



Treball Final de Carrera

Entorn de gestió de dades d'ornitologia

Manel Muñoz Argemí

**Enginyeria Tècnica de Telecomunicació (Esp. Sistemes de
Telecomunicació)**

Director: Albert Baucells Colomer

Vic, gener de 2008

ÍNDEX

1 – RESUM	2
2 – INTRODUCCIÓ.....	4
3 – OBJECTIUS	6
4 – EXPOSICIÓ TREBALL	8
4.1 – SUPORT UTILITZAT	8
4.2 – RECURSOS DE DESENVOLUPAMENT.....	10
4.2.1 – MySQL	10
4.2.2 – PHP.....	15
4.2.3 – SMARTY	33
4.2.4 – CSS (Cascading Style Sheets).....	37
4.3 – ESTRUCTURA GENERAL DEL PROGRAMA.....	39
4.4 – DESCRIPCIÓ DETALLADA DELS ARXIS DE GESTIÓ	43
4.4.1. – INDEX.PHP	43
4.4.2 – INDEX.TPL.....	46
4.4.3 – PRESENTACIÓ ASSOCIADA A L'INDEX.PHP.....	50
4.4.4 – GRAFICSA.PHP	53
4.4.5 – GRAFICSA.TPL	53
4.5 – ESTRUCTURA DELS ARXIS DE TRACTAMENT DE DADES.....	62
4.5.1 – PRIMERES ENTRADES I ÚLTIMES SORTIDES.....	63
4.5.2 – ACTIVITATS DIARIES NIU.....	66
4.5.3 – ENTRADES/SORTIDES DIARIES	69
4.5.4 – COMPARATIVA ENTRADES/SORTIDES 2 RANGS	70
4.5.5 – DISTRIBUCIÓ HORA D'ENTRADES DURANT EL DIA.....	73
4.5.6 – COMPTADOR DIARI D'OCELLS EN NIU	75
4.6 – QUANTIFICACIÓ DEL PROJECTE.....	76
5 – CONCLUSIONS I PROPOSTES D'AMPLIACIÓ	79
5.1 – CONCLUSIONS	79
5.2 – PROPOSTES D'AMPLIACIÓ	80
6 – BIBLIOGRAFIA	82



Resum de Treball Final de Carrera
Enginyeria Tècnica de Telecomunicació (Esp. Sistemes de Telecomunicació)

Títol: Entorn de gestió de dades d'ornitologia

Autor: Manel Muñoz Argemí

Director: Albert Baucells Colomer

Data: gener de 2008

Resum

Des de fa uns anys, des de l'EPS de la UVic, s'està duent a terme el desenvolupament d'un dispositiu electrònic que proporciona la capacitat de capturar dades sobre un niu d'ocells. El projecte e-niu, que es pot seguir a www.tutara.info/e-niu, està actualment en una fase de test., i s'ha desenvolupat sobretot la part hardware.

Aquest projecte té com a objectiu principal crear un entorn web per poder gestionar les dades que s'obtenen del niu d'ocells informatitzat (e-niu). Les dades que ens arriben dels e-nius estan en un arxiu de text, i el que es pretén és que l'usuari que controla el niu, pugui fer diversos anàlisis de les dades d'aquest. A més de poder veure els resultats en diversos tipus de gràfics, també se li vol donar la opció de treure els resultats en format taula o en format Excel, un format aquest últim, molt interessant, ja que donaria a les dades una gran possibilitat de ser tractades posteriorment, com fer seleccions del tipus de dades, treure percentatges, crear altres tipus de gràfics, etc.

L'altre gran objectiu és el fet de poder treballar en la creació d'un entorn web complert a nivell gairebé professional amb l'aprenentatge que això comporta, ja que s'ha aplicat la tecnologia client-servidor, és a dir, que el llenguatge de programació està dins el servidor, i quan algun usuari l'executa, el sistema només li envia la presentació en HTML.

El sistema de programació que es fa servir és el de les tres capes. La capa de dades, que està formada per una base de dades relacional del tipus MySQL i és on emmagatzemem tota la informació. La capa de programació de la que s'encarrega el llenguatge PHP, és on s'efectua tot el tractament de les dades i finalment, la capa de presentació, que és la que s'encarrega de mostrar les dades al client en el navegador mitjançant els templates de HTML.



Summary of Final Work of Career
Enginyeria Tècnica de Telecomunicació (Esp. Sistemes de Telecomunicació)

Title: Surroundings of data management of ornithology

Author: Manel Muñoz Argemí

Manager: Albert Baucells Colomer

Date: January 2008

Summary

Years ago, from the EPS of the UVic, is being carried out the development of an electronics that provides the capacity to capture data on a nest of birds. The project e-nest can be followed on www.tutara.info/e-niu, and nowadays is in a stage of test, and the mainly part has been developed is the hardware.

This project must like primary target create surroundings Web to be able to manage the data that are obtained from the nest of birds computerized (e-nest). The data arrive to us from the e-nests in a text file, and the nest control's user can make diverse analyses of these data. In addition to being able to see the results in several types of graphs, also we can take the results in formed table or Excel format; this one is very interesting because we have the great possibility of treated the data, like making selections of the data type, removing percentage, to create other types of graphs, etc.

The other great objective is the fact of being able to work in the creation of surroundings complete Web at almost professional level with the learning that this involve. It has been applied to the technology client-server which the programming language is inside the server, and when some user executes it, the system only shows the presentation in HTML.

We use the three layers programming system. The layer of data is formed by a relational database of the MySQL type and is where we will store all the information. The programming layer of which language PHP is in charge, is where all the treatment of the data takes place finally and, the presentation layer, show the data to the client in the navigator by means of templates of HTML.

2 – INTRODUCCIÓ

Aquest treball té per objectiu desenvolupar un entorn web per gestionar i analitzar les dades que s'obtenen d'un niu d'ocells informatitzat.

La realització d'aquest treball va sorgir arrel d'una conversa amb l'Albert Baucells, professor de la Escola Politècnica Superior de la UVIC, quan em va comentar que des de fa uns anys s'està desenvolupant el projecte e-niu des de la mateixa universitat, i el mateix Albert amb en Jordi Serra, entre d'altres, són els impulsors d'aquest projecte.

L'e-niu pròpiament dit, és una caixa de fusta equipada amb uns dispositius electrònics, entre els quals hi ha un microcontrolador i diversos sensors, que tenen la capacitat de generar un arxiu de text (.txt) amb totes les dades del niu d'ocells, com poden ser les activitats dels ocells (entrades, sortides, ...), l'estat de la bateria, la temperatura, així com la data i l'hora de cada dada.

El desenvolupament del projecte es pot seguir a www.tuatara.info/e-niu que actualment està en una fase de test, on s'hi pot veure des de la construcció de l'e-niu fins a la situació actual del projecte. Fins ara s'ha desenvolupat, sobretot la part hardware.

El que es pretén amb aquest projecte, per una banda és aprendre a realitzar un entorn web per treballar en una activitat real, i per l'altra proporcionar un entorn de treball útil per a poder gestionar, analitzar i treballar amb les dades que proporciona l'e-niu.

Amb aquest programa l'usuari dels e-nius podrà donar d'alta el niu amb el que està treballant, introduir-hi el fitxer de text amb totes les dades generades pel e-niu i posteriorment analitzar els resultats, ja sigui amb unes gràfiques que s'han considerat d'interès, ja sigui en una taula o en format Excel, molt útil per poder tractar posteriorment les dades.

Per a la realització d'aquest entorn web, s'ha treballat amb les 3 capes de programació, és a dir, la capa de base de dades, la capa d'entorn servidor i la capa de presentació.

Per la capa d'emmagatzemar les dades s'ha creat una base de dades relacional del tipus Mysql en la que s'hi guarda tota la informació, tant dels nius que crea cada usuari, com dels e-nius físics, com dels usuaris.

El llenguatge de programació d'entorn servidor que s'ha fet servir en la capa de programació ha estat el PHP, molt útil en les pàgines dinàmiques ja que permet fer tota la gestió de les dades, com són els càlculs, consultes a les bases de dades, crear les gràfiques, treure llistats en format Excel, etc.

Finalment, en la capa de presentació s'ha utilitzat l'estàndard HTML com a llenguatge descriptiu amb l'objectiu de donar format al text i imatges, que finalment visualitzarem en el navegador, així com petits programes en Javascript.

3 – OBJECTIUS

L'objectiu principal d'aquest treball és poder facilitar a l'usuari d'una forma clara, entenedora i realment útil els resultats de tota la informació extreta de l'e-niu.

Per això s'ha considerat oportú el fet de poder observar els resultats en una sèrie de gràfiques considerades d'interès ornitològic, com són les següents:

- **PRIMERES ENTRADES I ÚLTIMES SORTIDES:**
Aquesta gràfica ens permet veure la zona horària d'activitat del niu, ja que ens mostra de cada dia la primera entrada i l'última sortida del dia.
- **ACTIVITATS:**
Ens mostra les gràfiques amb les possibles activitats de cada dia, ja sigui entrades, sortides, activitats exteriors (quan l'ocell treu el cap al niu però no acaba d'entrar-hi), activitats interiors (quan l'ocell treu el cap del niu però no acaba de sortir-ne) i incerts (quan hi ha hagut una seqüència de detecció que no correspon a cap de les activitats exteriors).
- **NOMBRE TOTAL D'ENTRADES, SORTIDES I MOVIMENTS EN GENERAL:**
Ens mostra el sumatori total, ja sigui de les entrades, de les sortides o de totes les activitats que ha anat enregistrant l'e-niu. Com en totes les altres gràfiques, és important que es pugui seleccionar el rang d'anàlisi per poder efectuar un estudi més precís.
- **COMPARATIVA ENTRADES/SORTIDES DOS RANGS:**
Aquest anàlisi ens permet veure el total d'entrades i sortides en dos rangs de dates diferents del mateix gràfic.

- **DISTRIBUCIÓ ENTRADES DURANT EL DIA:**
Permet analitzar en quina zona del dia es realitzen les entrades. Les fraccions del dia es poden configurar. Si seleccionem 24 fraccions, equivaldrà a les entrades de cada hora.
- **DISTRIBUCIÓ ENTRADES COMPARAT AMB ACTIVITATS:**
La mateixa gràfica del punt anterior, però acompanyada de la gràfica d'activitats total del niu indicant de quina zona són les dades per obtenir la distribució.
- **COMPTADOR DIARI:**
Aquesta gràfica és especialment útil per veure en quines dates es van acumulant més ocells dins el niu o en quina queda buit. També pot ser important per detectar una diferència notable entre les entrades i les sortides del niu, tot indicant un possible mal funcionament i permetent la correcció.

També s'ha volgut donar la opció de passar les dades de qualsevol de les gràfiques anteriors a Excel, i així tenir tota la informació en un format molt accessible per a possibles tractaments posteriors.

A més de fer una eina útil per analitzar dades d'ornitologia, el que es pretén és l'aprenentatge de la creació d'un entorn web, mitjançant el llenguatge PHP, que permetrà connectar-nos amb la base de dades i poder fer tot tipus de tractament de les dades. És interessant també el fet de que el PHP és un llenguatge que ens permet treballar amb la programació orientada a objectes, és a dir, podem efectuar una programació molt més estructurada creant "classes", com ja explicarem en el proper punt.

Finalment, també es pretén aprendre a treballar amb el llenguatge HTML i la manera de que es pugui lligar amb els arxius de PHP mitjançant el paquet Smarty, que també s'explicarà més endavant.

4 – EXPOSICIÓ TREBALL

4.1 – SUPORT UTILITZAT

Aquest projecte es basa en emmagatzemar molta informació en una base de dades, per tal de poder tractar-la posteriorment, i això es fa mitjançant una pàgina web dinàmica.

En realitat, el llenguatge HTML es tracta d'un llenguatge descriptiu que es fa servir per donar format al que vulguem visualitzar en el navegador. Permet introduir enllaços cap a altres pàgines, seleccionar la mida de les *fonts* o intercalar imatges, però tot això d'una manera prefixada i en cap cas intel·ligent. HTML no ens permet realitzar cap càlcul matemàtic ni crear cap pàgina a partir d'una base de dades. És molt útil a petita escala però queda força limitat a l'hora de crear grans llocs o portals que tracten gran quantitat de dades.

Aquesta deficiència del HTML ens ha portat a fer servir llenguatges de programació més versàtils i que responen de manera intel·ligent a les demandes del navegador, permetent l'automatització de determinades feines mentre que s'ha deixat el HTML per la part de presentació.

Partint de la base que el navegador és una espècie d'aplicació, que tan sols és capaç d'interpretar i mostrar les ordres rebudes en forma de codi HTML fonamentalment, ens trobem davant la necessitat de treballar amb un llenguatge d'entorn servidor (llenguatges reconeguts, executats e interpretats pel propi servidor). El PHP que es fa servir en aquest projecte és un d'aquests llenguatges de programació, i són els que formaran part d'aquest concepte de pàgines dinàmiques. Són pàgines que ens permetran treballar amb un lloc web molt extens, amb continguts que poden variar ràpidament, i el fet de poder automatitzar les feines i processos ens portarà a una òptima explotació de les dades.

Aquest tipus d'arxius de PHP es componen exclusivament de text, de manera que els podem crear en qualsevol editor de textos com el bloc de notes, sempre i quan el guardem amb l'extensió .php.

S'ha fet servir la configuració Standard del PHP que hi ha a l'arxiu php.ini, malgrat que en alguna gràfica hem modificat manualment un paràmetre de configuració com és el temps d'execució, que l'hem passat de 30 a 50 segons amb l'instrucció:

```
set_time_limit(50);
```

També es podria haver fet modificant la configuració general del PHP modificant el temps en la següent sentència del php.ini.:

```
“ max_execution_time = 30 ; Maximum execution time of each script, in seconds”
```

Un cop realitzat el programa en PHP, s'han d'executar els *scripts* en un servidor, havent de penjar per FTP els arxius que el componen i fer la petició des del mateix navegador. No és aconsellable programar i fer les proves en un servidor de la xarxa, ja que possibles irregularitats o errors, com bucles infinits, poden estar consumint recursos importants d'aquests servidors, i ens impossibilita programar desconnectats de la xarxa. És per això que amb el paquet que ens hem descarregat, l'Apache2triad, compostat pel llenguatge PHP, el servidor de bases de dades MySQL i el servidor de pàgines web Apache, ens ha permès treballar en local, és a dir, convertir el nostre ordinador en un servidor i així poder realitzar tota mena de proves sense la necessitat d'estar connectats a la xarxa, permetent-nos deixar pel final, quan ja tot funciona, el penjar els arxius per FTP a un servidor de la xarxa.

Amb el servidor de pàgines web Apache, també hem treballat amb la configuració Standard, llevat d'una modificació. S'ha definit un àlias per fer la Url independent del lloc on es troba la web.

Alias /eniu/ "C:\AppServ\www\proves\DEFSMARTY\ENIU"

De manera que ara quan algú obre la pàgina no sap on tenim guardats els arxius, ja que la cridem per l'alias. Fixem-nos amb el canvi de direcció:

<http://127.0.0.1/proves/DEFSMARTY/ENIU/index.php>

<http://127.0.0.1/eniu/index.php>

4.2 – RECURSOS DE DESENVOLUPAMENT

Una part molt important en la creació d'un lloc web com aquest és la capa de bases de dades, és com i a on guardem la gran quantitat d'informació que tractem, i això ho fem en una base de dades del tipus MySQL.

4.2.1 – MySQL

El sistema de treball consisteix en crear una base de dades inicial, que contindrà les diverses taules que anirem creant, com és la dels e-nius existents, la dels usuaris que anem donant d'alta, la dels nius que creen els usuaris i finalment la dels moviments de cada niu.

S'ha utilitzat el sistema de bases de dades relacional, ja que les taules estan relacionades entre elles. Al crear un niu li corresponen uns determinats moviments de la taula moviments. A més, hi ha un usuari de la taula d'usuaris que és qui l'ha donat d'alta, de manera que per cada niu creat només li correspon un usuari. El niu també està relacionat amb un dels e-nius de la taula corresponent, ja que és el niu informatitzat que ha fet servir.

L'esquema d'aquesta base de dades relacional és el que ens mostra la figura 4.2.1.1:

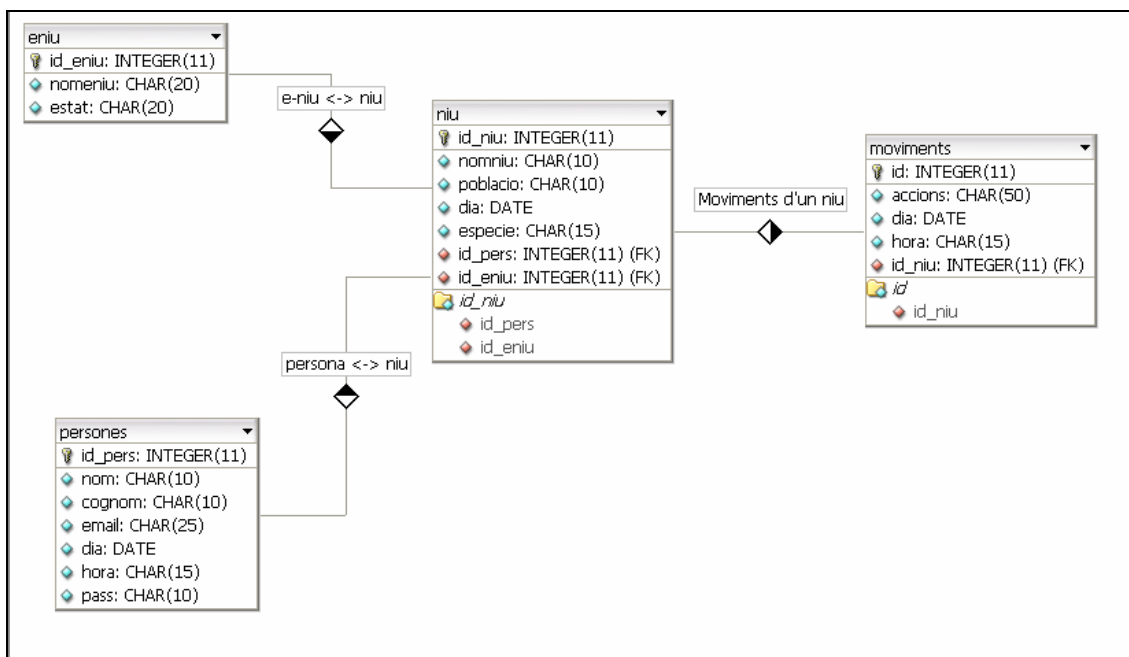


Fig 4.2.1.1

Taules (entitats):

- Eniu: Conté les dades de l'e-niu, el nom que li han donat i l'estat. Hem creat per cada taula un camp id_ que serà la PRIMARY KEY, és a dir, un valor que no es pot repetir ja que identificarà cada valor de la taula.
- Persones: Conté les dades de cada persona com pot ser nom, password, email, etc.
- Niu: Els paràmetres són el nom, població, espècie d'ocell entre d'altres. També té l'id_pers de la persona que l'ha creat i l'id_eniu de l'e-niu que ha fet servir.
- Moviments: Aquí hi ha les dades del niu, activitats, dia, hora, a més del id_niu del niu que pertanyen els moviments.

