persona de contacto es Enrique Solano (*CAB Archive Support Team*), una URL de referencia que será la página web de GUAIX, y por último la página para hacer la consulta VO, que es precisamente la dirección de nuestro archivo vosearch.jsp: <http://sdc.cab.inta-csic.es/gucm/ConeSearch/vosearch.jsp> (*ParamHTTP Interface*)



The screenshot shows the EURO-VO AIDA website interface. At the top, there is a navigation bar with the text "The Euro-VO projects: VOTECH EuroVO-DCA EuroVO-AIDA". Below this is a sidebar menu with various search resources like "Resource", "Organisation", "Authority", etc. The main content area displays "Search Results" for the "Local star-forming galaxies catalog - Universidad Complutense de Madrid [GUCM]". It includes the IVOA identifier, publisher information (SVO CAB), creator (SVO), contact details (Enrique Solano), and a description of the catalog. A "ParamHTTP interface" URL is also provided.

Además de estos datos, se pueden incluir más detalles sobre el servicio. A partir de ese momento nuestro catálogo aparecerá en las consultas que hagamos utilizando el Observatorio Virtual, junto con otra gran cantidad de catálogos. De esta manera nos será más sencillo comparar nuestros datos con los de otras exploraciones, podremos utilizar las herramientas VO y lo que es más importante, nuestros archivos estarán disponibles para el resto de la comunidad científica de manera muy accesible.

En el siguiente capítulo explicaremos un caso práctico en el que explotamos las ventajas de construir un servicio VO.

4. EJEMPLO DE EXPLOTACIÓN CIENTÍFICA

Existe una gran cantidad de *software* compatible con VO en la web y disponible para toda la comunidad científica (<http://www.euro-vo.org/pub/fc/software.html>). Las aplicaciones son numerosas y diversas. Con Aladin, por ejemplo, podemos trabajar con imágenes, TOPCAT es útil para tablas de datos y además nos permite hacer representaciones gráficas, VOSA o VOSpec se utilizan para espectros.

Estas herramientas son útiles ya que son compatibles con el estándar de VO. Esto nos permite por una parte encontrar los datos de diferentes catálogos de forma muy sencilla, y además trabajar con ellos sin necesidad de realizar ajustes de formato, ya que éste responderá al mismo estándar VO.

A continuación describiremos tres casos prácticos utilizando herramientas VO.

4.1 MUESTRA DE LCBGs

Este ejemplo de explotación científica vamos a realizarlo con la ayuda del formulario web y la aplicación de VO TOPCAT (*Tool for OPERations on Catalogues And Tables*).

El objetivo es contribuir con las galaxias UCM a una muestra de LCBGs (*Luminous Compact Blue Galaxies*). Para ello debemos localizar las galaxias de nuestra Exploración que cumplen los criterios de las LCBGs, esto es: $M_b \leq -18.5$, $SBe \leq 21$ y $(B-V) \leq 0.6$. En nuestro caso no tenemos datos de V, así que el criterio será $(B-r) \leq 0.8$.

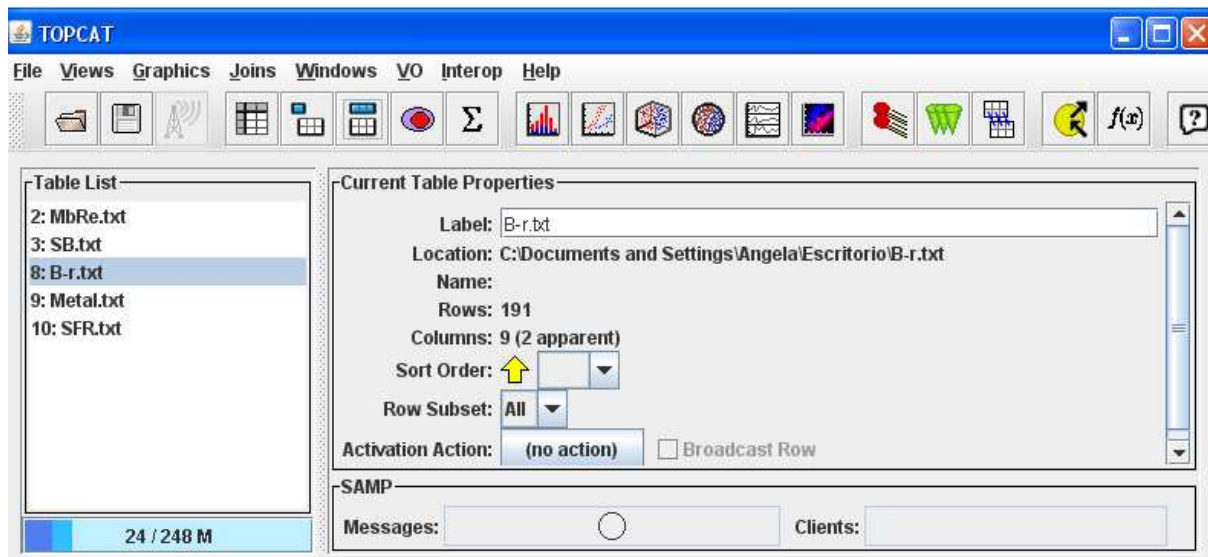
Una vez halladas las galaxias, nos interesarán las medidas de *redshift* (z), magnitud absoluta en B (M_b), Brillo superficial (SBe), índice de color, radio efectivo (R_e), tasa de formación estelar (SFR), masa de estrellas (M_{star}) y metalicidad.

Estas medidas se utilizarán para demostrar la capacidad de derivar parámetros globales de una muestra de LCBGs con espectroscopia tradicional (no 3D).

Un estudio similar de las medidas globales de una muestra completa de LCBGs a $d \leq 200$ Mpc se llevará a cabo utilizando los datos fotométricos y espectroscópicos publicados en el SDSS. Este estudio servirá de base para caracterizar las propiedades estadísticas de una muestra completa de LCBGs en el Universo cercano, de la cual se ha seleccionado una submuestra para su estudio detallado con espectroscopia 3D en óptico y en radio.