Relacions: Observem a l'esquema que les relacions estan precedides d'un rombe vermell

- Relació eniu-niu: expressa el fet que un niu físic pot tenir diverses utilitzacions al llarg del temps. Això porta a definir-la del tipus "1 a n". El camp clau és id_eniu (clau primària de eniu), i la clau forana és id_eniu (clau de la taula niu).
- Relació persones-niu: idènticament a l'anterior. Relació del tipus "1 a n". El camp clau és id_pers.
- Relació niu-moviments: passa exactament igual que els anteriors casos, relació tipus "1 a n" amb el camp clau id_niu.

Índex: S'han definit els següents índex per millorar l'eficiència de les consultes SQL:

- La unió dels camps id_pers, id_eniu de la taula niu.
- Camp id_niu de la taula moviments.

La programació per crear la base de dades i les taules és la següent (el codi d'aquesta primera part de connexió l'explicarem més endavant quan s'explica la programació orientada a objectes (POO). La funció estableix connexió amb la base de dades MySQL):

```
$consultsql=new Consulta;  
$link=$consultsql->con;  
if (! $link){  
$avis1='ERROR: Imposible establir connexió amb el servidor';  
exit;  
}
```

Per crear la base de dades del MySQL amb el PHP ho fem enviant una consulta SQL amb la instrucció "CREATE DATABASE", que en el nostre cas a més notifica un error si no la pot crear:

```
if (! mysql_query("CREATE DATABASE $basedatos", $link)){
```

```
$avis2='ERROR2: Impossible crear base de dades query';  
exit;  
}
```

Ara per crear les diferents taules ho farem de la mateixa manera que creant la base de dades però amb una sentència que ens indica que el que volem crear és una taula. Fixem-nos que en la mateixa instrucció ja li indiquem quins són els camps que hi ha d'haver a dins la taula, com per exemple l'identificador d'aquesta taula, que serà un nombre enter, que no pot ser nul i que s'auto incrementa a mesura que es vagin entrant dades.

A més de l'identificador de la mateixa taula moviments, hi ha l'identificador dels nius "id_niu", que serà el que ens diu a quin niu correspon cada moviment (recordem que hem creat una base de dades relacional):

```
$sql = " CREATE TABLE `moviments` (  
  `id` int(11) NOT NULL auto_increment,  
  `accions` char(50) default NULL,  
  `dia` date default NULL,  
  `hora` char(15) default NULL,  
  `id_niu` int(11) NOT NULL,  
  PRIMARY KEY (`id`),  
  KEY `id` (`id_niu`));  
  
if (@mysql_db_query($basedatos, $sql, $link)) {  
  $avis1="La taula moviments s'ha creat amb èxit";  
} else {  
  $avis1="No s'ha pogut crear la taula moviments";  
}
```

De la mateixa manera anem creant la taules “persones”:

```
$sql = " CREATE TABLE `persones` (  
  `id_pers` int(11) NOT NULL auto_increment,  
  `nom` char(10) default NULL,  
  `cognom` char(10) default NULL,  
  `email` char(25) default NULL,  
  `dia` date default NULL,  
  `hora` char(15) default NULL,  
  `pass` char(10) default NULL,  
  PRIMARY KEY (`id_pers`))";  
  
if (@mysql_db_query($basedatos, $sql, $link)) {  
  $avis2="La taula persones s'ha creat amb èxit";  
} else {  
  $avis2="No s'ha pogut crear la taula persones";  
}
```

La taula “niu”:

```
$sql = " CREATE TABLE `niu` (  
  `id_niu` int(11) NOT NULL auto_increment,  
  `nomniu` char(10) default NULL,  
  `poblacio` char(10) default NULL,  
  `dia` date default NULL,  
  `especie` char(15) default NULL,  
  `id_pers` int(11) NOT NULL,  
  `id_eniu` int(11) NOT NULL,  
  PRIMARY KEY (`id_niu`),  
  KEY `id_niu` (`id_pers`, `id_eniu`))";
```

```
if (@mysql_db_query($basedatos, $sql, $link)) {  
    $avis3="La taula niu s'ha creat amb èxit";  
} else {  
    $avis3="No s'ha pogut crear la taula niu";  
}
```

I finalment la taula “eniu”:

```
$sql = " CREATE TABLE `eniu` (  
    `id_eniu` int(11) NOT NULL auto_increment,  
    `nomeniu` char(20) default NULL,  
    `estat` char(20) default NULL,  
    PRIMARY KEY (`id_eniu`));  
  
if (@mysql_db_query($basedatos, $sql, $link)) {  
    $avis4="La taula eniu s'ha creat amb èxit";  
} else {  
    $avis4="No s'ha pogut crear la taula eniu";  
}
```

Ara que ja hem comentat la base de dades, passem a mencionar quines han estat les eines, tècniques o programes utilitzats en la part de programació, és a dir, en les dues altres capes.

4.2.2 – PHP

Com qualsevol llenguatge de programació, el PHP consta d'unes variables, d'uns operadors i d'unes instruccions. Aquestes variables, per definició són uns contenidors d'informació i les fem servir per guardar resultats, com per exemple, quan hi ha una

sessió oberta, qui és l'usuari de la sessió, quin niu hem seleccionat, quin tipus de gràfic volem veure, etc.

Hi ha casos en que el valor d'aquestes variables l'hem hagut de passar a un altre arxiu, com per exemple, en una pàgina seleccionem el tipus de gràfic que volem veure i una altra mostra el gràfic seleccionat, de manera que aquesta segona pàgina necessita saber quin valor se li ha donat a la variable en la primera pàgina que indica el tipus de gràfic.

Els mètodes que s'ha fet servir per la transferència de variables entre arxius són els següents:

TRANSFERÈNCIA DE VARIABLES AMB EL MÈTODE GET: Amb aquest sistema passem les variables al indicar la direcció URL del link, és a dir, aquest conté la direcció de la pàgina web on hem d'anar seguida de les variables que hi hem de passar. En el projecte, unes de les variables que passem per URL és l'idioma seleccionat. De manera que la direcció del link on seleccionem l'idioma és:

`CATALA`

Aquest enllaç ens permet seleccionar l'idioma, retornant a la pàgina principal, que en aquest cas s'anomena "index.php" amb la variable "idioma" que porta assignat el valor "ca" de català.

En la pàgina destí, la forma de recuperar el valor de la variable és mitjançant la instrucció:

`$_GET['idioma']` o bé `$_REQUEST['idioma']`.

TRANSFERÈNCIA DE VARIABLES AMB EL MÈTODE POST: D'aquesta manera, el que fem és crear un formulari del tipus post en el HTML on ens indica diverses opcions per assignar a la variable, un cop feta la selecció premem un botó del tipus

submit que executa la transferència de la informació a l'arxiu destí. En la figura 4.2.2.1 es mostra el formulari del HTML on s'ha de triar el tipus de gràfic que volem observar. Com podem veure, les dades s'envien a la pàgina destí "graficsa.php" que és la que ens mostrarà el tipus de gràfic. La variable que portarà la informació del tipus de gràfic és la "GRA", i els possibles valors són: "lin", "rad" o "bar", que corresponen a un gràfic de línies, radar o de barres.

```
13
14 <form action="graficsa.php?idioma={%idioma}" method="post" >
15 <p align="left">LINIES<input type="radio" checked name="GRA" value="lin" ></p>
16 <p align="left">RADAR<input type="radio" name="GRA" value="rad"></p>
17 <p align="left">BARRES<input type="radio" name="GRA" value="bar"></p>
18 <input name="PROCESSAR" type="submit" />
19 </form>
20
```

Fig. 4.2.2.1

En l'arxiu destí recuperarem la informació de la variable amb la instrucció `$_POST['GRA']` o bé `$_REQUEST['GRA']`.

Com s'ha pogut observar, la instrucció `$_REQUEST['']` ens permet recuperar el valor de la variable en tots dos mètodes.

Hi ha casos en que el valor de les variables no només els volem passar al proper arxiu com en els casos anteriors, sinó que volem mantenir el valor durant tota la sessió, com pot ser el cas de l'usuari que ha obert la sessió, informació que volem mantenir mentre no ens desconnectem, de manera que per aquests casos l'eina que s'ha fet servir és la variable de sessió :

VARIABLES DE SESSIÓ: Aquesta és una opció que ens dona el PHP i ens permet definir unes variables que es mantindran actives mentre estiguem connectats. Per treballar amb aquestes variables només hem d'escriure la instrucció "session start();", i a partir d'aquí ja podem assignar els valors. En el proper exemple mostrem com s'assigna a aquestes variables la informació de si estem connectats o no i el nom de l'usuari.

```
session_start();  
$_SESSION['conectat']=true;  
$_SESSION['usuari']=$row['nom'];
```

Les variables de sessió també ens han estat de gran utilitat per mantenir l'idioma seleccionat, ja que un cop l'hem triat a la pàgina principal i s'ha assignat el valor pel mètode "GET", tal i com hem explicat anteriorment, llavors el posem a una variable de sessió i ens permetrà mantenir l'idioma durant tota la sessió. La manera que ho hem fet és la següent:

En el primer condicional, s'hi entra si la variable que ens ve amb el mètode "GET" té algun valor, és a dir, si algú ha seleccionat algun canvi d'idioma. Un cop a dins, passem aquest valor a la variable "\$idioma" que ens anirà bé pel tractament dins l'arxiu, i d'aquesta la passem a la variable de sessió que és la que ens mantindrà el valor mentre anem canviant de pàgines:

```
if($_REQUEST['idioma']!=""){  
    $idioma=$_REQUEST['idioma']; //AGAFEM L'IDIOMA  
    $_SESSION['idioma']=$idioma;  
}
```

El cas que no ens vingui donat un valor per la direcció URL, vol dir que ja tenim la sessió iniciada i que el valor per la variable "\$idioma" l'agafarem de la variable de sessió:

```
else $idioma=$_SESSION['idioma'];
```

Quan entrem en el programa per primer cop, la variable "\$idioma" no té cap valor perquè encara no li ha arribat cap valor pel mètode "GET". En aquest cas que serà la inicialització de l'idioma, forcem un valor a la variable de sessió i d'aquí a la variable "\$idioma", que en aquest cas ha estat el català.

```
if ($idioma=="")
{ $_SESSION['idioma']='ca';
  $idioma=$_SESSION['idioma'];
}
```

A partir d'aquí, ja tenim aquesta informació per totes les pàgines del programa.

PROGRAMACIÓ ORIENTADA A OBJECTES (POO): És una metodologia de treball molt estesa i actual, ja que consisteix en crear unes classes (podríem dir que són com uns patrons) que contenen uns atributs (variables) i uns mètodes (funcions). Un cop creada la classe creem (instanciem) un objecte d'aquesta classe i automàticament té els atributs i sap fer els mètodes definits en la classe.

Aquest tipus de programació ha estat molt útil per aquells casos en que una part de programa l'hem d'anar repetint en els diversos arxius, com és la comunicació amb la base de dades on hi ha una part constant i una altra que és la seqüència SQL, que ens variarà depenent de si hem de fer una consulta, una introducció de dades o borrar les dades.

El que hem fet ha estat crear la classe Consulta:

```
<?
```

Iniciem la classe:

```
class Consulta{
```

Definim les variables que necessitem:

```
var $host;
var $usuari;
```

```
var $password;  
var $basedades;  
var $sentencia;
```

Creem una funció constructor, que s'executa al instanciar un nou objecte de la classe consulta que ens inicialitza les variables i ja ens fa la connexió amb el servidor:

```
function  
Consulta($host="localhost",$usuari="root",$password="10manan10",$baseda  
des="dadniu",$sentsql=""){  
$this->host=$host;  
$this->usuari=$usuari;  
$this->password=$password;  
$this->basedades=$basedades;  
$this->sentsql=$sentsql;  
$this->con=mysql_connect($this->host, $this->usuari, $this->password);  
}
```

La propera funció executarà l'ordre de comunicació amb la base de dades, i la sentència SQL l'entrarem amb la variable "\$sentsql" quan haguem creat el nou objecte.

El resultat ens el retornarà amb la variable "\$result", que pot ser una variable simple o una taula dependent del resultat:

```
function executar($sentsql){  
$sql=$sentsql;  
$result=mysql_db_query($this->basedades,$sql);  
return $result;  
}
```

La funció que ve a continuació és com l'anterior però només executa la sentència, sense retornar el resultat en una variable:

```
function executarsol($sentsql){  
    $sql=$sentsql;  
    mysql_db_query($this->basedades,$sql);  
}  
}  
?>
```

En tots els arxius PHP que s'ha necessitat tractar amb la base de dades, s'ha instanciat un objecte d'aquesta classe i ja en quatre línies s'ha realitzat la operació. Per exemple, veurem com s'ha fet per borrar un niu.

Primerament, s'ha d'informar quin és l'arxiu que conté la classe:

```
require ("Consulta.php");
```

En la variable "\$idniu" hi tenim el niu seleccionat, i en la variable "\$s" hi posem la sentència SQL que volem executar. En aquest cas, al voler borrar el niu que ens indica la variable "\$idniu", la sentència queda:

```
$s="delete from niu where id_niu=".$idniu;
```

Llavors creem l'objecte de la classe Consulta:

```
$consultsql=new Consulta;
```

Finalment, un cop està creat l'objecte, se li indica quina funció de la classe ha d'executar i amb la variable "\$s" quina és la sentència:

```
$consultsql->executarsol($s);
```

La programació orientada a objectes també ha estat molt útil per a la creació dels gràfics que s'han fet utilitzant la llibreria eZ-Components, i per la utilització del Smarty, programa que ens ha permès enllaçar els arxius PHP amb les plantilles HTML com s'explica tot seguit.

EZ-COMPONENTS: Aquesta llibreria, és un conjunt de components o arxius referents a la representació gràfica que permeten crear, configurar i mostrar una gràfica de les dades que estem tractant. Passem directament a veure la part que s'ha fet servir en aquest treball per a poder mostrar gràficament els múltiples resultats dels e-nius.

Primer hem de tenir en compte que la informació que s'ha de representar a la gràfica, que en el nostre cas seran registres dels e-nius, ha d'estar emmagatzemada en una taula que s'anomena "\$wikient".

A partir d'aquí, veiem que el programa comença amb el condicional IF sobre si existeixen les variables LIN, BAR o RAD, que ens indiquen el tipus de gràfic que volem visualitzar, si amb línies, amb barres o del tipus radar. Depenent de la variable que es trobi activa, s'instanciarà l'objecte "\$graph" de la classe que correspongui al tipus de gràfic:

```
if($_REQUEST['RAD']){  
    $graph = new ezcGraphRadarChart();  
}  
if($_REQUEST['BAR']){  
    $graph = new ezcGraphBarChart();  
}
```

```

if($_REQUEST['LIN']){
    $graph = new ezcGraphLineChart();
}

```

Un cop instanciat l'objecte “\$graph”, anem assignant les variables com el títol, els drivers o el format d'imatge entre d'altres:

```

//$graph->title = $dat5.'-'. $dat6;
$graph->driver = new ezcGraphGdDriver();
$graph->options->font = 'tutorial_font.ttf';
$graph->driver->options->supersampling = 1;
$graph->driver->options->jpegQuality = 100;
$graph->driver->options->imageFormat = IMG_JPEG;

```

Amb la instrucció “foreach”, anem recorrent tota la taula “\$wikient” i anem creant un nou objecte per cada tipus de dades que hem de mostrar, com seran les entrades, les sortides, etc..

```

foreach ( $wikient as $dates => $data )
{
    $graph->data[$dates] = new ezcGraphArrayDataSet( $data );
}

```

Finalment, s'assignen les variables “w” i “h” a la funció “renderToOutput de l'objecte creat “\$graph”, on s'indica la mida que ha de tenir el gràfic final, i aquesta ja ens mostra el gràfic sol·licitat.

```

$graph->renderToOutput( $_REQUEST['w'], $_REQUEST['h']);

```


A la figura 4.2.2.2, 4.2.2.3 i 4.2.2.4 podem observar els diferents gràfics del nombre d'activitats de l'e-niu, segons s'hagi triat una representació en línies, barres o format radar.

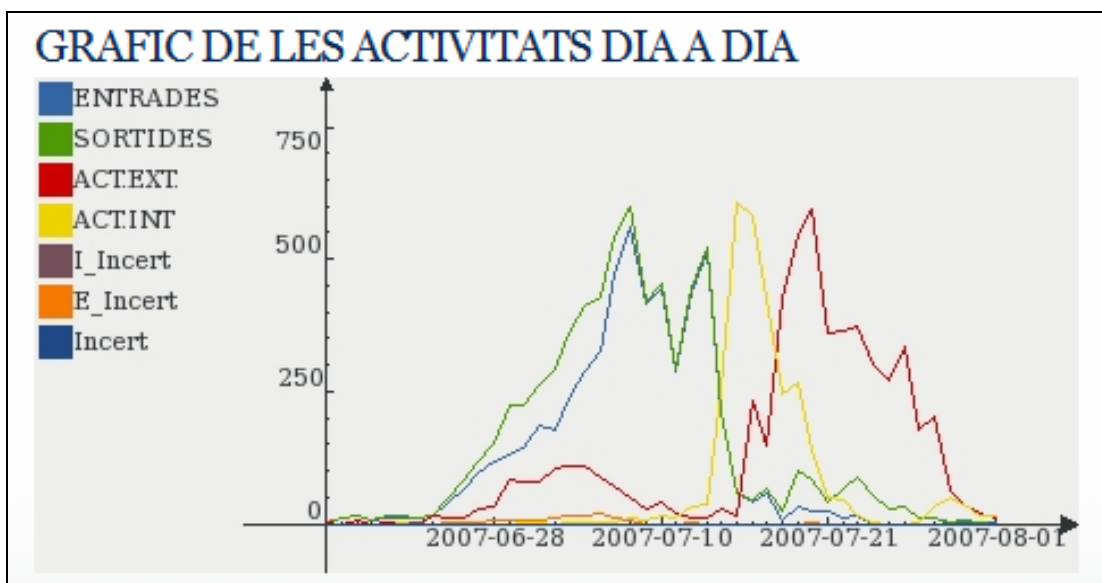


Fig. 4.2.2.2

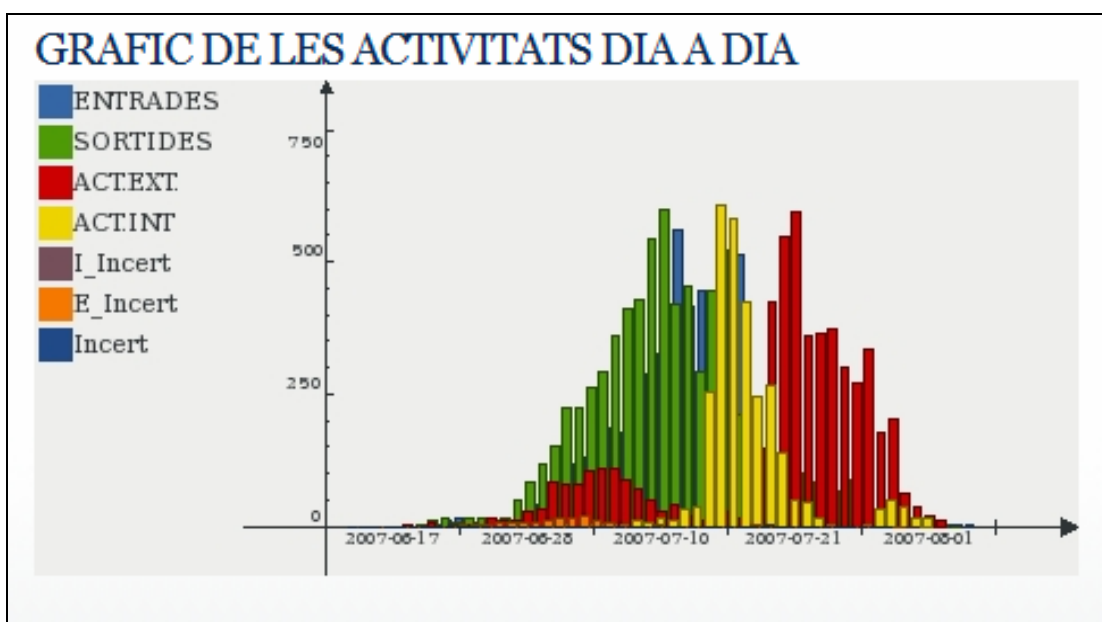


Fig. 4.2.2.3

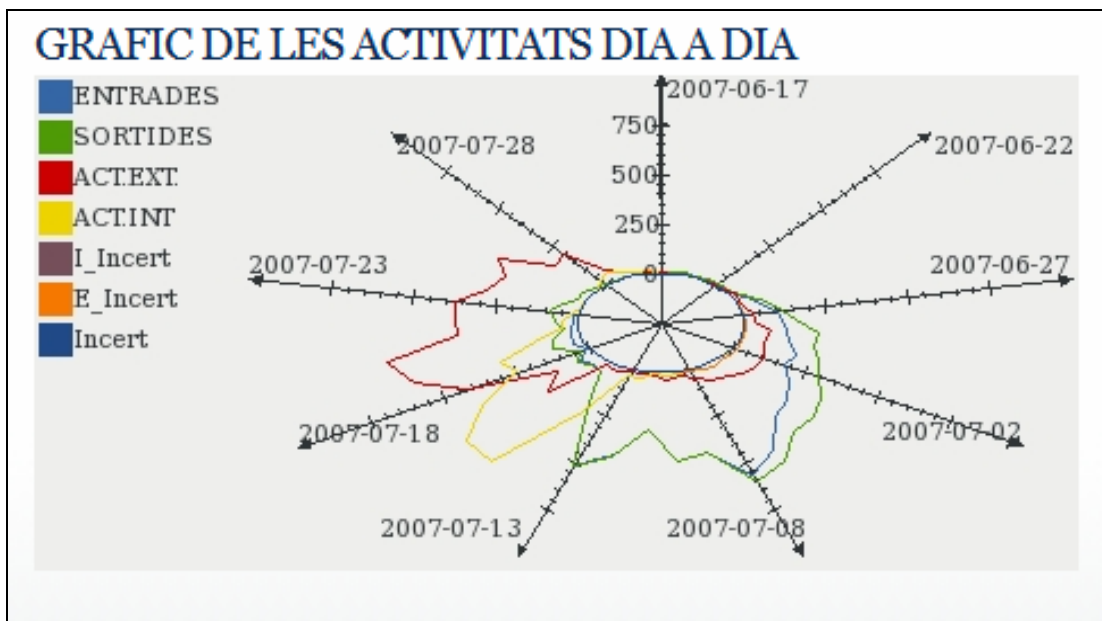


Fig. 4.2.2.4

FORMATS DISPONIBLES PER OBTENIR LES DADES: Un cop vista la llibreria EZ-COMPONENTS, parlarem dels tres tipus de formats que podem seleccionar per obtenir les dades, que són el tipus gràfic, el tipus taula i el tipus Excel.

Hem decidit una estructura comú per tots els processaments de dades que faci el programa. Aquesta consisteix primer en executar el processament i emmagatzemar les dades en un format que permet independitzar el resultat obtingut, de la comunicació amb l'usuari. L'etapa de sortida de dades l'hem pensat de tres maneres possibles: Excel, gràfic o taula, on podem veure en l'esquema del diagrama 4.2.2.1.

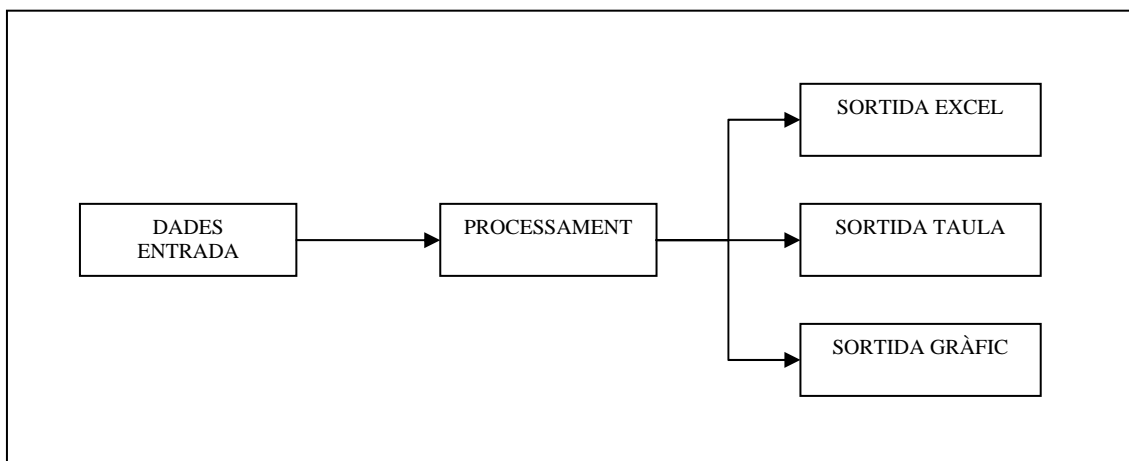


Diagrama 4.2.2.1

En la figura 4.2.2.5 podem veure com ens seran mostrades les dades en els tres formats:

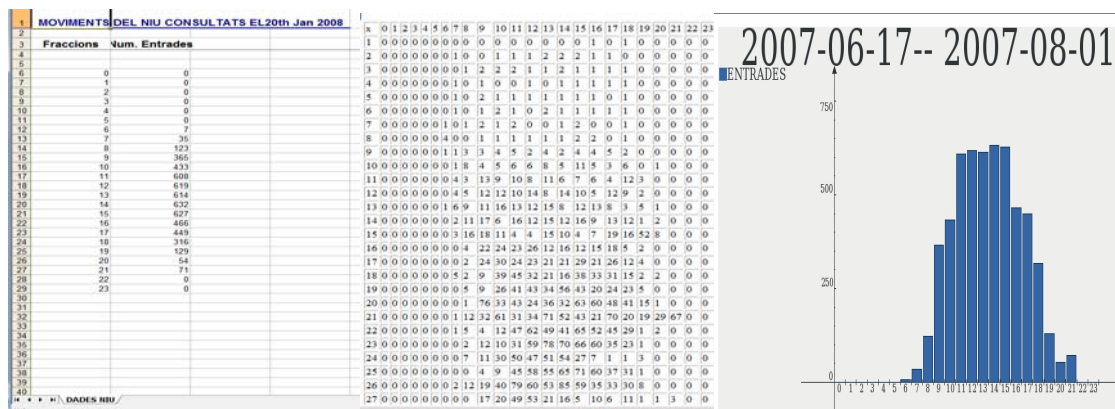


Fig. 4.2.2.5

El format de sortida el controlarem amb una variable que anomenarem “\$format”, i ens arriba al full que calcula la sortida de dades mitjançant el mètode “GET”.des de la pàgina que ens les mostrarà. Si el valor de la variable és “taula”, les dades ens les mostrarà en aquest format. Si el valor val “xls”, ens mostrarà les dades en format Excel, en canvi si no val res, ens mostrarà les dades en una gràfica, que serà la sortida predeterminada si no marquem cap format.

Primerament, els que s’ha de fer abans de mostrar les dades en qualsevol format és haver-les calculat, creant una taula on s’hi posen els resultats.

Per exemple, si tenim entrades i sortides, la taula podria ser com la següent:

```
Array([ENTRADES] => Array(
    [2007-06-17] => 1
    [2007-06-18] => 11
    [2007-06-19] => 14
    .
    .
    . )
[SORTIDES] => Array(
    [2007-06-17] => 3
    [2007-06-18] => 10
    [2007-06-19] => 16
    [2007-06-20] => 7
    .
    .
    . )
)
```

Si l'opció triada és mostrar la taula, el codi fet servir és el següent:

```
if ($_REQUEST['format'] == 'taula'){
```

La part de l'encapçalament i de la taula:

```
echo 'Num. Entrades diaries per hores [0-23]';
echo '<table align="center" border=1><tr>';
```

Bucle per mostrar la fila de les hores de 0 a 23:

```

for($k=-1;$k<='23';$k++){
    $s=$k;
    if($k==-1){
        $s=x;
    }
    echo '<td>'.$s."</td>\r\n";
}
</tr>

```

Bucle on anirem posant les dades que tenim en la taula anomenada "\$contadia". Anem fent un bucle per les files (dies) amb la variable "\$i", i dins de cada fila fem un bucle per les columnes (les hores de cada dia) amb la variable "\$j",

```

for($i=1;$i<=$diataula;$i++){
    echo '<tr><td>'.$i."</td>\r\n";
    for($j=0;$j<='23';$j++){
        echo '<td>'.intval($contadia[$i][$j])."</td>\r\n";
    }
    echo "</tr>\r\n";
}
echo "</table>\r\n";
}

```

La taula HTML finalitzada seria per exemple:

```

<table align="center" border=1>
<td>x</td>
<td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>
6</td>

```

```

<td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><
td>13</td>
<td>14</td><td>15</td><td>16</td><td>17</td><td>18</td>
.
.
. <td>0</td><td>0</td><td>0</td>
</tr>
</table>

```

Permetent-nos extreure la següent taula:

5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0
0	0	1	0	0	1	1	1	2	2	2	1	1	0	0	0	0	0	0
0	0	0	1	2	2	2	1	1	2	1	1	1	1	0	0	0	0	0
0	0	1	0	1	0	0	1	0	1	1	1	1	1	0	0	0	0	0
0	0	1	0	2	1	1	1	1	1	1	1	0	1	0	0	0	0	0
0	0	1	0	1	2	1	0	2	1	1	1	1	1	0	0	0	0	0
0	1	0	1	2	1	2	0	0	1	2	0	0	1	0	0	0	0	0
0	4	0	0	1	1	1	1	1	1	2	2	0	1	0	0	0	0	0
0	1	1	3	3	4	5	2	4	2	4	4	5	2	0	0	0	0	0
0	0	1	8	4	5	6	6	8	5	11	5	3	6	0	1	0	0	0
0	0	4	3	13	9	10	8	11	6	7	6	4	12	3	0	0	0	0
0	0	4	5	12	12	10	14	8	14	10	5	12	9	2	0	0	0	0
0	1	6	9	11	16	13	12	15	8	12	13	8	3	5	1	0	0	0
0	0	2	11	17	6	16	12	15	12	16	9	13	12	1	2	0	0	0
0	0	3	16	18	11	4	4	15	10	4	7	19	16	52	8	0	0	0
0	0	0	4	22	24	23	26	12	16	12	15	18	5	2	0	0	0	0
0	0	0	2	24	30	24	23	21	21	29	21	26	12	4	0	0	0	0
0	0	5	2	9	39	45	32	21	16	38	33	31	15	2	2	0	0	0
0	0	0	5	9	26	41	43	34	56	43	20	24	23	5	0	0	0	0
0	0	0	1	76	33	43	24	36	32	63	60	48	41	15	1	0	0	0
0	0	1	12	32	61	31	34	71	52	43	21	70	20	19	29	67	0	0
0	0	1	5	4	12	47	62	49	41	65	52	45	29	1	2	0	0	0

0	0	0	2	12	10	31	59	78	70	66	60	35	23	1	0	0	0	0
0	0	0	7	11	30	50	47	51	54	27	7	1	1	3	0	0	0	0
0	0	0	0	4	9	45	58	55	65	71	60	37	31	1	0	0	0	0
0	0	2	12	19	40	79	60	53	85	59	35	33	30	8	0	0	0	0
0	0	0	0	17	20	49	53	21	16	5	10	6	11	1	1	3	0	0
0	0	0	2	0	0	2	2	12	24	7	0	0	0	0	7	1	0	0
0	0	1	3	0	2	15	14	0	0	7	0	0	0	0	0	0	0	0
0	0	0	3	28	16	3	1	0	1	6	2	0	0	0	0	0	0	0
0	0	0	0	4	1	0	0	0	0	1	1	0	1	0	0	0	0	0
0	0	0	0	1	2	4	6	8	5	2	2	1	0	1	0	0	0	0
0	0	0	0	0	0	1	4	6	6	4	2	1	1	0	0	0	0	0
0	0	0	1	0	0	2	4	1	0	1	5	4	4	1	0	0	0	0
0	0	0	1	1	3	0	2	1	1	2	1	0	0	1	0	0	0	0
0	0	0	0	0	1	1	2	1	5	1	2	1	1	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
0	0	0	0	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0
0	0	0	1	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Taula 4.2.2.1

I amb la taula, ja podem efectuar qualsevol tipus de tractament o representació com la de la figura 4.2.2.6, que mostra el numero d'entrades que ha efectuat l'ocell en 45 dies repartides en les 24 hores del dia:

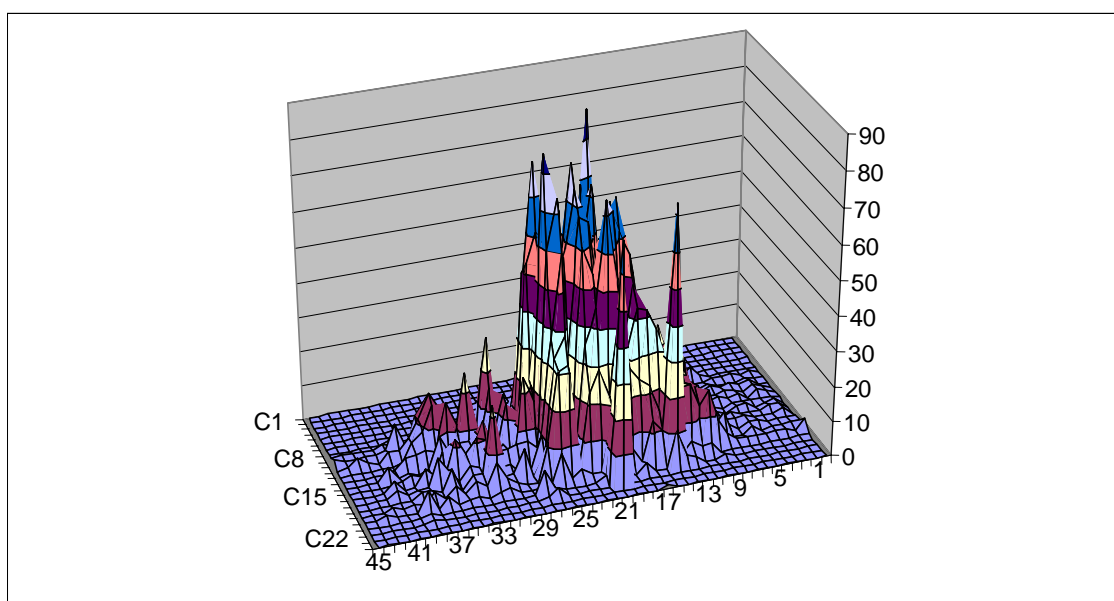


Fig. 4.2.2.6

Quan l'opció és mostrar les dades en format Excel, el codi de programació és el següent:

```
elseif ($_REQUEST['format']=='xls'){
```

Afegim una columna a la taula que ens indiqui quin és el dia de les dades:

```
$wikidata['datain']=$data2;
```

La programació d'aquesta part consisteix en crear un objecte de la classe Spreadsheet_Excel_Writer, que està en unes llibreries creades per la casa PEAR pensades per mostrar dades en Excel.

```
require_once "Spreadsheet/Excel/Writer.php";
```

```
$xls =& new Spreadsheet_Excel_Writer();
```


A continuació creem un full nou del llibre Excel i preparam el que serà el títol i els formats com el tipus de lletra:

```
$cart = & $xls->addWorksheet('DADES NIU');  
$titleText = 'MOVIMENTS DEL NIU CONSULTATS EL'. date('dS M Y');  
$titleFormat = & $xls->addFormat();  
$titleFormat->setFontFamily('Helvetica');  
$titleFormat->setBold();  
$titleFormat->setSize('13');  
$titleFormat->setColor('navy');  
$titleFormat->setBottom(2);  
$titleFormat->setBottomColor('navy');  
$titleFormat->setAlign('merge');  
$cart->write(0,0,$titleText,$titleFormat);  
$cart->write(0,1,"",$titleFormat);  
$cart->write(0,2,"",$titleFormat);  
$cart->write(0,3,"",$titleFormat);  
$cart->setRow(0,30);  
$cart->setColumn(0,3,15);  
$colHeadingFormat = & $xls->addFormat();  
$colHeadingFormat->setBold();  
$colHeadingFormat->setFontFamily('Helvetica');  
$colHeadingFormat->setSize('13');  
$colHeadingFormat->setAlign('center');  
$cart->writeRow(2,0,$colNames,$colHeadingFormat);  
$freeze = array(3,0,4,0);  
$cart->freezePanels($freeze);  
$currentRow = 0;
```

Un cop tenim els formats del full preparats, passem a extreure les dades de la taula i passar-les al full Excel. Finalment ja ens el mostrarà com a “DADESNIU.xls”:

```
foreach ( $wikient as $item )
{
    $cart->writeCol(5,$currentRow,$item);
    $currentRow++;
}
$xls->send("DADESNIU.xls");
$xls->close();
die();
}
```

En canvi, si el que volem és que ens mostri les dades en un gràfic, el que s'executa és el programa per representar gràfiques ja explicat del paquet de llibreries EZ-COMPONENTS.

4.2.3 – SMARTY

Un dels principals objectius que té aquest programa és poder separar el llenguatge de programació PHP amb el de presentació HTML. Cada vegada és més habitual que no sigui el mateix programador qui s'encarregui de fer les dues parts, i el què ens permet aquesta aplicació és lligar els dos llenguatges amb l'assignació de variables, però mantenint independents els dos codis, és a dir, el programador de PHP s'encarregarà de la gestió i del tractament de les dades sense ocupar-se de la presentació, assignant finalment uns valors a unes variables com a resultats, en canvi, per altra banda hi ha el programador de la presentació, que s'ha d'encarregar de mostrar en el format que més convingui les dades que li han estat assignades, sense tenir que preocupar-se del programa en PHP que li subministra aquestes dades.