Para hacer esta selección vamos a obtener los datos desde el formulario web. Así pues, en primer lugar hacemos una consulta tal que nos devuelva para todas las galaxias estas tablas: fotometría en Johnson B, índices de color, cosmología, abundancias y modelos de población estelar. Guardamos cada una de las tablas que nos sean devueltas en formato *ASCII*.

A continuación iniciamos TOPCAT y cargamos las tablas que hemos guardado. Al hacerlo, es importante indicar que el formato de las tablas es *ASCII*. Una vez hecho esto veremos la ventana principal de TOPCAT así:



Recordemos que en nuestras tablas hay mucha información contenida, para seleccionar solamente aquella que nos interesa vamos a ver la cabecera de la tabla, haciendo clic en

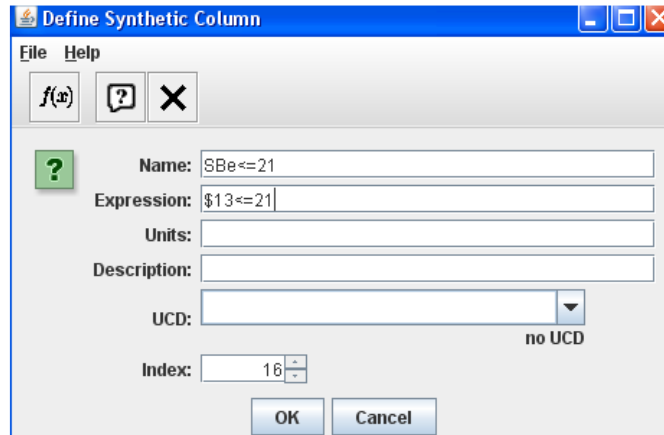
display column metadata, que es el sexto icono que vemos en la imagen. Desde ahí podremos seleccionar aquellas columnas que queremos que se muestren en la tabla.

Una vez tenemos en cada tabla las magnitudes que nos interesan, vamos a juntar todas las tablas en una sola. Esto se hace en *Joins, Match*. Ahí indicamos qué tablas queremos juntar y qué magnitud queremos comparar para hacer la unión de filas correctamente. En nuestro caso se ha escogido como algoritmo “*exact value*” y como columnas a comparar el nombre UCM de las galaxias. La tabla resultante es la siguiente, donde no se han mostrado todas las filas:

	id UCM	(B-r)	12+log(O)	SFR specific	M_star	Ra (deg)	Dec (deg)	Mb (mag)	Re (kpc)	SBe (mag/arcsec)
1	0000+2140		8,435		7,413100E10	0,79042	21,9608	-20,5	3,3	18,96
2	0003+1955		0,			1,58167	20,2031	-21,3	0,5	
3	0003+2200	1,19	8,4808	0,49	1,905460E9	1,40833	22,2825	-18,06	1,4	21,04
4	0003+2215		0,			1,46792	22,5358	-19,35	2,6	27,71
5	0005+1802		0,			2,12542	18,3242	-18,3	1,5	21,31
6	0006+2332		8,392		6,918312E9	2,22833	23,8178	-19,54	2,4	23,04
7	0013+1942	0,28	8,3967	0,48	2,691537E9	3,95792	19,98	-18,3	1,8	32,58
8	0014+1748	1,05	0,	0,62	1,174897E10	4,34917	18,0844	-19,83	5,3	20,18
9	0014+1829	0,55	8,2901	0,23	4,073800E9	4,31333	18,7717	-18,21	1,1	24,86
10	0015+2212	0,46	8,4637	0,24	5,248078E9	4,53167	22,4789	-18,12	1,	21,44
11	0017+1942	0,65	8,2641	0,12	1,023294E10	4,98917	19,9811	-19,6	2,8	23,
12	0017+2148		8,6772		3,019949E9	5,10958	22,0886	-17,9	1,	27,83
13	0018+2218		0,		4,570880E10	5,39042	22,5931	-19,24	4,2	26,59
14	0019+2201	0,89	8,9781	0,54	1,995263E9	5,4525	22,3019	-18,07	1,9	22,76
15	0022+2049	1,16	8,817	0,11	6,456549E9	6,1825	21,1014	-18,99	1,5	23,44
16	0023+1908		8,6278		2,238723E9	6,51292	19,4194	-18,62	1,2	
17	0034+2119		0,			9,17958	21,6075	-19,94	3,5	19,34
18	0037+2226		8,9067		2,630268E10	10,045	22,7158	-20,16	3,3	23,33
19	0038+2259	0,79	0,	0,32	2,238723E10	10,2875	23,2633	-20,29	4,1	25,35
20	0039+0054		0,			10,4425	1,1775	-19,34	4,8	
21	0040+0022		0,			10,8658	0,12583	-20,32	2,0	24,27

Como podemos observar, para algunas galaxias no tenemos datos del índice de color. En concreto existen 34 de las 191 galaxias que carecen de él, y que se van a descartar por la imposibilidad de aplicarles los criterios de selección.

Con esta tabla ya estamos en condiciones de aplicar los criterios de selección. Por ello, y de nuevo desde *display column metadata* creamos tres nuevas columnas, cada una de ellas para un criterio de selección. Al crear una nueva columna nos aparece el siguiente cuadro de diálogo:



Donde podemos nombrar la columna como se desee. Lo importante es la expresión que pongamos para la columna. En este apartado se pueden introducir fórmulas de todo tipo y además, como es nuestro caso, hacer comparaciones. Cuando creamos una columna como en la imagen superior, nos aparecerá una nueva columna de booleanos, cuyo valor será *true* en caso de ser cierta la condición que hemos escrito, donde \$13 es el identificador de la columna, que podemos ver al consultar la cabecera de la tabla. En este caso se está refiriendo a la columna de SBe.

Con las tres nuevas columnas creadas, no tenemos más que crear una tercera columna cuya expresión será “\$14==true&\$15==true&\$16==true”. Es decir, estamos creando una columna cuyas filas serán *true* cuando las otras tres condiciones sean también *true*.