La manera de funcionar del Smarty és que la pàgina que cridem amb el navegador no és la del template (HTML), sinó que crida el programa PHP, el que realitza el tractament de les dades. Llavors el PHP realitza les assignacions de variables al motor de la plantilla de HTML, mitjançant les instruccions que ens proporciona el Smarty i fa que aquest template (o plantilla) en HTML sigui el que ens mostri el navegador.

Les instruccions per treballar amb el Smarty que s'han fet servir en aquest projecte són les següents:

S'ha d'informar de l'arxiu que conté la classe que ens interessa:

```
require '../libs/Smarty.class.php';
```

Creem l'objecte de la classe Smarty amb algunes inicialitzacions:

```
$smarty = new Smarty;  
$smarty->compile_check = true;  
$smarty->debugging = true;
```

Abans de transferir les variables cap al HTML, s'ha de donar un valor a aquestes variables. En aquesta part del programa en concret, es mostra l'opció de triar l'idioma que s'ha fet en aquest programa. Segons l'idioma que s'ha seleccionat, les variables tindran un valor o un altre per mostrar en pantalla, i això es fa amb els condicionals IF:

```
if($idioma == 'ca'){  
    $titol = "E-niu";  
    $analisi="ANÀLISI";  
    $titol2='GESTIO ARXIUS';  
}  
if($idioma == 'es'){  
    $titol = "E-nido";
```

```

$analisi="ANALISIS";
$titol2='GESTION ARCHIVOS';
}

```

Un cop es tenen les variables preparades amb els valors depenent de l'idioma que s'ha seleccionat, s'assignen cap al HTML mitjançant la funció “assign” de l'objecte “Smarty”.

```

$smarty->assign("titol2",$titol2);
$smarty->assign("analisi",$analisi);
$smarty->assign("Titol",$titol);
$smarty->assign("idioma",$idioma);

```

Per exemple, podem seguir el camí en l'assignació de la variable nombrada com \$titol2 amb un valor de “GESTIÓ ARXIUS” i l'anirem encerclant en vermell.

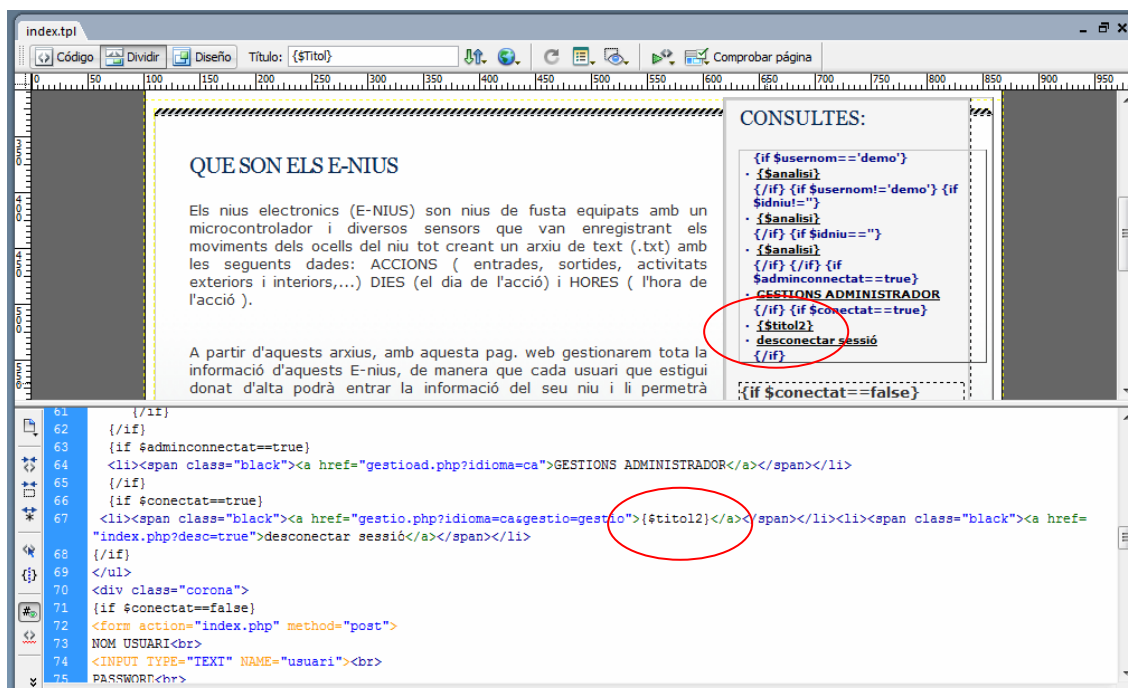


Fig. 4.2.2.7

Un cop s'ha efectuat l'assignació ja podem disposar d'ella en el template. Com podem veure a la figura 4.2.2.7 podem programar en la part de disseny (la part superior del programa), però quan comença a haver-hi tantes condicions es va embolicant aquesta part de disseny i acabem per programar en la part inferior, la part de codi HTML.

Podem veure el resultat de la variable \$titol2 en la presentació final ja mostrant el seu valor "GESTIÓ ARXIUS" en la figura 4.2.2.8.

S'ha de mencionar la diferència entre el quadre d'opcions anomenat consultes on hi ha el "GESTIO ARXIUS" encerclat de la presentació final de la figura 4.2.2.8, respecte el mateix quadre a la fase de disseny ple d'opcions i condicions on encara apareix com "\$titol2" en la figura 4.2.2.7.

E-niu

CATALA CASTELLA ANGLÉS FRANÇÉS

Sessió: ADMINISTRADOR-GENERAL

CONSULTES:

- ANALISI
- GESTIONS
- ADMINISTRADOR
- **GESTIO ARXIUS**
- desconnectar sessió

QUE SON ELS E-NIUS

Els nius electronics (E-NIUS) son nius de fusta equipats amb un microcontrolador i diversos sensors que van enregistrant els moviments dels ocells del niu tot creant un arxiu de text (.txt) amb les següents dades: ACCIONS (entrades, sortides, activitats exteriors i interiors,...) DIES (el dia de l'acció) i HORES (l'hora de l'acció).

A partir d'aquests arxius, amb aquesta pag. web gestionarem tota la informació d'aquests E-nius, de manera que cada usuari que estigui donat d'alta podrà entrar la informació del seu niu i li permetrà realitzar diversos anàlisi amb les gràfiques i estrene la informació en format Excel

FUNCIONAMENT DEL PROGRAMA:

Amb el menu superior podem sel.leccionar l'idioma de la pàgina i amb el menú de la dreta podem entrar en les diferents opcions, així com entrar en la nostra sessió si es que som usuaris donats d'alta. Per donar-nos d'alta com usuari s'haurà de contactar amb l'administrador.

Si entrem en la opció ANALISI, passarem a sel.leccionar un dels nius que tenim donats d'alta i podrem analitzar les dades així com veure les diverses gràfiques disponibles.

Fig. 4.2.2.8

Ara, amb la funció “display” el PHP selecciona l'arxiu que s'ha de mostrar en el navegador. En aquest cas s'ha creat una carpeta per a cada idioma possible, i dins d'aquestes carpetes hi tenim els templates o plantilles (arxiu de presentació) que corresponen a cada idioma, de manera que si s'observa la següent sentència, en el cas de que l'idioma seleccionat és el català, la variable “\$idioma” contindrà el valor “ca” i el navegador carregarà el template corresponent al PHP que hi hagi en la carpeta “/ca”.

```
$smarty->display($idioma.'/index.tpl');
```

En resum, a cada PHP li corresponen tants templates com idiomes es puguin seleccionar.

Referent als HTML, per definir la presentació dels seus documents s'ha optat per utilitzar els fulls d'estil en cascada, anomenats CSS (*Cascading Style Sheets*).

4.2.4 – CSS (Cascading Style Sheets)

Els fulls d'estil en cascada són un llenguatge que es fa servir per definir la presentació d'un document estructurat escrit en HTML. La idea del CSS és separar l'estructura del document i la seva presentació. En aquest projecte, el fet de treballar amb aquest sistema de programació ha permès crear el lloc web amb diverses pàgines, mantenint totes elles el mateix format i els mateixos estils, ja que totes elles estan lligades al mateix CSS, on defineix la mida del full, la mida i color de les lletres, la mida de l'encapçalament, etc.

La definició que fa el CSS d'aquest projecte del format general de la pàgina, dels marges, l'amplada i d'altres paràmetres és la següent:

```
#container {  
margin: 0 auto 0 auto;  
width: 965px;
```

```
background: #fff;  
padding: 0;  
font-size: 12px;  
border: 1px solid #000000;  
text-align: left;  
}
```

D'aquesta manera podem mantenir un format unificat per tots els arxius del programa, ja que si la l'amplada la reduïm de 965 a 765 pixels, ens reduirà l'amplada en la totalitat de les pàgines.

Un altre exemple és el de l'encapçalament, on ens defineix les característiques i a més quina és la fotografia que hi ha d'aparèixer. El codi és el següent:

```
#header {  
width: 765px;  
height: 180px;  
background: #eee url("images/anibanner01.jpg") no-repeat ;  
}
```

Aquesta part del CSS anomenada "header", ens defineix el format de l'encapçalament de la presentació, que el podem veure en la figura 4.2.2.9 marcada en groc, i la manera que es defineix en el codi HTML és la part subratllada en blau.

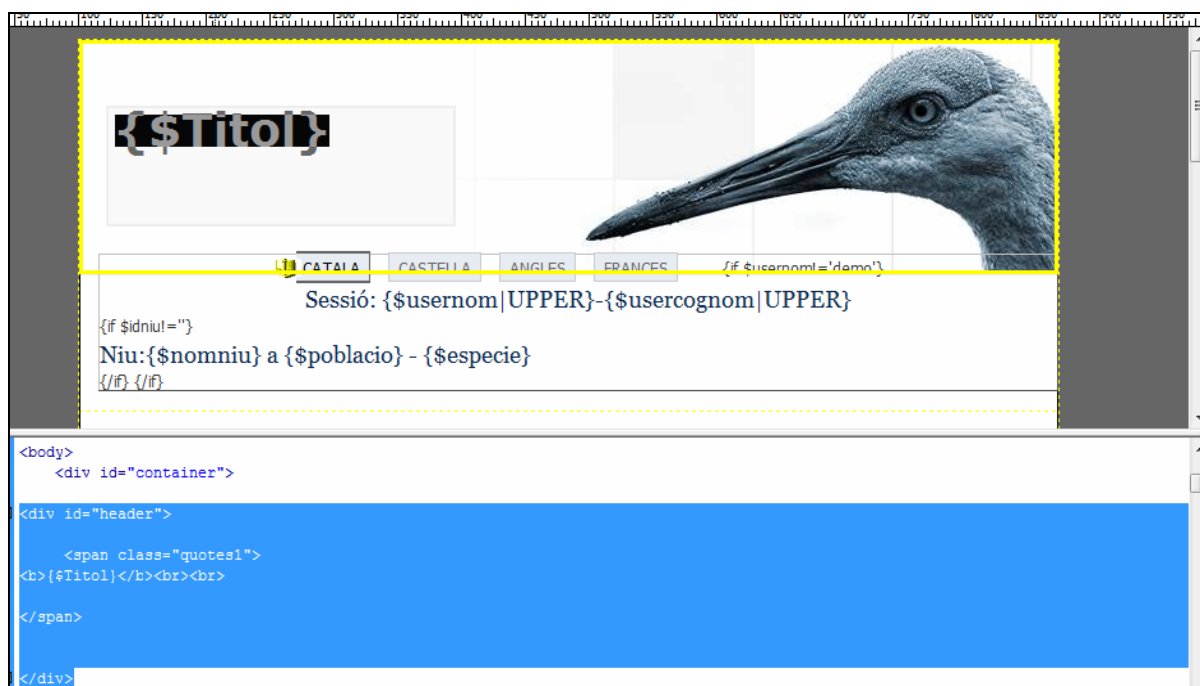


Fig.4.2.2.9

La manera d'indicar en el HTML quin és el full d'estils corresponent, es fa amb la sentència:

```
<link rel="stylesheet" type="text/css" href="templates/style.css"
media="screen" title="screen"/>
```

4.3 – ESTRUCTURA GENERAL DEL PROGRAMA

L'estructura del programa està representada en l'esquema del diagrama 4.3.1.

El programa s'inicia a la pàgina "home", on permetrà triar l' idioma i entrar el login de l'usuari. A partir d'aquí ja podem anar avançant mitjançant les opcions que ens mostra, com és el pas cap a l'anàlisi de dades. Per accedir a aquest anàlisi veiem que hi ha dos camins. Un que és directament i l'altre que és passant per la pàgina on ens deixa triar el niu. Aquest segon camí marcat amb un (1) a l'esquema és una opció que només apareix quan hi ha algun usuari connectat. De manera que quan l'usuari obre una sessió, per

poder anar a l'anàlisi de dades ha de passar per la pàgina on li deixa triar un dels nius que té donats d'alta (marquem en vermell les pàgines on només s'hi pot anar si hi ha algú connectat). Si no hi ha sessió oberta ja no apareix aquesta opció, es passa directament a l'anàlisi i podem analitzar un niu de mostra.

Un cop a la pàgina de l'anàlisi, ens mostra les opcions disponibles per seleccionar quin és l'anàlisi que es vol veure, opcions descrites en el punt 3 de la memòria (OBJECTIUS).

Al seleccionar una de les opcions, s'obre la pàgina on ja apareix el gràfic. Aquesta pàgina, permet triar diverses opcions del gràfic, com pot ser seleccionar un rang determinat, ja que al obrir-se mostrarà totes les dades que hi ha del niu que s'ha triat. Una altra opció que es pot triar és la del tipus de gràfic que ha de dibuixar, ja sigui de línies, de barres o del tipus radar. Depenent de la gràfica que volem visualitzar, també ens permetrà seleccionar quines de les activitats volem que entrin a l'anàlisi, com poden ser entrades, sortides, etc.

En aquesta pàgina hi ha un link que permet treure totes les dades que estem analitzant en format Excel. Opció molt interessant ja que passa les dades a un format que facilita molt fer qualsevol tractament posterior, així com qualsevol altre tipus de gràfic.

També tenim l'opció de treure en un altre full una ampliació del gràfic, i així poder tenir una visió més detallada d'aquest.

També hi ha la possibilitat de treure algunes de les dades en una taula.

A la pàgina "home", quan es compleix la condició de que hi ha un usuari connectat ens permet una altra opció a més de passar a l'anàlisi de dades, com és la gestió d'arxius (a l'esquema marcades en vermell totes les pàgines que necessiten aquesta condició). En aquest punt les opcions disponibles són donar d'alta un nou niu o borrar-ne un que ja estigui creat. Si la opció és donar d'alta un niu nou, passarem a una pàgina on podrem

posar la informació del tipus d'ocell, la població on es troba el niu i del nom que li vulguem posar. Seguidament passem a la pàgina on ja posem el nom de l'arxiu on hi ha les dades de l'e-niu, i automàticament l'introdueix en la base de dades.

Quan ens trobem a la pàgina principal o "home" i l'usuari que s'ha connectat sigui l'administrador, el programa mostra una nova opció que és on hi ha les gestions de l'administrador pròpiament dites (camí marcat amb un (2) en l'esquema i pàgines pintades en blau). Aquestes opcions consisteixen en donar d'alta un nou usuari, donar d'alta un nou e-niu fabricat, o marcar els e-nius ja existents com a lliures o ocupats. Aquestes opcions de l'estat dels e-nius són degudes a que quan un usuari vol donar d'alta un nou niu, el programa només li permet seleccionar els e-nius que han estat marcats com a lliures per l'administrador, ja que els altres o bé estaran ocupats o bé no estaran en servei per diversos motius com pot ser una reparació de l'e-niu.

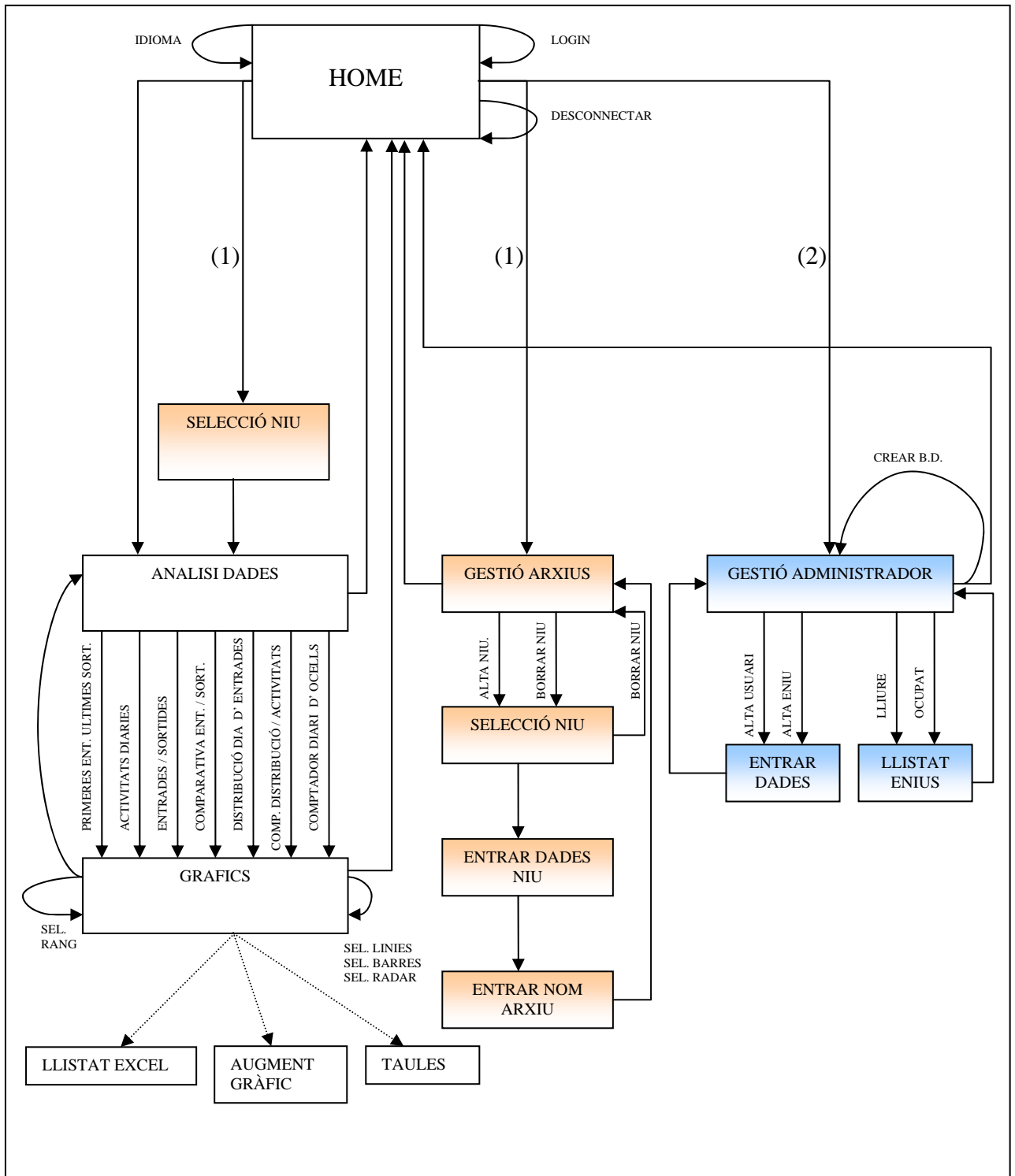


Diagrama 4.3.1

4.4 – DESCRIPCIÓ DETALLADA DELS ARXIUS DE GESTIÓ

En aquest apartat s'anirà explicant l'estructura interna de cada arxiu així com la definició de variables, explicació de bucles, condicionals, etc.

L'ordre que seguirem serà el mateix que hem fet servir a l'explicar l'estructura general del programa en el punt anterior. A continuació de l'explicació de cada arxiu de PHP, farem l'explicació de l'arxiu template o HTML que porta associat en els cassos que n'hi hagi, per mantenir una coherència en l'estructura.

4.4.1. – INDEX.PHP

Comencem amb les instruccions “require” on indiquem quines classes farem servir per crear objectes, seguides amb la creació de l'objecte “Smarty”, que com ja s'ha dit, serà el que ens lligarà el PHP amb el corresponent template.

A continuació tenim la part de l'idioma. En la primera entrada en el programa forçarem el català però les posteriors ja agafarà el valor de la variable de sessió si és que no marquem l'opció del canvi d'idioma. El codi ja l'hem vist a les variables de sessió.

Tot seguit, reinicialitzem una colla de variables de sessió, com són les que corresponen al niu seleccionat, o d'altres que afecten a seleccions de les gràfiques.com poden ser les dates del canvi de rang.

Ara arribem a un primer condicional “IF”. Aquest conté una part de codi que s'executarà en el cas de que ens arribi un valor del nom de l'usuari pel mètode “POST”, és a dir, que en la pàgina principal, s'hagi omplert la casella del nom d'usuari per connectar-se.

```
if($_POST['usuari']!=""){
```

Un cop a dins, creem un objecte per fer la consulta SQL a la base de dades, amb la instrucció de seleccionar les dades de l'usuari que coincideixin amb el nom i el password que ens arriba pel mètode "POST".

```
$consultsql=new Consulta;
$sql="select persones.nom, persones.cognom, persones.id_pers, persones.pass
from persones Where nom="$_POST['usuari']." And
pass="$_POST['password']."";
$result=$consultsql->executar($sql);
```

La instrucció "mysql_fetch_array()" ens permet passar el resultat de la consulta a una taula, en aquest cas l'anomenarem "\$row":

```
$row=mysql_fetch_array($result);
```

En el cas que no hi hagi cap usuari donat d'alta amb aquestes dades, maquem la variable de sessió que controla la connexió com a no connectat. Si existeix l'usuari, passem totes les seves dades que necessitem a la resta de pàgines a variables de sessió, i la variable que ens indica si hi ha connexió amb valor "true", ens indica que l'usuari s'ha connectat.

```
if ($row===false){
$_SESSION['conectat']=false;
}
else{
$_SESSION['usuari']=$row['nom'];
$_SESSION['cognom']=$row['cognom'];
$_SESSION['id_pers']=$row['id_pers'];
$_SESSION['pass']=$row['pass'];
$_SESSION['conectat']=true;
}
```

En el cas en que l'usuari connectat sigui l'administrador, també ho posem a variables de sessió, ja que com hem comentat, hi ha unes opcions específiques per a ell en determinades pàgines. De manera que tenim una variable de sessió específica per l'administrador.

```

if(($_POST['usuari']=='administrador')and($_POST['password']=='eni
u')){
    $_SESSION['adminconnectat']=true;
    $_SESSION['conectat']=true;
    $_SESSION['usuari']='Administrador';
    $_SESSION['cognom']='general';
}
}

```

Tot seguit tenim l'opció de desconnectar, on ens passa les variables de connexió a "false".

En el cas que no hi hagi connexió, passa a actuar com a demostració, és a dir, ens força la variable del nom de l'usuari a valor "demo", i ens borra totes les altres variables referents al niu i a l'usuari.

Seguidament entrem en la selecció de l'idioma, és a dir, que depenent de l'idioma, determinades variables tindran un valor o un altre com pot ser el títol o el nom de les opcions.

```

if($idioma == 'ca'){
    $titol = "E-niu";
    $analisi="ANÀLISI";
    $titol2='GESTIÓ ARXIUS';
}
if($idioma == 'es'){

```

```
$titol = "E-nido";  
$analisi="ANALISIS";  
$titol2='GESTION ARCHIVOS';  
}
```

L'últim pas que hem fet són les assignacions de valors cap al corresponent template, que en aquest cas és "l'index.tpl" de dins la carpeta que correspon a l'idioma:

```
$smarty->assign("usernom",$_SESSION['usuari']);  
$smarty->assign("usercognom",$_SESSION['cognom']);  
$smarty->assign("nomniu",$_SESSION['nomniu']);  
$smarty->assign("poblacio",$_SESSION['poblacio']);  
$smarty->assign("especie",$_SESSION['especie']);  
$smarty->assign("idniu",$_SESSION['idniu']);  
$smarty->assign("conectat",$_SESSION['conectat']);  
$smarty->assign("adminconnectat",$_SESSION['adminconnectat']);  
$smarty->assign("titol2",$titol2);  
$smarty->assign("analisi",$analisi);  
$smarty->assign("Titol",$titol);  
$smarty->assign("idioma",$idioma);  
$smarty->display($idioma.'/index.tpl');
```

4.4.2 – INDEX.TPL.

Aquesta és la part on ja es defineix la presentació de la pàgina que ens mostrarà el navegador.

Com s'ha explicat al definir els fulls d'estil, la part de presentació s'ha fet amb programes HTML formats per blocs de contingut basats en els estils definits en l'arxiu CSS. De manera que cada bloc de contingut portarà una etiqueta identificant l'estil que li correspon, com poden ser:

`<div id="container">` , `<div id="header">` , `<div id="navcontainer">` , `<div id="content">` , `<div class="corona">` , entre d'altres.

Per exemple, l'id container ens defineix tot el full, mides del full, contorns, etc. . L'id header és l'encarregat de definir la part de l'encapçalament, amb la imatge inclosa. En aquesta part és on hi posem la variable que ens dóna el títol. El programa en el HTML és:

```
<div id="header">
    <span class="quotes1">
        <b>{$Titol}</b><br><br>
    </span>
</div>
```

Després ens trobem que hem definit un segon bloc (id navcontainer) on hi ha els links per la selecció dels idiomes, la informació de l'usuari i el niu triat en la sessió. Fixem-nos amb els condicionals fets servir on les dades de l'usuari apareixeran sempre i quan el nom de l'usuari sigui diferent de "demo", això es produeix quan hi ha un usuari connectat. De la mateixa manera es fa amb les dades del niu, no el mostra si no hi ha un niu seleccionat. El codi és el següent:

```
<div id="navcontainer">
    <ul id="navlist">
        <li id="active"><a href="index.php?idioma=ca"
id="current">CATALA<span>Obrim la pàgina de gràfics.</span></a></li>
        <li><a href="index.php?idioma=es">CASTELLA<span>This is the
link to consecetuer.</span></a></li>
        <li><a href="index.php?idioma=en">ANGLES<span>Anyone for
dignissim qui blandit?</span></a></li>
        <li><a href="index.php?idioma=fr">FRANCES<span>Did you vero
your eros et today?</span></a></li>
```



```

    {if $usernom!='demo'}
    <li>
    <h2 class="noprnt">Sessió: {$usernom/UPPER}-
    {$usercognom/UPPER}</h2>
    </li>
    {if $idniu!=""}
    <h2 class="noprnt">Niu:{$nomniu} a {$poblacio} - {$especie}</h2>
    {/if}
    {/if}
  </ul>
</div>

```

L'altra part de programació que val la pena comentar és el quadre de les opcions a triar. Aquestes opcions varien segons si hi ha connexió, o si és l'administrador el qui està connectat.

El link ANÀLISI depèn de diverses condicions. Si no tenim cap usuari connectat (\$usernom='demo'), ens envia a la pàgina "analisi.php" que és on hi ha les opcions del gràfic. Si hi ha un usuari connectat llavors apareix una altra condició i és que si tenim un niu seleccionat ens envia igualment a la pàgina "analisi.php" per fer la selecció d'opcions del nostre niu. En el cas que hi ha connexió i no hi ha cap niu seleccionat, aquest link ens envia a la pàgina "analisi.php" que és on ens deixarà triar un dels nius que aquest usuari té donats d'alta i posteriorment anirem a "l'analisi.php" .

Ho hem fet d'aquesta manera:

```

<div class="float-right">
<h2 class="noprnt">CONSULTES:</h2>
<ul>
    {if $usernom=='demo'}
        <li><span class="black"><a
        href="analisi.php?idioma=ca">{$analisi}</a></span></li>

```

```

{/if}
{if $usernom!='demo'}
    {if $idniu!=""}
        <li><span class="black"><a
            href="analisi.php?idioma=ca">{$analisi}</a></span></li>
    {/if}
    {if $idniu=""}
        <li><span class="black"><a
            href="analisi.php?idioma=ca">{$analisi}</a></span></li>
    {/if}
{/if}

```

Pel que fa a la resta d'opcions, si hi ha un usuari connectat ens mostra un nou link amb la variable “\$titol2” que és la “GESTIO ARXIUS” i ens porta a l'arxiu “gestió.php” on tenim les opcions de gestió de l'usuari com donar d'alta un niu o borrar-ne un. També apareix l'opció de desconnectar. Aquesta és un link que ens retorna a la mateixa pàgina “index.php”, però mitjançant el mètode “GET”, activa la variable que ens desconnectarà.

En el cas que l'usuari sigui l'administrador, ens mostra una nova opció. Aquest nou link ens porta a la pàgina “gestioad.php”, i és on l'administrador pot crear les bases de dades, donar d'alta els usuaris, donar d'alta els e-nius i marcar la disponibilitat d'aquests.

```

{if $adminconnectat==true}
    <li><span class="black"><a
        href="gestioad.php?idioma=ca">GESTIONS
        ADMINISTRADOR</a></span></li>
{/if}
{if $conectat==true}

```

```

</li><span class="black"><a
href="gestio.php?idioma=ca&gestio=gestio">{$titol2}</a></span></l
i></li><span class="black"><a
href="index.php?desc=true">desconnectar sessió</a></span></li>
{/if}
</ul>

```

Ja per acabar amb la part HTML d'aquest arxiu, comentar la part on ens permet connectar-nos. Això ho hem fet amb un formulari del tipus post i només apareix quan no hi ha cap usuari connectat, quan la variable “\$conectat” té el valor “false”:

```

<div class="corona">
  {if $conectat==false}
    <form action="index.php" method="post">
      NOM USUARI<br>
      <INPUT TYPE="TEXT" NAME="usuari"><br>
      PASSWORD<br>
      <INPUT TYPE="password" NAME="password"><br>
      <INPUT TYPE="SUBMIT" value="Login">
    </form>
  {/if}
</div>

```

4.4.3 – PRESENTACIÓ ASSOCIADA A L'INDEX.PHP.

En la figura 4.4.3.1, veiem la presentació d'aquesta pàgina principal. Podem veure com encara no hi ha cap usuari connectat, fixem-nos que hi ha l'opció d'entrar el login i només permet anar a l'ANÀLISI. .

E-niu

CATALA CASTELLA ANGLÉS FRANCES

QUE SON ELS E-NIUS

Els nius electronics (E-NIUS) son nius de fusta equipats amb un microcontrolador i diversos sensors que van enregistrant els moviments dels ocells del niu tot creant un arxiu de text (.txt) amb les següents dades: ACCIONS (entrades, sortides, activitats exteriors i interiors,...) DIES (el dia de l'acció) i HORES (l'hora de l'acció).

A partir d'aquests arxius, amb aquesta pag. web gestionarem tota la informació d'aquests E-nius, de manera que cada usuari que estigui donat d'alta podrà entrar la informació del seu niu i li permetrà realitzar diversos anàlisi amb les gràfiques i estrene la informació en format Excel

FUNCIONAMENT DEL PROGRAMA:

Amb el menu superior podem sel.leccionar l'idioma de la pàgina i amb el menú de la dreta podem entrar en les diferents opcions, així com entrar en la nostra sessió si es que som usuaris donats d'alta. Per donar-nos d'alta com usuari s'haurà de contactar amb l'administrador.

Si entrem en la opció ANALISI, passarem a sel.leccionar un dels nius que tenim donats d'alta i podrem analitzar les dades així com veure les diverses gràfiques disponibles.

CONSULTES:

ANALISI

NOM USUARI

PASSWORD

Login

Fig. 4.4.3.1

Si obrim sessió veurem com canvien les opcions a més d'indicar quin és l'usuari connectat. Ens connectarem com a administrador i així ens permetrà veure totes les opcions disponibles. En la part esquerra superior a sota la selecció de l'idioma, ens apareix l'usuari connectat, i en les consultes apareixen les opcions de “GESTIÓ D'ARXIUS”, “GESTIONS D'ADMINISTRADOR” i l'opció de “DESCONNECTAR SESSIÓ”. Aquest nou cas el mostrem a la figura 4.4.3.2.



E-niu

CATALA CASTELLA ANGLÉS FRANCES

Sessió: ADMINISTRADOR-GENERAL

CONSULTES:

- **ANALISI**
- **GESTIONS ADMINISTRADOR**
- **GESTIO ARXIUS**
- **desconnectar sessió**

QUE SON ELS E-NIUS

Els nius electronics (E-NIUS) son nius de fusta equipats amb un microcontrolador i diversos sensors que van enregistrant els moviments dels ocells del niu tot creant un arxiu de text (.txt) amb les següents dades: ACCIONS (entrades, sortides, activitats exteriors i interiors,...) DIES (el dia de l'acció) i HORES (l'hora de l'acció).

A partir d'aquests arxius, amb aquesta pag. web gestionarem tota la informació d'aquests E-nius, de manera que cada usuari que estigui donat d'alta podrà entrar la informació del seu niu i li permetrà realitzar diversos anàlisi amb les gràfiques i estrene la informació en format Excel

FUNCIONAMENT DEL PROGRAMA:

Amb el menu superior podem sel.leccionar l'idioma de la pàgina i amb el menú de la dreta podem entrar en les diferents opcions, així com entrar en la nostra sessió si es que som usuaris donats d'alta. Per donar-nos d'alta com usuari s'haurà de contactar amb l'administrador.

Si entrem en la opció ANALISI, passarem a sel.leccionar un dels nius que tenim donats d'alta i podrem analitzar les dades així com veure les diverses gràfiques disponibles.

Fig. 4.4.3.2

Un cop vista la metodologia de programació en la pàgina principal, passem directament a comentar la pàgina que ens mostrarà les gràfiques ja que és on hi ha potser més feina de programació degut a que hi poden haver més condicions. Això es deu a que hem programat de manera que un sol arxiu “graficsa.php” amb el seu corresponent “graficsa.tpl” ens mostri qualsevol de les opcions de dades triades, és a dir, que l’estructura de la pàgina va variant segons l’opció seleccionada.

4.4.4 – GRAFICSA.PHP

La part PHP d'aquesta pàgina es centra pràcticament en l'assignació de les múltiples variables que necessita l'arxiu TPL. Aquest TPL és on realment hi ha hagut la feina forta de programació pel fet de voler mantenir el mateix arxiu per les set seleccions diferents de dades possibles, amb les diferents opcions dins de cadascuna. Anem doncs a parlar directament de la part HTML.

4.4.5 – GRAFICSA.TPL

En aquesta pàgina ja no hi ha les opcions per seleccionar l'idioma que hem vist en la pàgina principal. En lloc d'aquests hi ha els links per anar a la pàgina on seleccionarem l'anàlisi de les dades i per tornar a la pàgina principal.

La resta de la pàgina la podem dividir en dos grans parts, la que seleccionem les opcions per veure les dades i la que ens mostra les dades:

QUADRE DE SELECCIÓ D'OPCIONS:

En el quadre d'opcions, comencem per mostrar l'usuari, el niu seleccionat i la primera i última data dels moviments del niu. Si ens fixem en el codi, aquesta primera part només ens la mostrarà en determinades gràfiques, ja que en d'altres, la quantitat d'opcions és tant gran que aquesta informació s'ha passat a un altre lloc de la pàgina.

```
{if($tipusgraf=='rang2')or($tipusgraf=='compt')or($tipusgraf=='fracc2')or($tipusgraf
=='rang')or($tipusgraf=='fracc')}
    {if $usernom!='demo'}
        <h2 class="noprint">Sessió: {$usernom/UPPER}-
        {$usercognom/UPPER}</h2>
    }

```

```
{/if}
```

```

    {if $idniu!=""}
        <h2 class="noprint">Niu:{$nomniu} - {$poblacio} - {$especie} </h2>
        _____
        <h2 class="noprint">{$primeradat}</h2>
        <h2 class="noprint">{$ultimadat} </h2>
        _____
    {/if}
{/if}

```

A continuació hi ha la part que ens deixa triar el tipus de gràfic a visualitzar, ja sigui de línies, de barres o del tipus radar. Ho farem mitjançant un formulari del tipus “POST”. Aquesta opció només apareix en unes determinades gràfiques, de manera que en el codi ja hi veurem la condició prèvia, que ens ho mostrarà mentre no siguin determinades gràfiques, en concret quan no siguin les anomenades: “rang”, “rang2”, “fracc2” i “compt”:

```

<form action="graficsa.php?idioma={$idioma}" name="fgraf" method="post" >
{if $tipusgraf!='rang'}
{if $tipusgraf!='rang2'}
{if $tipusgraf!='fracc2'}
{if $tipusgraf!='compt'}
    <br><h2 class="noprint">TIPUS GRAFIC:</h2><br>
    <blockquote>
    {if $gra==""}
        <p align="left">LINIES<input type="radio" value="lin" checked
name="GRA"></p>
        <p align="left">RADAR<input type="radio" name="GRA"
value="rad"></p>
        <p align="left">BARRES<input type="radio" name="GRA"
value="bar"></p>
    {/if}

```

```

{if $gra=='rad'}
    <p align="left">LINIES<input type="radio" value="lin"
    name="GRA"></p>
    <p align="left">RADAR<input type="radio" checked name="GRA"
    value="rad"></p>
    <p align="left">BARRES<input type="radio" name="GRA"
    value="bar"></p>
{/if}
{if $gra=='bar'}
    <p align="left">LINIES<input type="radio" value="lin"
    name="GRA"></p>
    <p align="left">RADAR<input type="radio" name="GRA"
    value="rad"></p>
    <p align="left">BARRES<input type="radio" checked name="GRA"
    value="bar"></p>
{/if}
</blockquote>
{/if}{/if}{/if}{/if}

```

Fixem-nos que fem un bucle entre l'arxiu PHP i el TPL amb la variable “\$gra”. Això ho fem perquè un cop hem seleccionat el format del gràfic, el PHP ens retornarà la variable “\$gra” amb aquest valor. Suposem el “bar”, llavors amb aquest valor el TPL ens mostra el formulari de les opcions de triar el format del rang ja amb l'opció marcada de “bar”, és a dir, un cop hem seleccionat un tipus gràfic, ens va mantenint aquest tipus seleccionat mentre no en marquem un altre. En el cas que no s'hagi seleccionat cap tipus de format de gràfic, ens quedarà el de línies.