Para obtener una tabla que contenga sólo aquellas galaxias que cumplan las tres condiciones, es decir las LCBGs, seleccionaremos aquellas cuya última columna sea *true* y entonces con *Subset*, *Subset from selected rows* obtendremos el resultado final, que mostramos a continuación:

TOPCAT(15): Table Browser														
Table Browser for 15: LCBG														
	(B-r)	12+log(O)	SFR specific	M_star	Ra (deg)	Dec (deg)	Mb (mag)	Re (kpc)	SBe (mag/ar...	z	B-r<=0.8	SBe<=21	Mb<=-18.5	Total
31	0,6	8,9781	0,07	2,238723E10	12,4796	21,1247	-20,21	3,2	20,6	0,0577	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
81	0,72	8,3932	0,67	1,445439E10	195,432	29,9275	-19,44	1,4	20,29	0,0307	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
102	0,44	8,6226	0,23	6,309576E9	201,715	26,5914	-19,9	1,3	17,68	0,0249	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
112	0,72	8,674	0,47	1,862089E10	218,758	26,5447	-20,36	4,9	20,84	0,0307	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
131	0,67	8,62	0,25	1,174897E10	252,272	29,7586	-20,36	3,6	20,98	0,029	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
138	0,79	0,		255,283	29,4067	-20,43	3,	19,76	0,0369		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
143	0,46	8,893	0,13	1,995263E10	343,145	24,7303	-21,59	3,2	17,65	0,0421	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
152	-0,47	0,		344,888	24,9183	-21,01	0,6	19,31	0,0345		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
166	0,52	8,4204	0,35	8,511386E8	349,502	19,6656	-18,9	2,1	19,48	0,0385	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
181	0,34	8,328	0,12	5,888443E9	352,541	25,5333	-19,23	1,	20,77	0,0206	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
182	0,6	8,253	0,17	3,388439E9	352,541	25,5322	-19,06	1,2	21,	0,0206	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
184	0,66	0,		352,973	25,2953	-19,82	1,7	19,83	0,0305		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Como vemos, 12 galaxias de nuestro catálogo cumplen los criterios para las LCBGs.

El proceso que hemos seguido para hacer la selección de galaxias se ha complicado por el hecho de tener que hacer las consultas en el formulario web, copiar y guardar las tablas como ascii, cargarlas por separado en TOPCAT y por último tener que unir las.

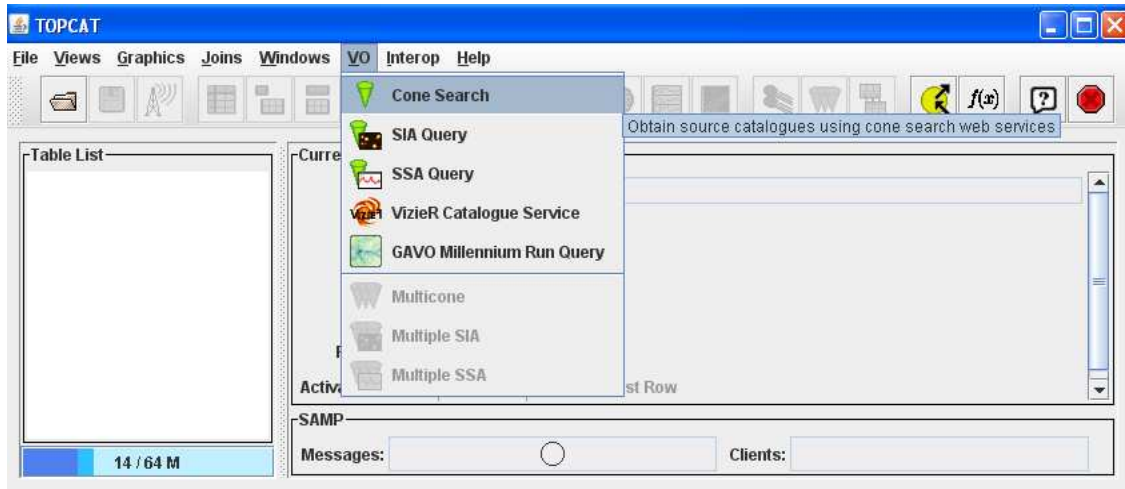
En el siguiente ejemplo vamos a comprobar que utilizando el servicio VO es mucho más rápido y sencillo recuperar nuestros datos con una herramienta VO.

4.2 *PLANO FUNDAMENTAL DE GALAXIAS CON FORMACIÓN ESTELAR*

Recientemente se ha encontrado una relación entre la tasa de formación estelar, la metalicidad y la masa de estrellas para galaxias con formación estelar. En concreto, se ha llevado a cabo un estudio con galaxias con formación estelar de la *main galaxy sample* del Sloan Digital Sky Survey – Data Release 7 (SDSS-DR7) cuyo resultado ha sido un plano fundamental que se obtiene al representar en 3D las tres magnitudes previamente mencionadas (2010arXiv1005.0509L).

En este apartado vamos a reproducir este gráfico en 3D para comprobar si las galaxias de la Exploración UCM se ajustan también a este plano fundamental. Para ello utilizaremos de nuevo TOPCAT.

Al iniciar el programa, en el menú superior tenemos la opción *VO, ConeSearch*. Desde aquí vamos a obtener nuestros datos:



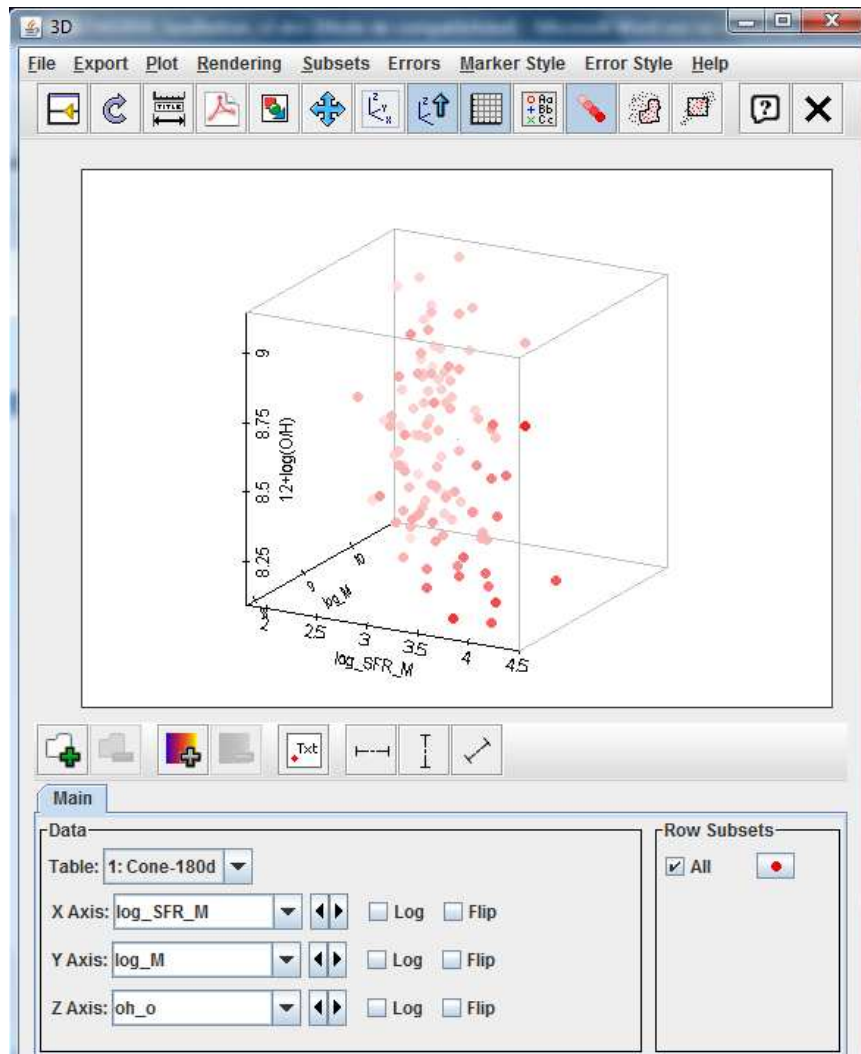
En la ventana que nos aparece buscaremos nuestro servicio VO, introduciendo en *ConeSearch URL* la dirección <http://sdc.cab.inta-csic.es/gucm/ConeSearch/vosearch.jsp> o bien buscando el servicio en *Registry*.

A continuación introducimos unas coordenadas y un radio suficientemente grande para que el resultado de la búsqueda englobe todo nuestro catálogo. Este resultado nos aparecerá en forma de tabla, como la siguiente:

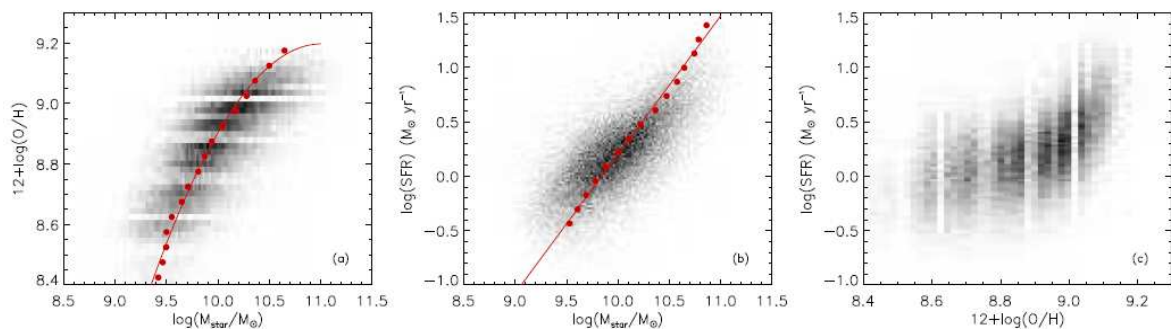
unique_id	ra	dec	Sp_type	L_Ha_spe	Mb_pag	Mb_pag_e	EW_Ha	b
1	0000+2140	0,79042	21,9608	IIIIH		-20,78	0,06	99,8
2	0003+1955	1,58167	20,2031	Sy1		-21,49	0,06	291,5
3	0003+2200	1,40833	22,2825	DANS	0,28	-18,15	0,05	34,8
4	0003+2215	1,46792	22,5358	SBN		-19,45	0,05	21,6
5	0005+1802	2,12542	18,3242	SBN		-18,38	0,08	10,2
6	0006+2332	2,22833	23,8178	IIIIH		-19,68	0,06	54,6
7	0013+1942	3,95792	19,98	IIIIH	0,87	-18,61	0,05	121,5
8	0014+1748	4,34917	18,0844	SBN	2,97	-19,81	0,06	83,3
9	0014+1829	4,31333	18,7717	IIIIH	0,56	-18,62	0,11	127,9
10	0015+2212	4,53167	22,4789	IIIIH	0,99	-18,3	0,05	116,9
11	0017+1942	4,98917	19,9811	IIIIH	1,65	-19,79	0,04	96,6
12	0017+2148	5,10958	22,0886	IIIIH		-18,16	0,07	70,9

Con estos resultados seleccionamos las columnas de las magnitudes que nos interesan: tasa de formación estelar, metalicidad, masa de estrellas y sus respectivas incertidumbres. Con nuestra tabla de datos construida, pasamos ya a representar gráficamente desde *Graphics, 3D*. Se abre ahora una nueva ventana donde podemos