Una altra opció és la que podem seleccionar quines són les dades que volem analitzar dins del tipus d'anàlisi triat, amb això volem dir que si per exemple estem analitzant les activitats del niu, podem seleccionar quina de les activitats en concret volem que ens mostrin amb els resultats, com seran entrades, sortides, etc.

També hi ha l'opció de canviar la data inicial i final per efectuar l'anàlisi, així ens podem centrar el nostre estudi en un rang més concret. Un cop s'ha seleccionat una data diferent a l'inicial, el programa ja no ens mostra més el formulari per canviar de dates, sinó que ens mostra una casella que hem de marcar per tornar a les condicions inicials, modificant la variable "RAN".

En aquest punt, hem de fer menció que en l'opció de seleccionar les dates per canviar el rang, hi hem associat un javascript, de manera que al clicar sobre el botó, ens obre un calendari per poder triar la data.

Al variar el menú d'opcions, segons el tipus d'anàlisi seleccionat, mostrarem com s'ha fet el programa per triar les opcions de la gràfica de les primeres entrades i últimes sortides per agafar un exemple:

```
{if $tipusgraf=='priment'}
  <br><h2 class="noprint">MOSTRAR DADES:</h2><br>
  <p align="left">PRIMERES ENTRADES<input type="checkbox" name="ENT"
  value="ON"></p>
  <p align="left">ULTIMES SORTIDES<input type="checkbox" name="SORT"
  value="ON"></p>
  {if $rangmodif!='si'}
    <br><h2 class="noprint">TRIAR RANG:</h2><br>
    DATA INICI (A-M-D)
    <INPUT TYPE="TEXT" NAME="dat1" id="dat1">
    <br>
    <INPUT TYPE="button" VALUE="Calendari"
    onClick="newWindow('dat1')">
    DATA FI (A-M-D)
    <INPUT TYPE="TEXT" NAME="dat2" id="dat2"><br>
    <br>
```

```

<INPUT TYPE="button" VALUE="Calendari"
onClick="newWindow('dat2')">
{/if}
{/if $rangmodif=='si'}
<p align="left">RANG INICIAL<input type="checkbox" name="RAN"
value="ON"></p>
{/if}
{/if}

```

Aquest exemple en el navegador ho veurem tal com ens mostra la figura 4.4.5.1. En la figura 4.4.5.2, mostra la selecció d'opcions de la gràfica on es compara les entrades i sortides de dos rangs, així podem comparar com fem variar el menú d'opcions segons la gràfica:

TIPUS GRAFIC:

LINIES

RADAR

BARRES

MOSTRAR DADES:

PRIMERES ENTRADES

ULTIMES SORTIDES

TRIAR RANG:

DATA INICI (A-M-D)

DATA FI (A-M-D)

Fig.4.4.5.1

Sessió:
ADMINISTRADOR-GENERAL

Niu: abell - vic - abellerols

2007-06-17
 2007-08-01

TRIAR RANG GRAFICA 1:

DATA INICI 1 (A-M-D)
 2007-06-17
 DATA FI 1 (A-M-D)
 2007-08-01

TRIAR RANG GRAFICA 2:

DATA INICI 2 (A-M-D)

 DATA FI 2 (A-M-D)

Fig. 4.4.5.2

El javascript inclòs en la pàgina per mostrar un calendari, que passa el valor a les caselles del nostre formulari tipus "POST" de selecció de dates, és la següent:

```
<SCRIPT TYPE="text/javascript" LANGAUGE="JavaScript">
var camp_nom = "";
function y2k(number) { return (number < 1000) ? number + 1900 : number; }
var today = new Date();
var day = today.getDate();
var month = today.getMonth();
var year = y2k(today.getYear());
function padout(number) { return (number < 10) ? '0' + number : number; }
function restart() {
document.getElementById(camp_nom).value=" + year + '-' + padout(month - 0
+ 1) + '-' + padout(day);
mywindow.close();
}
function newWindow(camp_nom_func) {
camp_nom = camp_nom_func;
mywindow=open('cal.htm','myname','resizable=no,width=350,height=270');
mywindow.location.href = 'cal.htm';
if (mywindow.opener == null) mywindow.opener = self;
}
</SCRIPT>
```

I en el navegador ens obre una petita pagina amb un calendari per seleccionar la data.



Fig. 4.4.5.3

REPRESENTACIÓ DE RESULTATS:

Per mostrar-nos els resultats, ja hem comentat que podien ser de tres formes diferents, gràfic, Excel o taula.

Els càlculs per mostrar els tres tipus estan en l'arxiu PHP encarregat de fer tot el tractament per cada tipus d'anàlisi, de manera que el que fem en aquest TPL és crear uns enllaços a la pàgina del tractament de les dades, i que ens mostri els resultats en el format que li demanem.

Totes les opcions que triem per veure els resultats els hi passem pel mètode "GET", de manera que la direcció URL de l'enllaç va acompanyada de múltiples variables com son el tipus de gràfic, les dates modificades, el tipus de acció a mostrar en l'anàlisi, així com la forma en que ens ha de mostrar els resultats.

Teníem el problema que la informació per passar en variables cap a la pàgina de tractament la teníem de diverses formes i l'havíem d'agrupar, ja que algunes surten d'un formulari "POST" com les opcions i d'altres les passem juntament amb la direcció amb el mètode "GET" com el format per rebre les dades.

Llavors el que s'ha pensat és que totes les variables de les opcions les passem des de "graficsa.tpl" que estem, cap al "graficsa.php" amb el mètode "POST", de manera que aquest segon, automàticament ens fa les assignacions d'aquestes variables i ens torna al graficsa.tpl amb el Smarty. Podent d'aquesta manera ajuntar totes les variables (les de cada enllaç en particular i les generals que venen assignades), així per exemple, per mostrar una gràfica, hi ha la direcció URL de l'arxiu que la calcula juntament amb les variables característiques del tipus d'anàlisi que estem efectuant més les variables que venen assignades.

Observarem que per la programació hem fet servir quatre crides a la direcció on ens calcula els resultats, una és per mostrar la gràfica, la segona és per mostrar la gràfica en un full apart i més ampliada, la tercera és per mostrar les dades en format Excel i la quarta per mostrar les dades en format taula. Com a exemple veurem els links de l'anàlisi de l'hora del dia que es produeixen les entrades (la variable "\$tipusgraf" identifica aquest tipus d'anàlisi amb el valor "fracc". Enllaça amb l'arxiu "grafracc.php") :

```
{if $tipusgraf=='fracc'}
```

Enllaç per l'ampliació:

```
<a href="http://127.0.0.1/ezcomponents-2007.1.1/ezcomponents-2007.1.1/Graph/docs/tutorial/grafracc.php?BAR={ $bar }&RAD={ $radX }&RAN={ $ran }&LIN={ $linX }&dat1={ $dat1 }&dat2={ $dat2 }&dat3={ $dat3 }&TIPUSGRAF={ $tipusgraf }&w=800&h=450" target="_blank">
```

Enllaç per mostrar gràfica normal:

```
<img src= "http://127.0.0.1/ezcomponents-2007.1.1/ezcomponents-
2007.1.1/Graph/docs/tutorial/graffracc.php?BAR={$bar}&RAD={$radX}&RA
N={$ran}&LIN={$linX}&dat1={$dat1}&dat2={$dat2}&dat3={$dat3}&TIPUS
GRAF={$tipusgraf}&w=460&h=220" width="460" height="220" border="0"
/></a>
```

Enllaç per format Excel:

```
[<a href="http://127.0.0.1/ezcomponents-2007.1.1/ezcomponents-
2007.1.1/Graph/docs/tutorial/graffracc.php?BAR={$bar}&RAD={$radX}&RA
N={$ran}&dat1={$dat1}&dat2={$dat2}&dat3={$dat3}&LIN={$linX}&SORT
={$SortOnOff}&ENT={$EntOnOff}&NUMENT={$numentX}&NUMSOR={$nu
msorX}&NUMACTEXT={$numactextX}&NUMACTINT={$numactintX}&INCE
RT={$incertX}&TIPUSGRAF={$tipusgraf}&format=xls"
target="_blank">Llistat dades EXCEL</a>]
```

Enllaç per format taula:

```
[<a href="http://127.0.0.1/ezcomponents-2007.1.1/ezcomponents-
2007.1.1/Graph/docs/tutorial/grafactiv.php?RAD={$radX}&RAN={$ran}&dat1
={$dat1}&dat2={$dat2}&LIN={$linX}&SORT={$SortOnOff}&ENT={$EntOn
Off}&NUMENT={$numentX}&NUMSOR={$numsorX}&NUMACTEXT={$num
actextX}&NUMACTINT={$numactintX}&INCERT={$incertX}&TIPUSGRAF=
{$tipusgraf}&format=taula" target="_blank">Taula entrades diaries per
hores</a>]
```

{/if}

4.5 – ESTRUCTURA DELS ARXIUS DE TRACTAMENT DE DADES

Hi ha sis arxius que són estrictament de tractament de dades, càlcul de resultats i preparació per la presentació d'aquests en el format corresponent. Aquests arxius en PHP no tenen el corresponent template de presentació, ja que són mostrats en el `graficsa.tpl` que acabem de veure, encarregat de totes les presentacions de resultats.

Anem a veure un esquema i una petita presentació de cada un:

4.5.1 – PRIMERES ENTRADES I ÚLTIMES SORTIDES

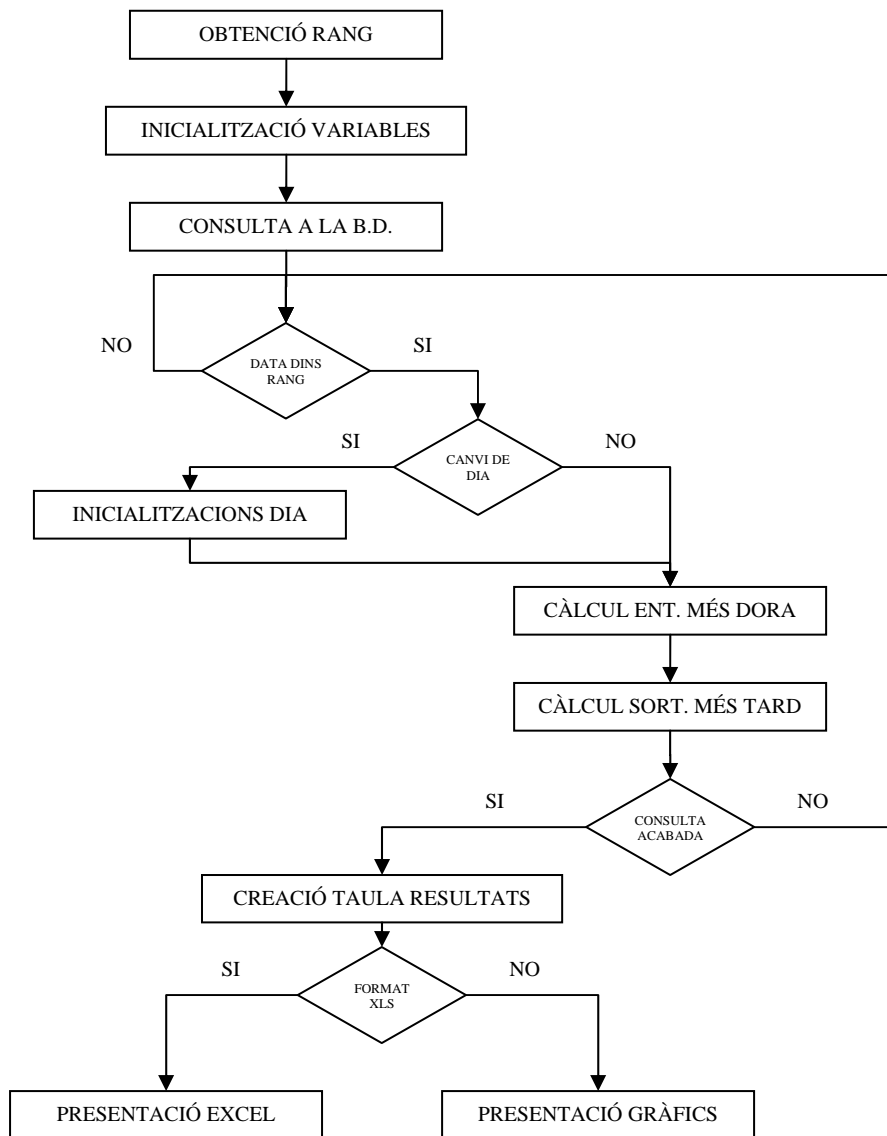


Diagrama 4.5.1.1

Anirem descrivint etapa per etapa la programació utilitzada:

OBTENCIÓ RANG: Aquesta part consta de diversos condicionals que analitzen si hi ha valor en les variables que ens canvien el rang. Si hi ha canvi de data, l’agafem i la posem en una variable de sessió. En el cas en que no ens demanin un canvi de rang,

mirarem si hi ha valors en les variables de sessió que ens marquen un rang seleccionat anteriorment. Si tampoc no tenen valor les variables de sessió, el que fem és agafar la primera i l'última data de l'arxiu per efectuar l'anàlisi.

INICIALITZACIÓ VARIABLES: En aquest punt, definim les variables que s'han de fer servir durant tot el programa i els hi assignem el valor d'inicialització.

CONSULTA A LA BASE DE DADES: Com s'ha comentat, la comunicació amb la base de dades l'hem fet amb una classe, de manera que per fer aquesta consulta, creem un objecte de la classe Consulta, i llavors li enviem la sentència del tipus "select", que ens assigni totes les dades de la taula moviments del niu seleccionat a una taula.

DATA DINS DEL RANG: Un cop tenim les dades en una taula, les anem llegint i amb dos condicionals només passem a tractar les que estan dins del rang d'anàlisi. Anem recorrent la taula amb la instrucció "while".

CANVI DIA: quan es produeix un canvi de dia, entrem a fer les inicialitzacions del dia:

INICIALITZACIONS DIA: Posem tots els comptadors a "0", per poder començar els càlculs de cada dia i el punter de referència del dia li donem el valor del nou dia. El punter d'entrada més aviat, el posem al final del dia (1440 min.) i el de sortida més tard a primera hora del dia (0 min.).

CÀLCUL ENTRADA MÉS AVIAT: Mirem l'acció del moviment actual, si el seu valor és una "entrada", agafem l'hora d'aquesta acció, li separem els camps, i passem hores i minuts tot a minuts. Llavors es compara amb la variable d'entrada més aviat. Si és més aviat que la de referència, passem l'hora de l'entrada a la de referència, ja que de moment ha estat la primera. El resultat, el passem a hores i el guardem en una taula on la referència de la posició és la data actual, de manera que queda guardada en la posició de data actual l'hora de la primera entrada.

CÀLCUL SORTIDA MÉS TARD: El procés és el mateix que en la primera entrada, però ho fem amb les sortides, de manera que creem una altra taula on les referències de posició són les dates, i els valors són les hores de les sortides més tard.

CONSULTA ACABADA: Anem repetint el bucle del “while” fins que hem recorregut tota la taula amb les dades.

CREACIÓ TAULA DE RESULTATS: Depenent del que ens demanin en l’anàlisi, creem una taula de taules, és a dir, segons la sol·licitud contindrà la taula d’entades, de sortides o bé les dues.

Finalment i depenent de la variable “\$format” que ens arribi, entrarem en la part del programa que ens faci Excel o gràfics, que ja hem explicat el codi.

GRAFIC DE LES PRIMERES ENTRADES I ULTIMES SORTIDES

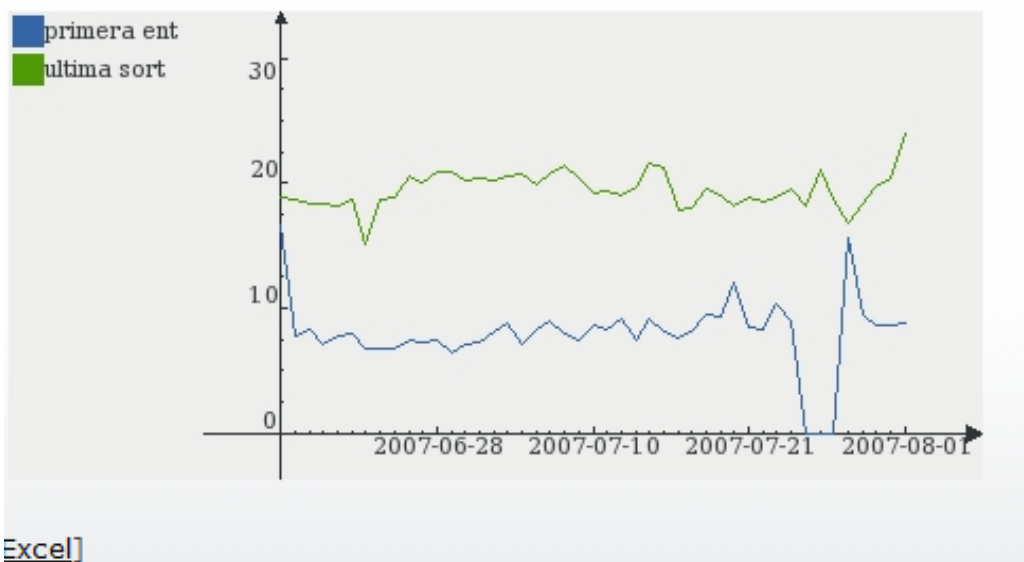


Fig. 4.5.1.1

4.5.2 – ACTIVITATS DIARIES NIU

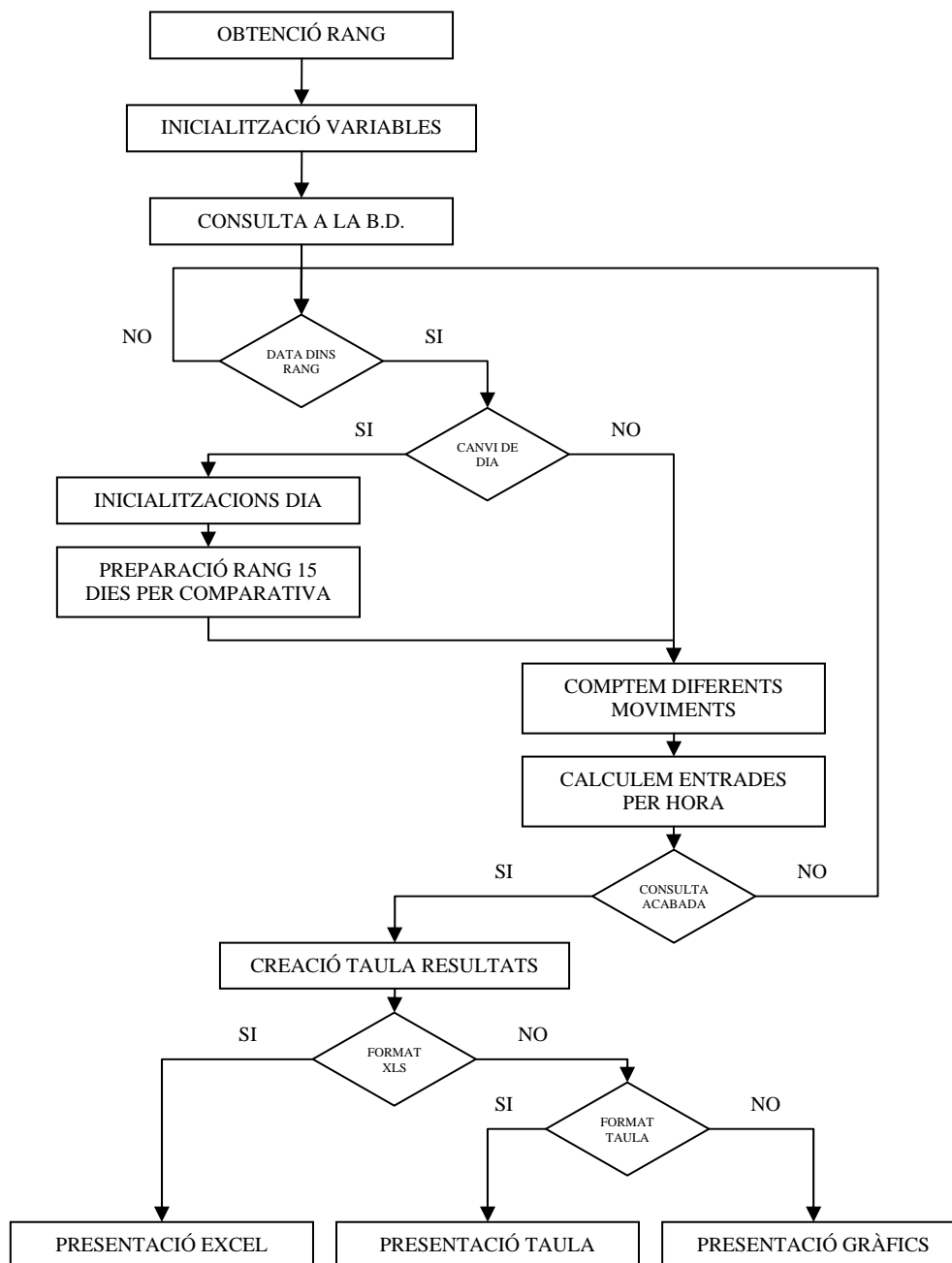


Diagrama 4.5.1.2

Hi ha molta part del programa que és comú, de manera que comentarem el que és la part central del programa, el tractament de dades pròpiament dit, que és on hi ha les diferències.