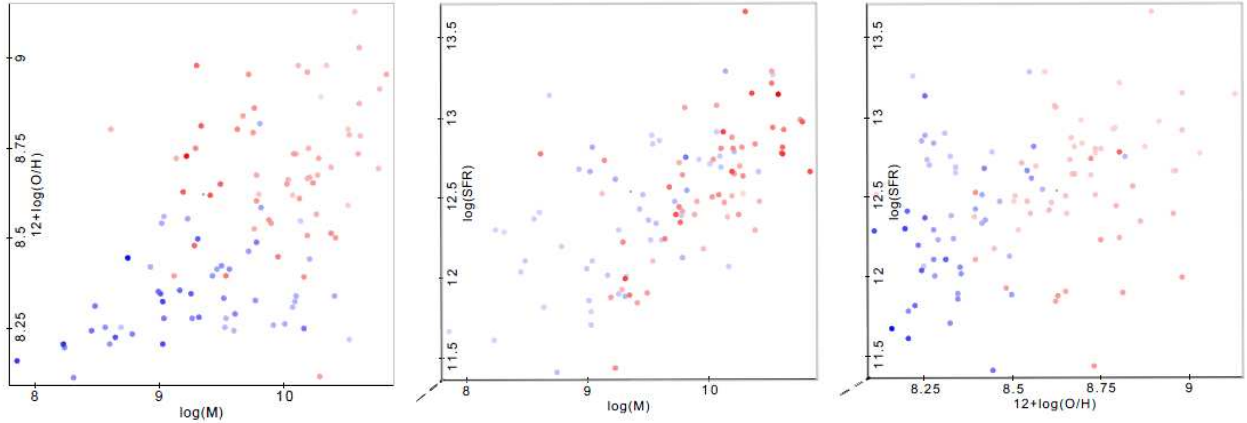
introducir las diferentes coordenadas así como sus incertidumbres, seleccionando la columna correspondiente mediante un menú desplegable:



En el artículo citado aparecen las siguientes proyecciones del gráfico en 3D, que vamos a reproducir para comparar el comportamiento:



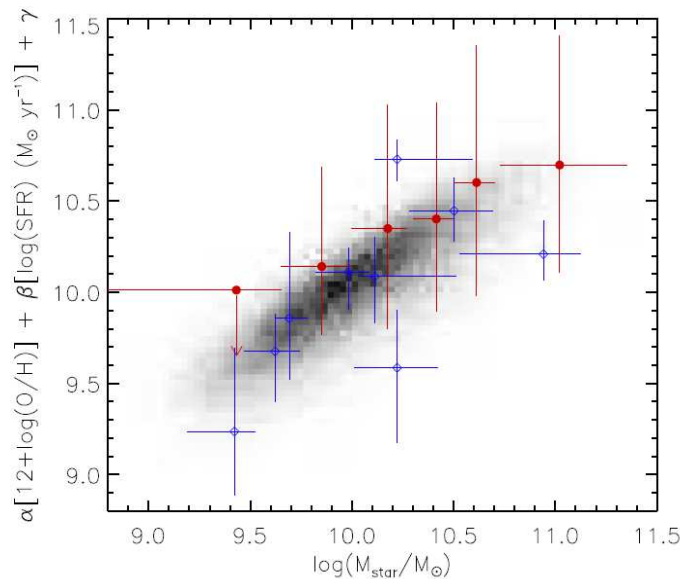
Si arrastramos el gráfico obtenido en 3D con el ratón, podemos rotarlo en cualquier dirección. Así, conseguimos las proyecciones sobre los mismos ejes:



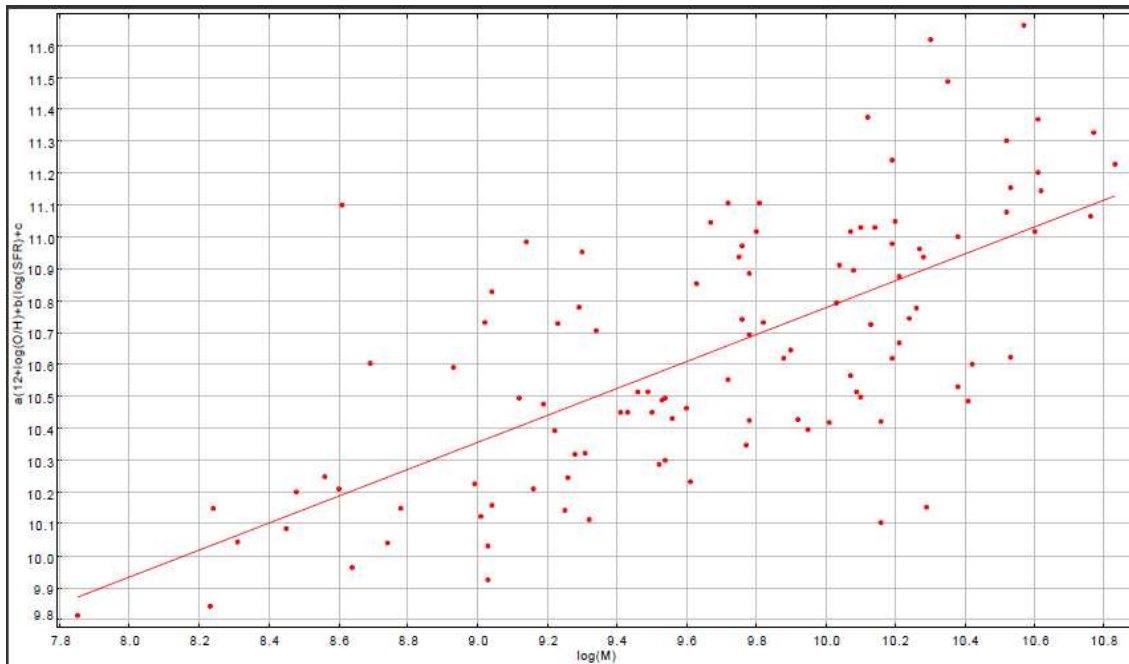
Hemos diferenciado las galaxias SBN/DANS en rojo y las BCD/HIII en azul. Los colores más débiles indican que los puntos están más profundos en el gráfico. A partir de estas gráficas ya podemos intuir que nuestros datos no se ajustan a los del artículo citado, sin embargo vamos a hacer una comprobación más rigurosa. La ecuación del plano que se obtiene en el artículo es:

$$\log(M_{\text{star}}/M_{\odot}) = \alpha [12 + \log(\text{O}/\text{H})] + \beta [\log(\text{SFR}) (M_{\odot}\text{yr}^{-1})] + \gamma$$