PREPARACIÓ RANG 15 DIES PER COMPARATIVA: Al canviar de dia, hem posat un comptador que va comptant fins a 15, ja que la gràfica d'activitats també la fem servir amb l'opció d'anàlisi del comptador de 15 dies, per veure per on ens anem movent. També portem una variable que ens compta el màxim nombre d'entrades en un dia de tota la taula, per marcar el rang en la comparativa, de manera que independentment del niu, fem la marca del rang proporcionada al valor més alt d'entrades d'aquell niu.

COMPTEM DIFERENTS MOVIMENTS: En aquesta part, mirem de quin tipus és l'acció. Per exemple, amb un condicional mirem si el valor és el "d'entrada", i si ho és augmentem la variable que ens fa de comptador i la introduïm en una taula indexada per les dates, de manera que el valor d'entrades per la data que està mirant, incrementa una unitat. Tenim tantes taules, variables i condicionals com accions possibles, set.

CÀLCUL D'ENTRADES PER HORA: Al mateix temps, anem creant una taula amb el nombre d'entrades de cada dia i cada hora, per poder mostrar al final el format taula d'aquests valors.

CREACIÓ TAULA RESULTATS: El procés és similar a l'anterior anàlisi, però aquí la taula que preparem és de set possibles taules de resultats segons les opcions demanades.

Mostrem dues gràfiques, la primera és la normal d'activitats del niu, en canvi, la segona és quan la posem al costat de la gràfica del comptador de 15 dies per veure quin rang estem tractant.

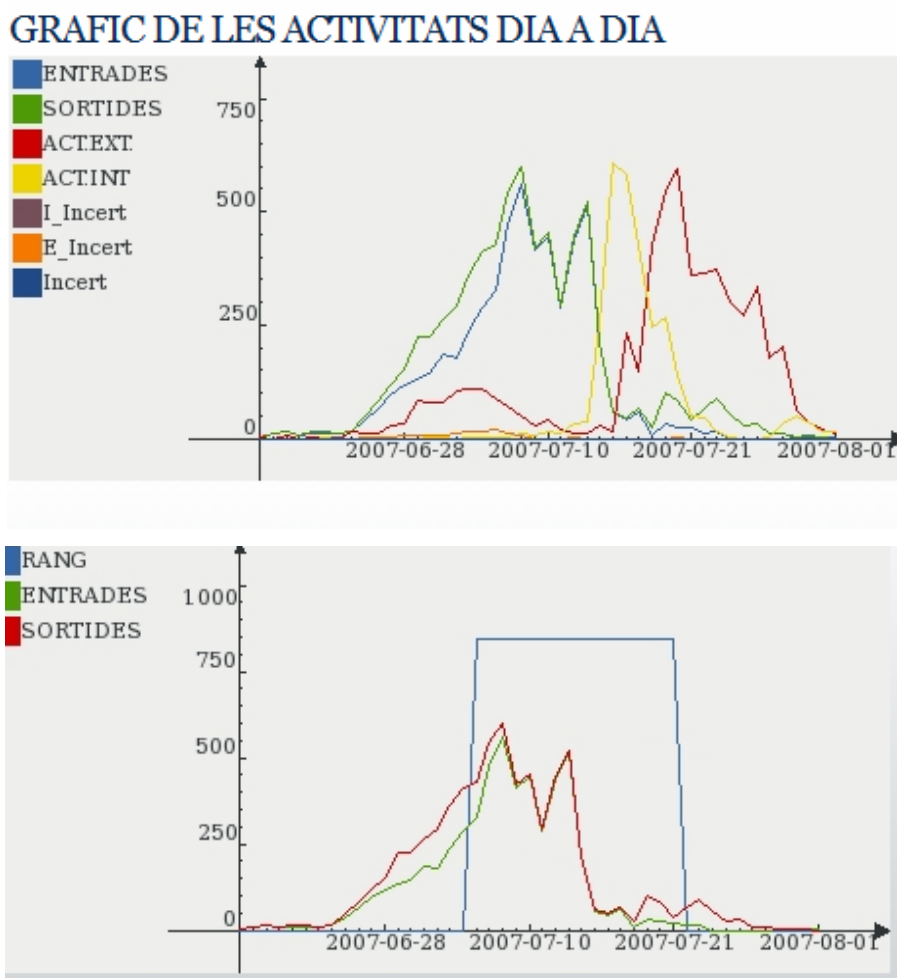


Fig. 4.5.1.2

4.5.3 – ENTRADES/SORTIDES DIARIES

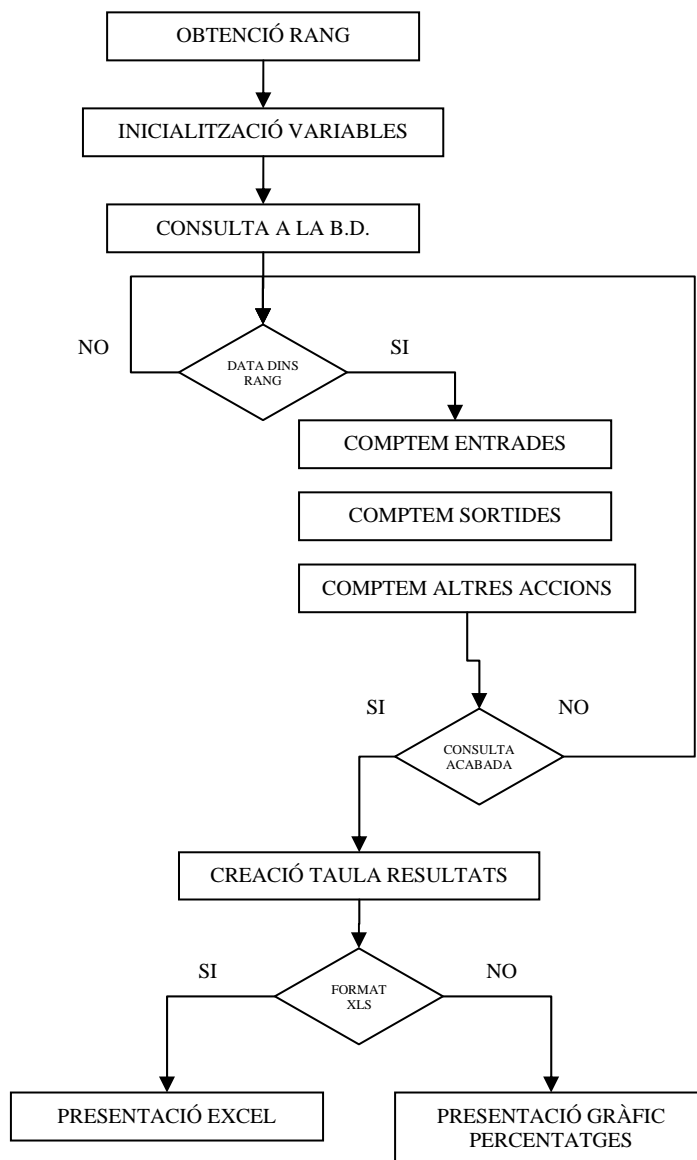


Diagrama 4.5.1.3

COMPTEM ENTRADES: Portem un comptador per les entrades, i el resultat el posem en una taula.

COMPTEM SORTIDES: Portem un comptador per les sortides, i el resultat el posem en una segona taula.

COMPTEM ALTRES ACCIONS: Portem un comptador on hi entrem la resta d'accions que no són ni entrades ni sortides, i el resultat el posem en una altra taula.

La gràfica resultant és la següent:

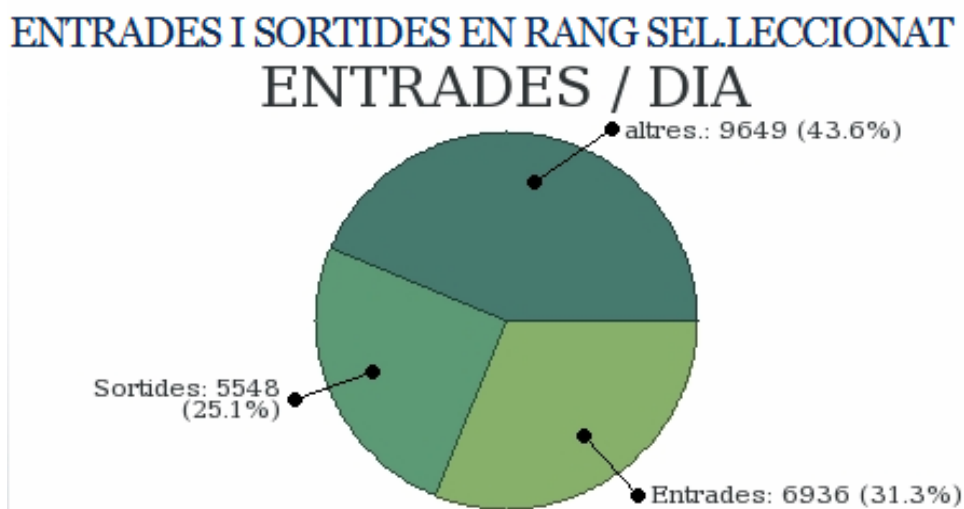


Fig. 4.5.1.3

4.5.4 – COMPARATIVA ENTRADES/SORTIDES 2 RANGS

Mostrem l'esquema de la segona gràfica de la comparativa, ja que la primera és la de l'esquema del punt 4.5.3, però en gràfic de barres. En aquest anàlisi, mostrem dues gràfiques, una és la d'entrades i sortides del punt anterior i l'altra és la que definim en aquest punt, que també calcula les entrades i sortides com l'anterior, però en aquesta el tercer valor no són les activitats restants com l'anterior, sinó que ens mostra el total. Aquest anàlisi ens va molt bé per poder mirar el total d'entrades i sortides en dos rangs diferents dins el mateix niu.

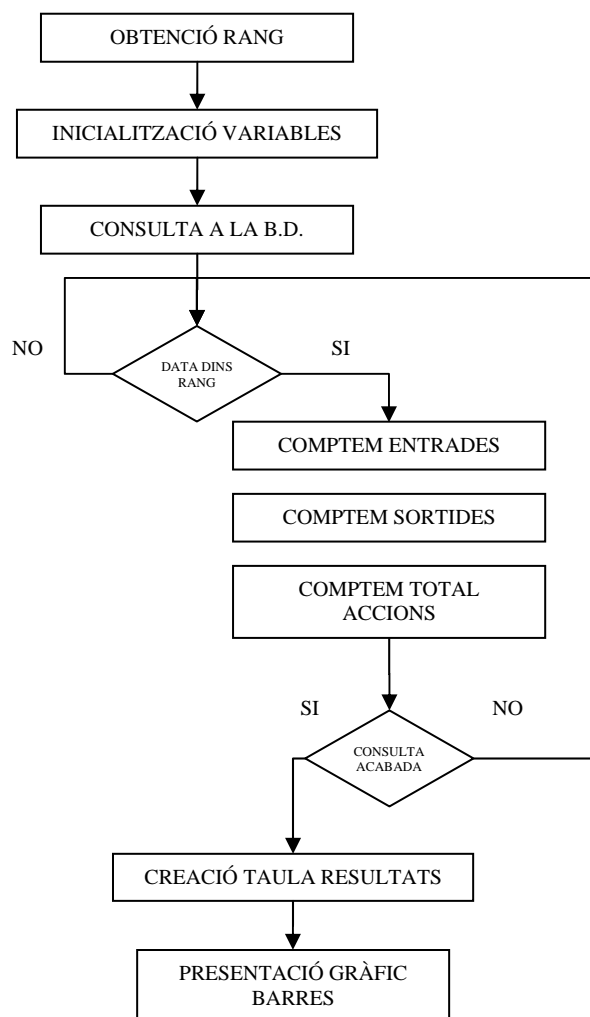


Diagrama 4.5.4.1

La única diferència amb l'esquema del punt anterior, és que el tercer comptador, compta el total de moviments enregistrats al niu, i que la representació únicament pot ser amb un gràfic de barres. Les variables que porten les dates pel canvi de rang són diferents que les anteriors, per tal de poder analitzar dos períodes diferents.

La representació en el navegador és:

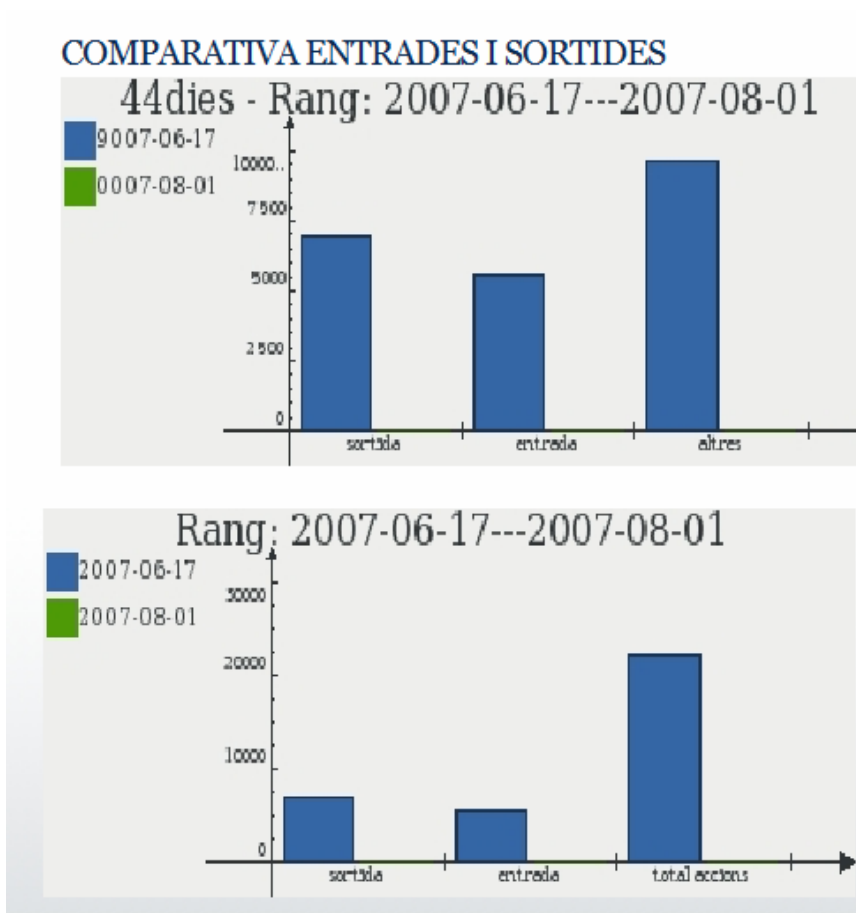


Fig. 4.5.4.1

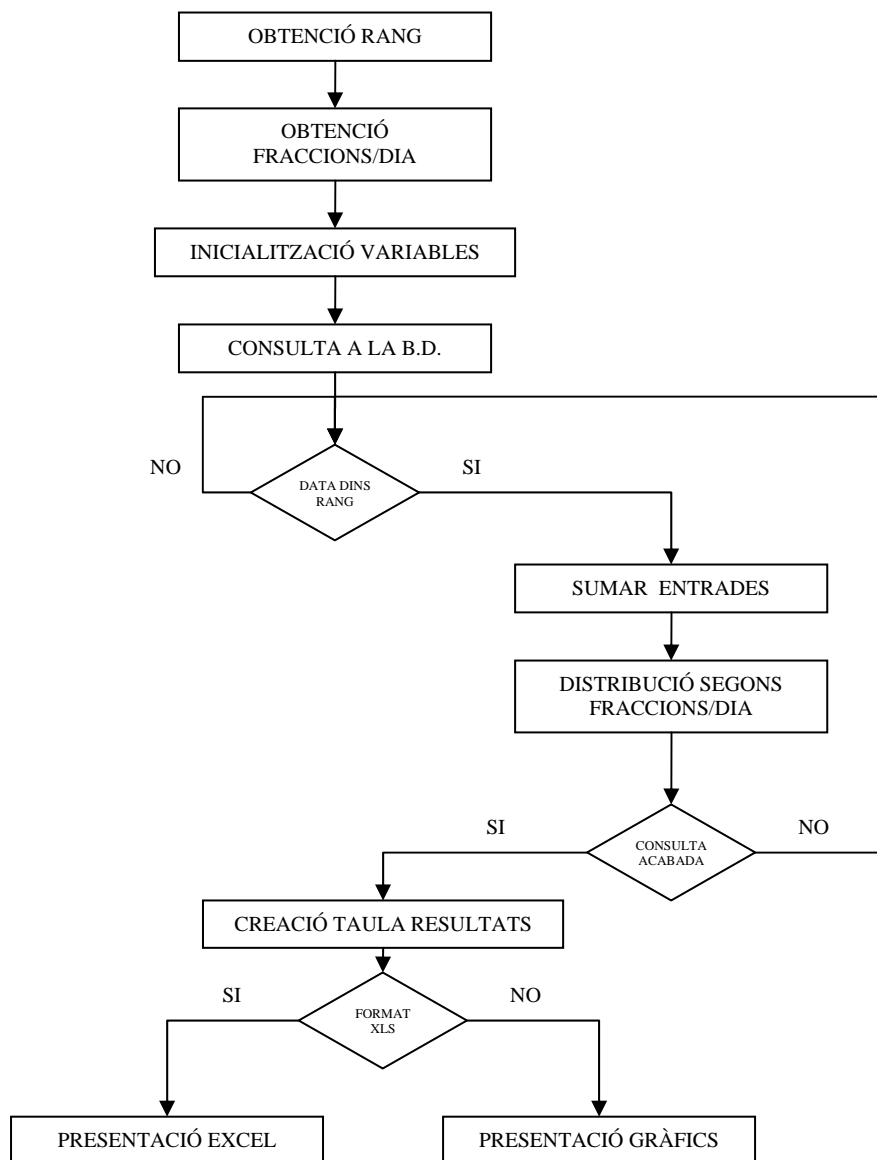
4.5.5 – DISTRIBUCIÓ HORA D'ENTRADES DURANT EL DIA

Diagrama 4.5.5.1

SUMAR ENTRADES: Si l'acció és una entrada, agafem l'hora de l'entrada i ho passem a minuts.