Donde $\alpha = 1.205 \pm 0.008$, $\beta = 0.460 \pm 0.004$, $\gamma = -0.79 \pm 0.07$. Esto es:



Vamos a realizar una representación y el correspondiente ajuste lineal a la anterior ecuación para comprobar la desviación de nuestros datos:



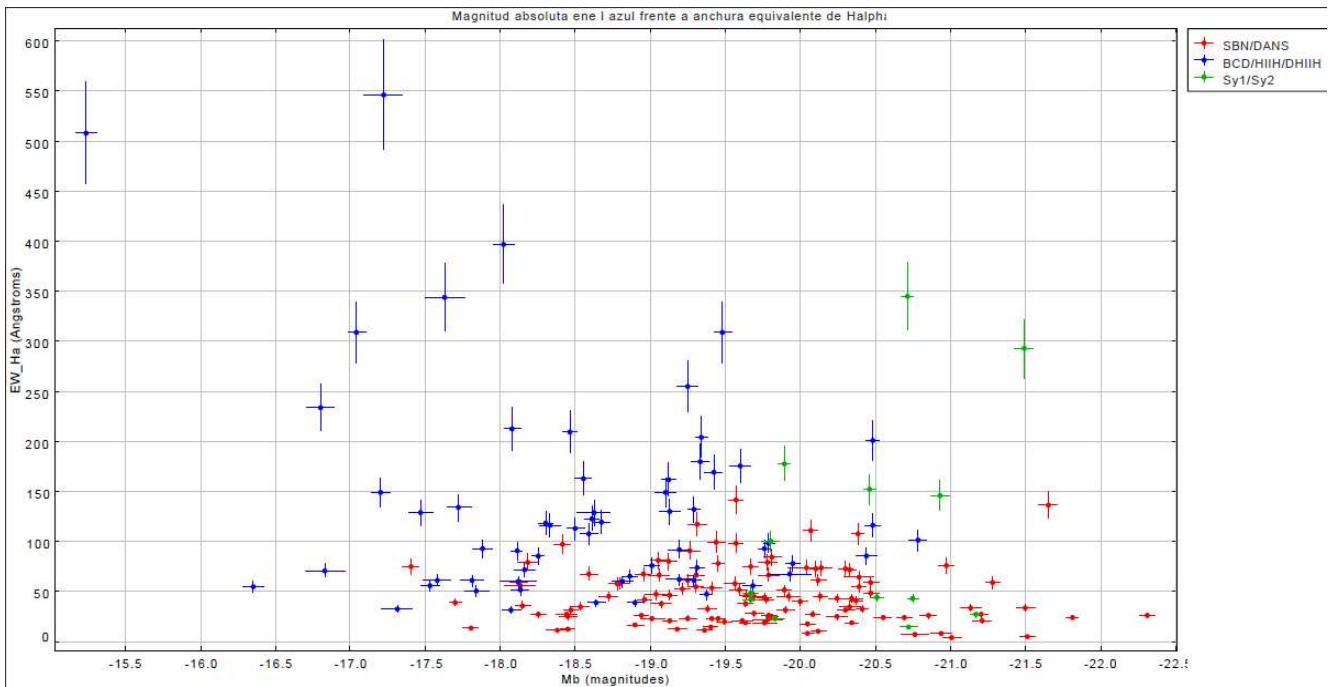
Donde la regresión lineal la hemos hecho desde *Row Subsets*, en la misma ventana de gráficos. Los parámetros a, b y c son los mismos que para la ecuación del plano, y en estas circunstancias la pendiente de esta gráfica debería ser igual a la unidad, como se puede apreciar en la gráfica del artículo mostrada anteriormente. Sin embargo, la pendiente obtenida es de 0.422 y la dispersión de puntos muy grande. Así pues, podemos concluir que los datos de nuestras galaxias no se ajustan a dicho plano fundamental. El estudio de este resultado científico queda entonces abierto para futuros trabajos.

4.3 ESTUDIO DE LA EXPLORACIÓN UCM

En este ejemplo vamos a trabajar exclusivamente con los datos de la Exploración UCM, y vamos a comprobar cómo TOPCAT es una herramienta idónea para estudiar las relaciones entre las magnitudes de nuestro catálogo.

Siguiendo el mismo procedimiento de obtención de nuestros datos con TOPCAT que en el apartado anterior, es decir, utilizando la opción *VO, ConeSearch*, vamos a realizar algunos gráficos y ajustes.

En primer lugar vamos a representar la magnitud absoluta en el azul frente a la anchura equivalente de $H\alpha$:

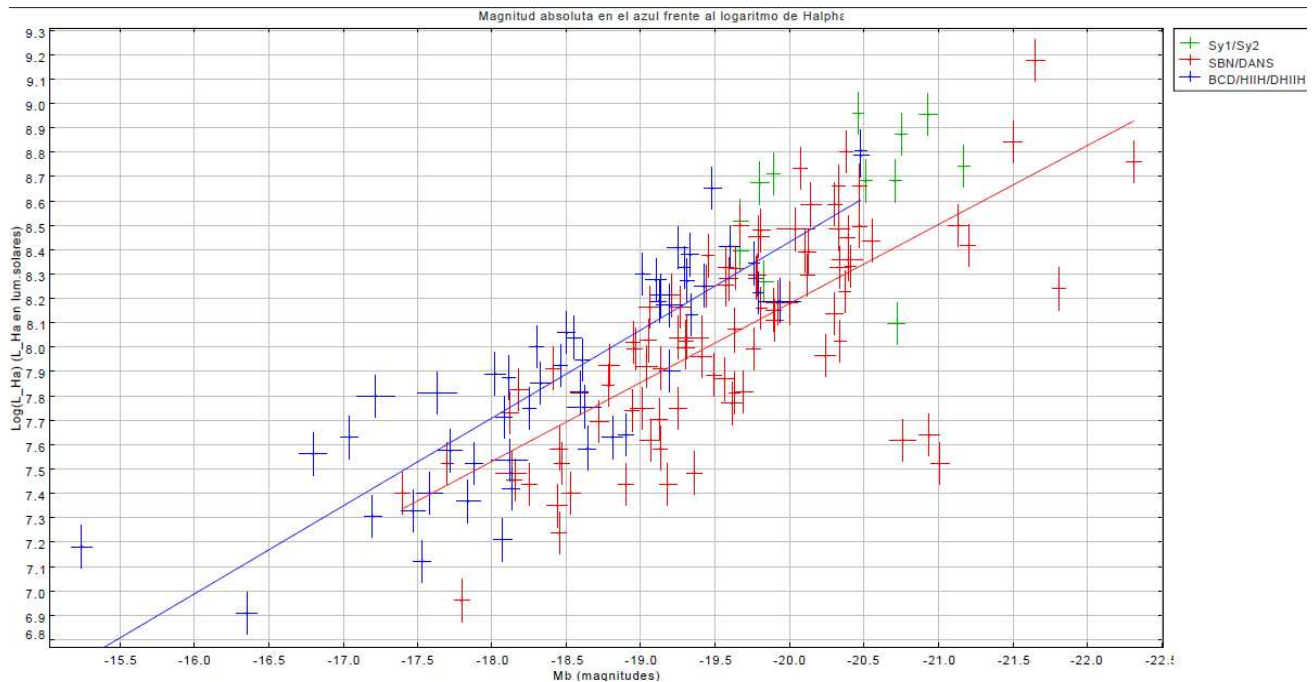


Podemos observar segregación de la población de SBN, cuyas galaxias tienden a ser más brillantes, frente a la población de HII que tienden a ser más débiles.

Por otra parte podemos observar la relación estructural universal por la que las galaxias con magnitudes absolutas más brillantes presentan anchuras equivalentes más altas. Esto es debido a que la contribución relativa del proceso de formación estelar a la luminosidad total de la galaxia es mayor para las galaxias de menor luminosidad.

Las galaxias Seyfert (verde) no siguen la tendencia general porque tienen la contaminación de un agujero negro, que suele presentarse en objetos muy masivos, por lo que sólo vemos el fenómeno Seyfert en la parte de las galaxias más masivas de la muestra.

A continuación representaremos la magnitud absoluta en el azul frente al logaritmo de la luminosidad en H α :



La luminosidad en H α es un trazador directo de la tasa de formación estelar instantánea y presenta una clara correlación con la magnitud absoluta, dado que de manera general las galaxias más luminosas tienden a presentar brotes de formación estelar más intensos.

Cabe notar que las poblaciones de las galaxias tipo HII y tipo SBN siguen dos tendencias similares pero paralelas, de manera que para una magnitud absoluta dada, las galaxias HII tienen fuerzas de brote mayores, y por lo tanto tasas de formación estelar mayores.

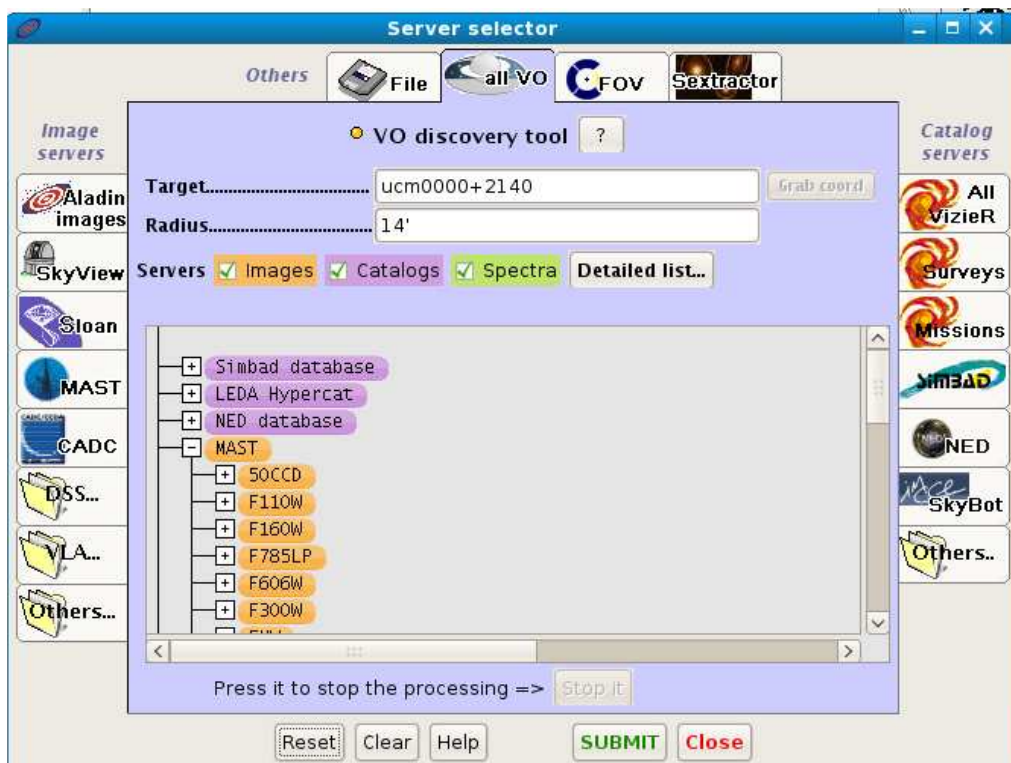
Las galaxias Seyfert (verde) presentan luminosidades H α muy intensas debido a que hay una contribución originada en el objeto compacto, que no está relacionada con el proceso de formación estelar.

4.4 TRABAJO CON DIFERENTES CATÁLOGOS

En el siguiente ejemplo vamos a trabajar con *Aladin Sky Atlas*, con el objetivo de comprobar la interoperabilidad que nos ofrece VO para trabajar con diferentes catálogos.

En primer lugar vamos a realizar una búsqueda. En este caso vamos a introducir el nombre de una galaxia: *ucm0000+2140* y vamos a comprobar cuántos catálogos nos dan un resultado para la búsqueda. Para ello, desde la pantalla principal de Aladin vamos a *File, Open*. Nos aparecerá entonces una ventana de *Server Selector*, y desde ahí vamos a la pestaña de *all VO*. En *Target* escribimos el identificador de la galaxia y desde *Detailed list* podemos ver los servicios disponibles de imágenes, catálogos y espectros.