DISTRIBUCIÓ SEGONS FRACCIONS/DIA: Un cop tenim les hores de les entrades en minuts, dividim els minuts d'un dia en les fraccions seleccionades, fem una taula

amb tantes posicions com fraccions del dia volem, i anem distribuïnt les entrades dins les fraccions. Per exemple, si les fraccions són dos, doncs repartirà les entrades entre les d'abans de les dotze del migdia i les de després. Si la fracció és de 24 (valor predeterminat), ens mostra les entrades de cada hora.

La gràfica és la següent:

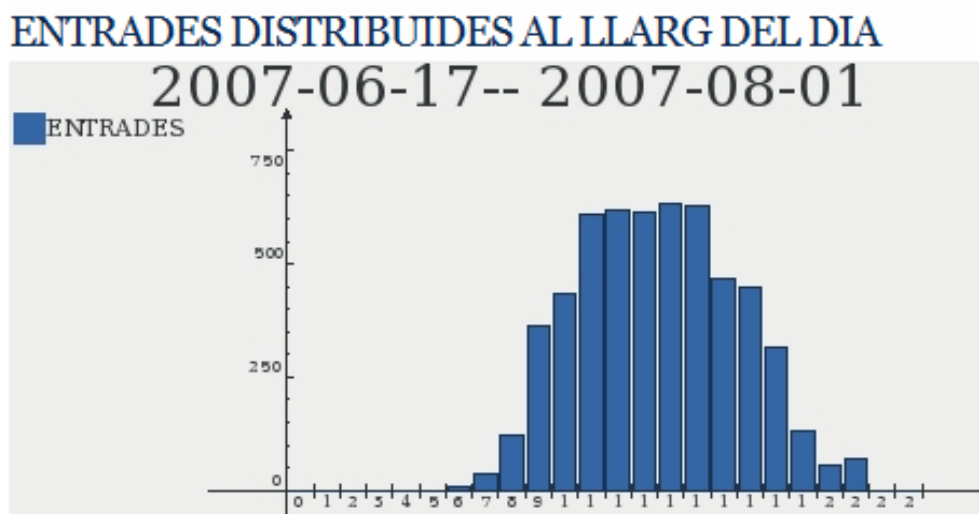


Fig. 4.5.5.1

4.5.6 – COMPTADOR DIARI D'OCELLS EN NIU

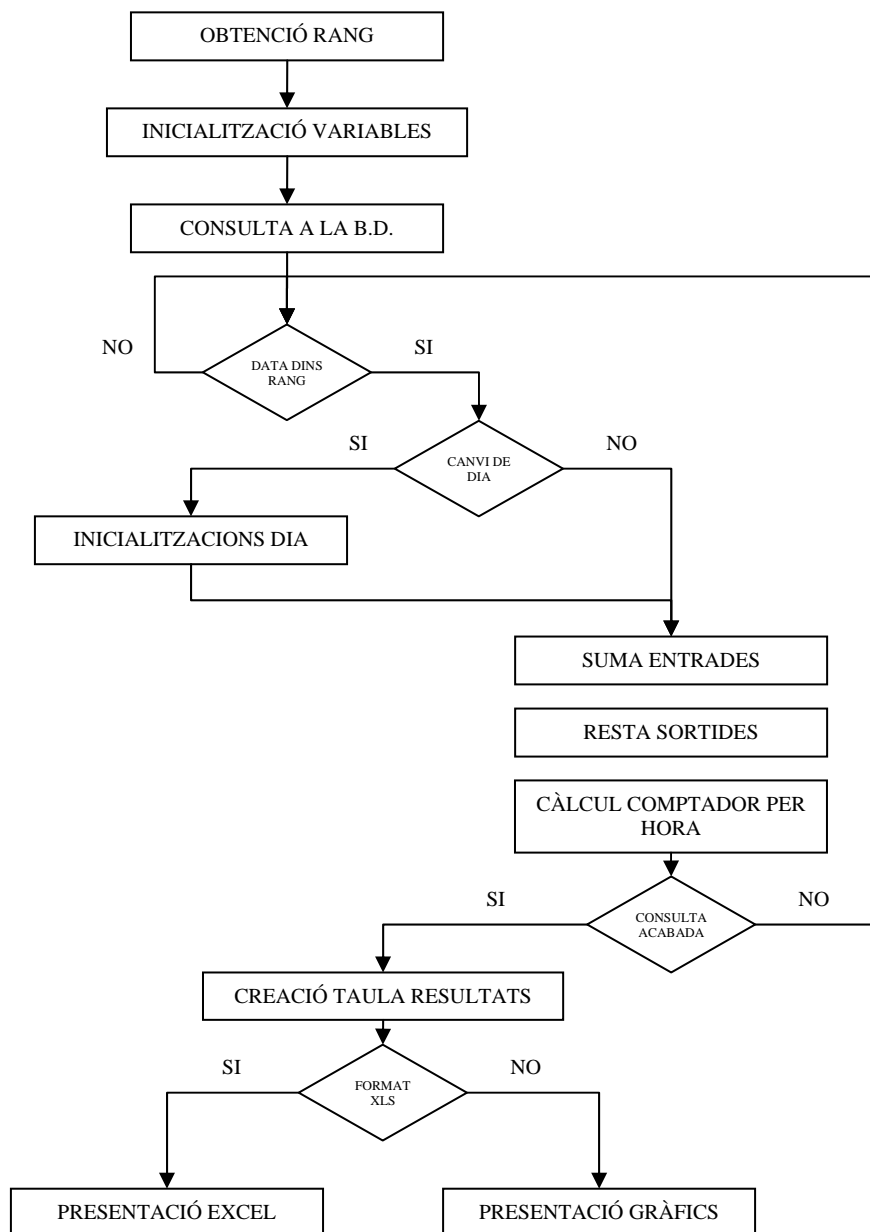


Diagrama 4.5.6.1

En aquest anàlisi, el que fem és analitzar les accions del niu. Quan és una entrada incrementem el comptador, quan és una sortida el disminuïm, i el resultat l'introduïm en

una taula cada hora durant 15 dies, amb la particularitat de que cada dia tornem el comptador a zero.

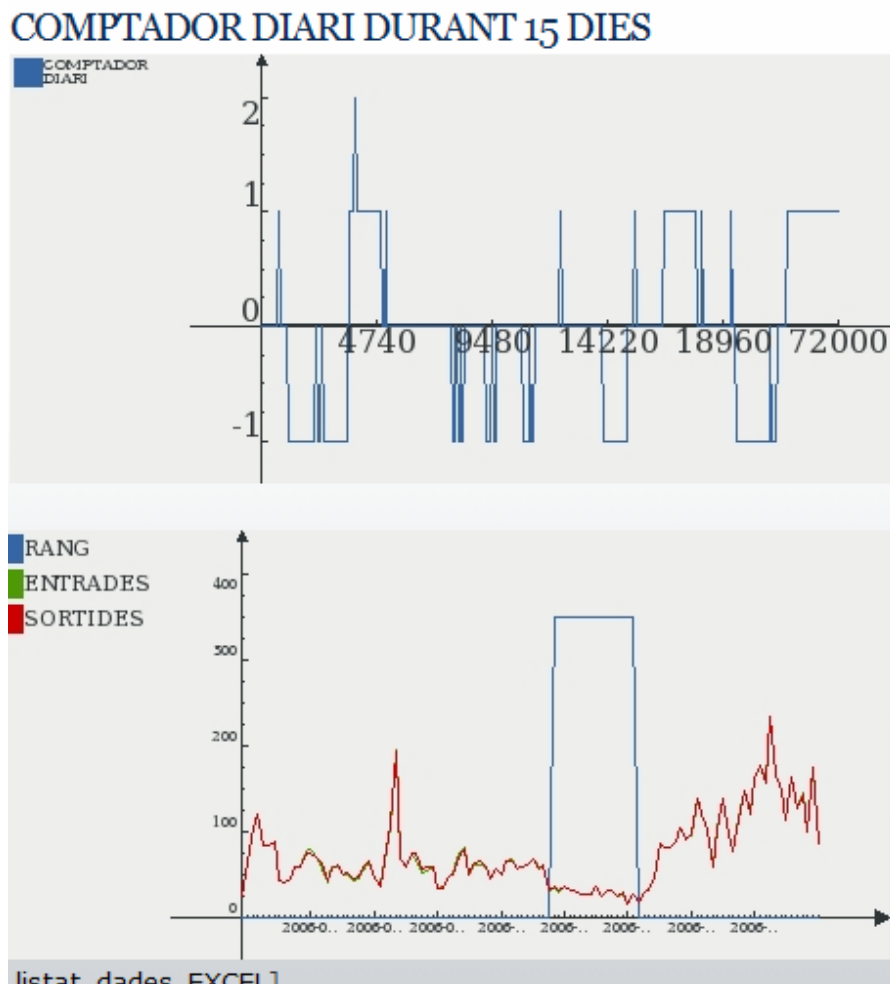


Fig. 4.5.6.1

4.6 – QUANTIFICACIÓ DEL PROJECTE

Anem a parlar del projecte amb números:

Per la realització del projecte hem pogut disposar de tres arxius de dades reals captades per e-nius:

- Abellerols : 24.250 moviments.
- Xots : 19.946 moviments.

- Pardals : 2.757 moviments.

Per a la capa de dades, s'ha creat una base de dades relacional amb quatre taules:

- Eniu : 3 camps creats.
- Persones : 7 camps creats.
- Niu : 7 camps creats.
- Moviments : 5 camps creats.

Per a la capa de processament s'han creat 17 arxius PHP:

- index.php.
- analisi.php.
- analisis.php.
- graficsa.php.
- gestio.php.
- enius.php.
- eniusb.php.
- eniusc.php.
- gestioad.php
- altesad.php.
- Consulta.php.
- grafmaxmin.php.
- grafactiv.php.
- grafrang.php.
- grafrang2.php.
- graffracc.php.
- grafcompta.php

Per a la capa de presentació, s'han creat 10 arxius TPL i un CSS:

- style.css.
- index.tpl.

- analisi.tpl.
- analisisb.tpl.
- graficsa.tpl.
- gestio.tpl.
- enius.tpl.
- eniusb.tpl.
- eniusc.tpl.
- gestioad.tpl
- altesad.tpl.

D'aquests arxius .tpl, se n'haurien de crear un per cada idioma possible, de manera que amb 4 idiomes estaríem parlant de 40 arxius .tpl.

5 – CONCLUSIONS I PROPOSTES D'AMPLIACIÓ

5.1 – CONCLUSIONS

En aquest projecte, com ja s'ha comentat els principals objectius són dos. En primer lloc proporcionar un suport de gestió de dades d'ornitologia per als nius informatitzats. S'ha aconseguit donar un ventall de possibilitats d'anàlisi de les dades amb diversos formats de presentació, perquè d'aquesta manera, l'usuari ja no només veu la presentació de l'anàlisi en unes gràfiques, sinó que les aconsegueix en format Excel pel possible tractament posterior que es desitgi. Esperem que aquest objectiu s'hagi aconseguit.

El segon objectiu ha estat el poder treballar en la creació d'un entorn web construït amb el màxim de tècniques possibles de creació d'entorns web professionals, amb tot l'aprenentatge que això comporta. Aquest segon objectiu s'ha acomplert sobradament ja que s'han tractat tots els camps de la programació amb el sistema de les tres capes, l'emmagatzemament en una base de dades, el posterior tractament de dades i finalment la posterior presentació en el navegador, amb totes les tècniques de transmissió de variables entre arxius, etc.

El fet de crear un entorn web, l'única limitació que ens hem trobat ha estat el temps, ja que el tema de la programació és pràcticament il·limitat, hi ha milers de coses que podríem anar fent i aplicant. Conscients d'aquestes possibilitats que presenta el projecte, enunciaré una colla de possibles ampliacions que es podrien fer per continuar fent créixer aquest projecte que és l'e-niu.

5.2 – PROPOSTES D'AMPLIACIÓ

Algunes de les propostes d'ampliació que hem pensat són les següents:

- **PARAMETRITZACIÓ DE GRÀFIQUES PER L'USUARI:**

En aquest punt, s'ha pensat que al mateix seleccionar el tipus de gràfic que volem veure, es podria fer un menú d'opcions per controlar els paràmetres del gràfic, com són els eixos, etiquetes sobre gràfic, colors, format de la llegenda, etc. Tots aquests paràmetres els podem modificar mitjançant les classes del EZ-COMPONENTS. De manera que fent un tractament molt més elaborat del programa de gràfics, i amb un increment notable de les variables a tractar es podria realitzar.

Fins i tot, hem pensat que es podria fer una taula a la base de dades amb aquests paràmetres perquè es carreguessin automàticament i mostrés el format corresponent a cada usuari al connectar-se.

- **CARACTERÍSTIQUES E-NIU::**

Hem pensat que seria interessant el poder disposar de les característiques tècniques de l'e-niu en algun lloc de la web, ja que així podríem saber exactament el tipus d'e-niu utilitzat, i poder comparar amb el temps les possibles variacions que hi pot haver en l'arquitectura o en els components en futures versions de nius informatitzats.

- **COMPARATIVA DE DIVERSOS NIUS SIMULTANIS:**

Aquí s'hi podria incloure dos tipus de comparatives entre nius diferents, per una banda podria comparar dos nius d'ocells totalment diferents, així podríem veure les diferències de comportament entre tots dos, per exemple en els mateixos dies. D'altra banda, podríem comparar dos arxius diferents però del mateix niu, de manera que comparariem el comportament del mateix ocell en el mateix niu en diferents èpoques.

Això ho podríem realitzar incrementant alguns camps en la taula del niu, i ampliant l'arxiu que s'encarrega de presentar els resultats.

- **SISTEMA PER TREURE DADES DE L'E-NIU.**

En aquests moments, les dades s'extreuen del niu connectant un ordinador portàtil amb el niu mitjançant una connexió USB. Una altra opció per a treballar-hi, seria en la transmissió de dades e-niu-PC per radiofreqüència, ja que a més de poder agafar les dades sense que l'ocell se n'adonés, podríem posar els nius en determinats llocs encara que fossin més inaccessibles per l'usuari, ja que només l'hauria d'agafar per canviar les bateries.

- **AVÍS ESTAT DE LA BATERIA:**

Un altre apartat on es podria treballar, seria en l'avís de bateria baixa, ja que l'e-niu ens informa periòdicament en l'arxiu de dades de l'estat de les bateries, de manera que fins i tot es podria fer una mena de gràfic on s'anés mostrant la descàrrega de bateries, i havent fet moltes proves i estudis, es podria preveure la data probable en que s'esgotaran definitivament les bateries, avisant a l'usuari en la web amb uns dies d'antelació prudencials, que al niu s'hi ha de canviar les bateries.

- **MAPA GPS:**

Una altra opció seria la de generar unes coordenades UTM de la posició exacta dels nius, de manera que a la web es podria mostrar un mapa amb la posició dels nius actius, i aquí seria un bon lloc per anar mostrant l'estat de bateries de cada niu, ja que tindríem molta informació en un sol cop d'ull.

Aquestes han estat tan sols unes de les moltes possibles ampliacions que permet aquest projecte de l'e-niu. Han començat amb la part de hard construint l'e-niu, hem continuat creant aquest entorn web per la gestió de les dades i aquí queda aquest projecte obert per a qualsevol que vulgui contribuir realitzant alguna de les ampliacions presentades o amb qualsevol altra proposta.

6 – BIBLIOGRAFIA

- CLASSES I OBJECTES EN PHP. <http://www.blasten.com/contenidos/?id=19013> (Consulta: setembre 2007)
- DESCARREGA SERVIDOR APACHE. <http://www.tallerwebmaster.com/Tutorial-Instalacion-de-AppServ-PHP-Apache-MySQL-c-67.html> (Consulta: juliol 2007)
- DESCARREGA SMARTY. <http://smarty.php.net/> (Consulta: novembre 2007)
- DESCARREGA SPEADSHEET PER EXTREURE EXCEL. http://pear.php.net/package/Spreadsheet_Excel_Writer (Consulta: desembre 2007)
- DISSENYS DE TEMPLATES. <http://www.oswd.org> (Consulta: novembre 2007).
- EXEMPLE DE SMARTY. http://en.wikipedia.org/wiki/Smarty#Code_example (Consulta: novembre 2007)
- GUIA DE CSS. <http://www.w3c.es/Divulgacion/Guiasbreves/HojasEstilo> (Consulta: novembre 2007)
- LÓPEZ ROCAFIGUERA, Enric ; MARTÍ PUIG, Pere (2005). *Módulo 2: Internet (segunda parte)*. Tecnologías de la Información y la Comunicación. EPS – UVIC.
- MANUAL DE PROGRAMACIÓ EN PHP. <http://www.desarrolloweb.com/manuales/12> (Consulta: juliol 2007)
- MANUAL EZ-COMPONENTS. <http://ezcomponents.org/docs/tutorials/Graph#bar-charts> (Consulta: setembre 2007)
- MANUAL HTML AMB CSS. <http://www.w3.org/Style/Examples/011/firstcss> (Consulta: novembre 2007)
- MANUAL SQL. <http://www.manualdephp.com/manualphp/lenguaje-sql-php.html> (Consulta: juliol 2007)
- PORTAL E-NIUS. <http://www.tuatara.info/e-niu/news.php> (Consulta: agost 2007)