Existen dos opciones, o bien realizamos una búsqueda para todos los servicios, o podemos escoger solamente aquellos que nos interesen. En este caso vamos a realizar una búsqueda amplia. Nos aparecerán entonces aquellos servicios para los que se obtenga un resultado a la *query*:



Vamos a escoger por ejemplo POSSII (ESO) para obtener una imagen, 2MASS y Simbad que se encuentran en el CDS, y nuestro catálogo de galaxias que está en Madrid. El resultado es el siguiente:

The screenshot shows the Aladin v6.0 software interface. The main window displays a grayscale image of a galaxy with a central bright spot and a surrounding diffuse structure. The position is given as 00:03:08.95 +21:57:33 in ICRS coordinates. The image is zoomed 4x. A table at the bottom lists associated data from various catalogs.

	MAIN_ID	DTYPE	RA	DEC	COO...	COO...	C...	PMRA	PMDEC	B			
<input type="checkbox"/>	Nrk 334	Seyf...	00 03 09.624	+21 57 36.90						14.62			
<input type="checkbox"/>	000_790016	+21.960224	00030960...	12.912	0.055	11.925	0.053	11.096	0.033	AAA	222	111	00
<input type="checkbox"/>	0000+2140	0.790417	21.9608	HIH		null	-20.78	0.06	99.8	3.3		0.2	

Así pues, vemos que con los servicios VO se consigue una mayor agilidad en la búsqueda y gestión de datos. Dadas unas coordenadas o un identificador podemos saber rápidamente qué catálogos tienen datos para el objeto con el que trabajamos, y además ver qué datos son estos y trabajar directamente con ellos, sin importar en qué lugar estén almacenados.

5. CONCLUSIONES Y TRABAJO FUTURO

Con este trabajo académicamente dirigido se ha logrado poner en marcha un servicio VO para el catálogo de galaxias con formación estelar de la UCM. Gracias a este servicio el acceso y tratamiento de los datos será, a partir de ahora, más sencillo y efectivo, tanto para los miembros de la UCM como para cualquier otro investigador.

Además, el servicio está integrado en el Centro de Datos Científicos del Centro de Astrobiología (CAB), lo que asegura visibilidad y mantenimiento a largo plazo.

Por otra parte existen algunas mejoras y complementos que se podrían hacer sobre este trabajo, para así conseguir un servicio VO y una página web más completas. A continuación citamos una serie de sugerencias a realizar en un futuro.

En primer lugar cabe destacar que la base de datos que hemos creado así como la página web de consulta, contienen más información de la que posteriormente el Servicio VO muestra. Así pues con la base de datos se ha dejado el camino abierto a ampliar la información de la exploración UCM en VO. En concreto, sería muy interesante implementar el servicio para imágenes y espectros.

Por otra parte la página web de consulta se podría completar para que ofreciese otras funcionalidades, como por ejemplo la devolución de los datos en otros formatos además de en HTML, o bien que existiera la posibilidad de realizar la búsqueda de una lista de nombres o coordenadas. Una explicación de la exploración UCM se podría incluir antes del formulario en la misma página de inicio. También sería interesante que en las tablas de resultados se incluyera un enlace VO por cada objeto, que nos condujera a toda la información espectroscópica, fotométrica y de catálogos existente.

6. BIBLIOGRAFÍA

- *Aladin Sky Atlas*:
<http://aladin.u-strasbg.fr/>
- Cardelli, Jason A. et al. “*The relationship between infrared, optical, and ultraviolet extinction*”, 1989ApJ...345..245C.
- ConeSearch, especificaciones sobre el servicio, versión 1.03. Recomendación IVOA del 22 de febrero de 2008:
<http://www.ivoa.net/Documents/REC/DAL/ConeSearch-20080222.html>
- Gallego Maestro, J. “*Estudio de una muestra completa de galaxias con emisión en H α* ”. Tesis Doctoral, febrero 1995.
- GUAIX, página web del Grupo de astrofísica extragaláctica e instrumentación astronómica de la UCM, apartado del Survey UCM:
http://guaix.fis.ucm.es/ucm_survey/
- IVOA, página web de la Alianza Internacional para el Observatorio Virtual:
<http://www.ivoa.net/>
- Lara-López, M. A. et al. “*A fundamental plane for field star-forming galaxies*”. 2010arXiv1005.0509L.
- Muñoz Mateos, J.C. “*Base de Datos de la Exploración UCM*”, Trabajo Académicamente Dirigido, 2004-2005.
- O’Donnell, James E. “*R_{nu}-dependent optical and near-ultraviolet extinction*”, 1994ApJ...422..158O.
- Pisano Daniel J. et al. “*The Evolution of Luminous Compact Blue Galaxies: Disks or Spheroids?*”. Bulletin of the American Astronomical Society, Vol. 42, p.427. 2010AAS...21532904P.

- PostgreSQL Global Development Group, “*PostgreSQL 8.1.11 Documentation*”, 1996.
<http://www.postgresql.org/files/documentation/pdf/8.1/postgresql-8.1-A4.pdf>
- Romero Laguillo, L.F. “*Publicar en Internet: guía práctica para la creación de documentos HTML*”. Servicio de publicaciones de la Universidad de Cantabria, 1997.
ISBN 84-8102-179-2
- Schlegel, David J. et al. “*Maps of Dust Infrared Emission for Use in Estimation of Reddening and Cosmic Microwave Background Radiation Foregrounds*”, 1998ApJ...500..525S.
- SVO, página web del Observatorio Virtual Español:
<http://svo.cab.inta-csic.es/>
- TOPCAT: Tool for OPerations on Catalogues And Tables, herramienta VO.
<http://www.star.bris.ac.uk/~mbt/topcat/>
- UCD, lista de palabras válidas y especificaciones sobre cómo construir los UCD:
<http://cdsweb.u-strasbg.fr/UCD/>
- VO-Software:
<http://www.euro-vo.org/pub/fc/software.html>